Recognition and Positioning of Rigid Objects using Algebraic and Moment Invariants

by Gabriel Taubin Licenciado en Ciencias Matemáticas, Universidad de Buenos Aires, Argentina, 1981

Thesis Submitted in partial fulfillment of the requirements for the Degree of Doctor of Science in the Division of Engineering at Brown University © Copyright by Gabriel Taubin 1991

Abstract of "Recognition and Positioning of Rigid Objects using Algebraic and Moment Invariants," by Gabriel Taubin, Ph.D., Brown University, May 1991

In this thesis we describe two model-based approaches to 3D rigid object recognition and positioning from range data, particularly in cluttered environments. Following a *model-based* approach, we recognize and locate objects in the data data set by comparing and geometrically matching small regions of the data with corresponding regions of known models, stored in a database. Due to the problem of occlusion, a known object is represented in the database as a hierarchical collection of regions, each of them approximated by a parameterized *model*. We use two types of parameterized models. With the first type we consider objects whose boundaries can be well approximated by piecewise algebraic curves or surfaces, or both, in which case the preliminary recognition and matching is based on comparing the coefficients of the corresponding polynomials; the final recognition and matching is based on determining how well the data fits a stored polynomial model. With the second type, we consider more general objects, objects which do not fall into the previous group, and use moments as the region descriptors. In order to develop practical systems for object recognition and position estimation, a number of problems must be solved first. Problems solved for this purpose, in this thesis, are the following. The first of these problems is how to fit models to regions of the data sets. Although computing moments from a data set is relatively straightforward, fitting algebraic curves and surfaces to regions of a data set is difficult. We show several different efficient and numerically stable algorithms for fitting implicit 3D curves and surfaces to data sets, in particular, for fitting algebraic 3D curves and surfaces to data sets. The algorithms also apply to 2D curves, and to curves and surfaces in fourth and higher dimensions. We also show how these fitting methods can be used in segmentation algorithms. The second problem is that when the coordinate system changes, the coefficients of the polynomial which define a given curve or surface change, and the same happens with the moments of a region. However, both coefficients and moments change in a well known fashion. The recognition and matching is based on computing and comparing invariants of, either the coefficients of the polynomials, or the moments. Invariants are functions of the coefficients or the moments, which are independent of the coordinate system. We introduce computationally efficient algorithms for computing invariants. The third problem solved is the problem of recovering the coordinate transformation which best aligns two matching curves or surfaces, or two vectors of moments. We solve these problems by defining an intrinsic coordinate system for both algebraic curves, and moment vectors. The parameters of this intrinsic coordinate system are functions of the coefficients of the polynomials, or the moments, and are very inexpensive to compute. Finally, the fourth problem dealt with is preliminary ideas on how to organize all of these tools to build object recognition and positioning systems.

The Vita of Gabriel Taubin

Gabriel Taubin was born on June 12, 1958, in Morón, Province of Buenos Aires, Argentina. He received the "Licenciado en Ciencias Matematicas" degree from the University of Buenos Aires, Argentina, in 1981, with a Master's thesis entitled "Analytic Rings". From 1981 to 1983 he did graduate studies in the same institution, working in Category Theory. While an undergraduate student, he served as a teaching assistant in the Department of Mathematics, starting in 1979. After graduation, he served as a lecturer in the same department, teaching different topics, both at the undergraduate and graduate level, until 1983. In 1983 he interrupted his studies to work full time in a company that he found during his second year as a graduate student, designing and manufacturing electromedical devices, and enclosures for electronic equipment. An Argentine patent was granted for one of his designs. In 1986 he left his company and returned to graduate school, this time in the Division of Engineering at Brown University. During the period of four years of his residence at Brown, he spent two summers working at the IBM T.J. Watson Research Center on different projects related to his Ph.D. thesis research. He was recipient of a Brown University fellowship for the academic year of 1986-1987, and two IBM Fellowships. The first IBM fellowship was granted by the Manufacturing Research Program for the academic year of 1988-1989. The second one was granted by the Computer Science Program for the academic year of 1989-1990. He is a member of the Institute of Electrical and Electronic Engineers (IEEE) and the Society of Industrial and Applied Mathematics (SIAM). Currently, his research interests are in the applications of Mathematics, particularly Geometry, to Computer Vision and Robotics problems, and more generally to the broad range of problems in Intelligent Machines.

Acknowledgments

My sincere thanks to my thesis advisor, Professor David B. Cooper, for his support, continuous encouragement, and friendship during the four years of my stay at Brown University.

My special thanks to Dr. Ruud M. Bolle for his continuous help during the course of these years. He made it possible for me to perform significant parts of the research, which lead to this thesis, while working with him on my two summer internships at IBM.

I want to express my appreciation to Professor Bruno Cernuschi-Frias, who made it possible for me to come to the United States, and to all my former professors and colleagues from the Department of Mathematics, at the University of Buenos Aires, Argentina, who gave me my basic knowledge, and shaped me as a researcher.

I have been lucky to interact with Professor Ulf Grenander, and have been benefited from his broad knowledge and experience. Some of our conversations lead to parts of this thesis. I also want to thank him, as well as the other reader, Professor William Wolovich, for taking their time to read this thesis.

Finally, I profoundly want to thank my wife Patricia, for her unmeasurable love, support, and understanding during these difficult years, since the moment we decided to undergo the painful cultural experience of coming to live in a different country. I could not have achieved what I did without her.

Contents

Acknowledgments		
1	Introduction1.1Geometric properties of implicit curves and surfaces	1 5 7 7 8
2	Implicit Curve and Surface Fitting2.1Implicit curves and surfaces2.2Object representation, segmentation and recognition2.3Approximate distance2.4Approximate mean square distance2.5Mean square error2.6Generalized eigenvector fit2.7Computational cost2.8Independence and invariance2.9The reweight procedure2.10Other families of curves and surfaces2.10.1Transformations of a curve or surface2.10.2Projection of space curves onto the plane2.10.3Parameterizations of some transformation groups2.10.4Cylinders, algebraic curves and surfaces2.10.5Superquadrics2.11Related work on implicit curve and surface fitting2.12Analysis of the generalized eigenvector fit	 10 11 13 17 20 22 22 27 28 32 34 35 35 37 39 40 41 43
3	Segmentation Algorithms3.1 Noise variance estimation3.2 Goodness of fit tests3.3 Region growing3.4 Merging3.5 Experimental results	46 47 48 52 54

4	Inva	ariant theory of algebraic forms	61
	4.1	Polynomials, forms, and linear transformations	64
		4.1.1 Polynomials	66
		4.1.2 Forms, homogeneous coordinates,	
		and projective transformations	66
		4.1.3 Vectors and matrices	67
		4.1.4 Euler's theorem	70
		4.1.5 Transformation rules	71
	4.2	Algebraic invariants and covariants	73
		4.2.1 Covariant and contravariant matrices	76
		4.2.2 Construction of algebraic invariants and covariants	77
		4.2.3 Construction of covariant and contravariant matrices	80
		4.2.4 Some examples of algebraic invariants	81
	4.3	Cartesian invariants and covariants	83
		4.3.1 The invariant inner product of forms	83
		4.3.2 Covariant matrices and the characteristic polynomial	84
		4.3.3 The harmonic decomposition	85
		4.3.4 Computing the harmonic decomposition	86
	4.4	Proofs	88
5	Geo	ometric Matching of Algebraic Curves and Surfaces	95
	5.1	The center of a planar curve or surface	99
		5.1.1 General case	99
		5.1.2 Singular case	101
	5.2	The center of a 3D curve	102
	5.3	Orientation of a curve or surface	102
	5.4	Geometric matching of groups of curves or surfaces	104
	5.5	A Remark on algebraic curve and surface fitting	106
	5.6	Proofs	107
6	Moi	ment invariants	108
U	6.1	Moments of discrete and continuous data sets	109
	6.2	Centered moments	111
	6.3	Cartesian and affine moment invariants	112
	0.0	6.3.1 Cartesian moment invariants	112
		6.3.2 Affine moment invariants	112
	64	Canonical frame of reference	113
	0.1	641 Cartesian case	114
		642 Affine case	114
	65	Examples and implementation	116
	6.6	Global testing and distances between shapes	121
	0.0	6.6.1 Shape polynomials	121
		6.62 Comparing shapes with the Λ -distances	122
	67	Related work on moment invariants	12/
	0.7		120

	6.8 Proofs	130
7	Recognition Algorithms	132
8	Conclusions 8.1 Future directions	134 135
A	The Levenberg-Marquardt algorithm	136

List of Figures

1.1	Basic structure of the recognition systems	2
1.2	Example of intrinsic frame of reference computation.	4
1.3	Implicit surface defined by a third degree polynomial	5
1.4	A pair of cylinders represented as a single fourth degree surface.	6
1.5	Fourth degree interest region.	9
2.1	Different representations of the same space curve	12
2.2	Degree 5 planar curve generalized eigenvector fit.	14
2.3	Degree 5 planar curve fit after reweight procedure	14
2.4	Regions well approximated by fourth degree algebraic curves	15
2.5	Regions well approximated by fourth degree algebraic curves	16
2.6	Third degree algebraic surface fit.	16
2.7	Approximate and real distance to the curve $\{x : x_1x_2 = 0\}$.	19
2.8	Approximate and real distance to the curve $\{x : 8x_1^2 + (x_2^2 - 4)^2 - 32 = 0\}$.	19
2.9	Generalized eigenvector space curve fit	25
2.10	Fourth degree algebraic surface fit.	25
2.11	Generalized eigenvector space curve fit	26
2.12	Invariance to translation and rotation.	30
2.13	Invariance to translation and rotation.	31
2.14	The reweight procedure for the generalized eigenvector fit	33
2.15	Reweight procedure improving generalized eigenvector fit	34
2.16	Simple position estimation.	36
3.1	Region growing algorithm for planar curves.	50
3.2	Core of region growing algorithm for planar curves.	51
3.3	Merging procedure.	53
3.4	Segmentation of planar curve	55
3.5	Segmentation of planar curve	56
3.6	Segmentation of planar curve	57
3.7	Segmentation of 3D scene : sphere, cylinder and box.	58
3.8	Segmentation of 3D scene : spray bottle and cylinders.	59
3.9	Segmentation of 3D scene : NRCC "jet4" file	60
5.1	Example of intrinsic frame of reference computation.	97
5.2	Example of intrinsic frame of reference computation.	98

6.1	Some affine moment invariants	117
6.2	Some affine moment invariants	118
6.3	Some affine moment invariants	119
6.4	Shape polinomials for data scattered around an ellipse	124
6.5	Shape polynomials for the boundary of a wrench.	125
6.6	Shape polynomials for the boundary of a plier.	126

Chapter 1 Introduction

In this thesis we describe two model-based approaches to 3D rigid object recognition and positioning from range data. Due to the problem of occlusion, known solid objects are represented in a database as hierarchical collections of regions of the boundary curves and surfaces. The two approaches differ basically in the models used to represent the regions. In the first approach, we assume that the regions can be well approximated by piecewise algebraic surfaces. In the second approach we consider more irregular objects, and the descriptors for the regions are vectors of moments. We show that these two models have many common features, in particular the techniques for deciding whether two regions match or not, and for orienting two matching regions with respect to each other are almost the same.

Figure 1 describes the common structure of both approaches. Models, either algebraic surfaces or vectors of moments, are fitted to small regions of the data set. These regions are small enough so that most of them correspond to a single object, but big enough to contain sufficient information to uniquely determine the location and orientation of an object. Alternatively, models are fitted to smaller regions, and symbolical methods, such as polynomial multiplication in the case of algebraic curves and surfaces, are used to compute the parameters of the models which fit a group of these small regions. Since the parameters of these models are coordinate system dependent, a vector of *invariants* is computed for each model. An *invariant* is a function of the parameters which yields the same values independently of the viewer coordinate system. This vector of invariants is used to index into the database of regions of known models. Using these invariants, the database can be organized for an efficient search. The database search produces a list of triples. Each of these triples consists of a region, i.e., a portion of a 3D surface, the object that the region correspond to, and the coordinate transformation from the object coordinate system to the local region coordinate system. If one or more matches are found in the database, the *intrinsic coordinate system* of the data set region is computed, and, for every match found, the coordinate transformation which best aligns the data region with the model region is computed using the intrinsic coordinate systems of the two regions. This coordinate transformation constitutes a hypothesis of the presence of the associated object in the computed position and orientation. The hypotheses generated in this way are then globally tested, and the final interpretation of the data set is produced. The intrinsic coordinate system of a data region is a viewpoint independent co-



Figure 1.1: Basic structure of the recognition systems.

ordinate system computed from the model parameters, which is commonly refered to as the *object coordinate system* in the Computer Vision literature. The model parameters recomputed in the intrinsic coordinate system are invariants of the data region.

This global structure has many points in common with other well known object recognition systems, and the main contributions of this thesis are in the implementation of several steps of the recognition and positioning processes described above. We develop several efficient and computationally stable algorithms for fitting implicit surfaces to data, not only for algebraic surfaces, but for arbitrary families of implicit curves and surfaces parameterized by a finite number of parameters. These methods not only apply to 3D surfaces, but to curves and surfaces of any dimension, in particular for 2D and 3D curves. We show how these fitting algorithms can be used for segmenting 2D edge maps and range images. Then, we develop methods for computing invariants of polynomials. Since our intended application is the recognition and positioning of objects from range data, we concentrate on Euclidean invariants, functions of the coefficients of the polynomials which do not change after an Euclidean coordinate transformation. However, other related problems, such as the recognition of 3D objects from the projection of their occluding boundaries, require the computation of invariants with respect to projective, affine, or similarity transformations. We also show methods for computing invariants with respect to these three groups.

We emphasize the *computational* aspect of the process, based on well known, efficient and numerically stable matrix algorithms. For example, if M is a square $n \times n$ matrix, Q is an orthogonal $n \times n$, matrix, and $M' = Q^t M Q$, then both M and M' have the same eigenvalues, i.e., the eigenvalues of a square matrix are invariant under orthogonal transformations. Equivalently, the characteristic polynomials $\chi(\theta) = \det(M - \theta I)$ and $\chi'(\theta) = \det(M' - \theta I)$ are identical, i.e., the coefficients of this polynomial are invariants under orthogonal transformations. Furthermore, the eigenvalues of M and the coefficients of its characteristic polynomial are functionally equivalent, but the eigenvalues can be computed by numerical methods in a number of arithmetic operations proportional to n^3 , while the symbolical expansion of the determinant which defines the coefficients of characteristic polynomial has n! terms.

For every algebraic curve or surface, we define its intrinsic coordinate system as a function of the coefficients of the defining polynomials. This intrinsic coordinate system of an algebraic curve or surface is independent of the viewer coordinate system, in the sense that the polynomial equations of the same curve or surface in its intrinsic coordinate system have the same coefficients, independently of the viewer coordinate system. They are Euclidean invariants. For example, figures (1.2-a) and (1.2-b) show two cubic 2D curves given by the union of three straight lines extracted from the edge images, and figures (1.2-c) and (1.2-d) show their corresponding frames of reference. The computation of the intrinsic coordinate system is based on efficient and numerically stable matrix operations. We show that the tools developed for the computation of algebraic invariants also apply to the computation of moment invariants without modification. The intrinsic coordinate system of a set of points based on moments is also developed, not only for the Euclidean case, but for the similarity and affine cases. Finally, we develop several methods for globally testing the hypotheses generated, based on particular geometrical properties of both algebraic curves and surfaces,



Figure 1.2: (a): Cubic 2D curve, union of the three straight lines fitted to the data points in the dark region. (b): Intrinsic frame of reference for the curve in (a). (c): Cubic 2D curve, union of the three straight lines fitted to the data points in the dark region, matching the curve in (a). (d): Intrinsic frame of reference for the curve in figure (c).



Figure 1.3: Implicit surface defined by the third degree polynomial $f(x_1, x_2, x_3) = x_1^2 + x_2^2 - x_3(x_3^2 - 1) - 1$.

and moments.

1.1 Geometric properties of implicit curves and surfaces

An *implicit surface* in 3D is the set of zeros of a smooth function $f : \mathbb{R}^3 \to \mathbb{R}$ of three variables

$$Z(f) = \{(x_1, x_2, x_3) : f(x_1, x_2, x_3) = 0\}.$$

An implicit surface Z(f) is *algebraic* if the function f is a polynomial. For example, figure 1.3 shows an algebraic surface, the set of zeros of the third degree polynomial $f(x_1, x_2, x_3) = x_1^2 + x_2^2 - x_3(x_3^2 - 1) - 1$.

Similarly, an implicit 2D curve is the set of zeros of a smooth function $f : \mathbb{R}^2 \to \mathbb{R}$ of two variables

$$Z(f) = \{(x_1, x_2) : f(x_1, x_2) = 0\},\$$

an algebraic 2D curve is the set of zeros of a polynomial of two variables, an implicit 3D curve is the intersection of two 3D surfaces, the set of zeros of a two dimensional vector function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^2$ of three variables

$$Z(\mathbf{f}) = \{ (x_1, x_2, x_3) : \mathbf{f}(x_1, x_2, x_3) = 0 \},\$$

and an algebraic 3D curve is the set of zeros of a two dimensional vector of polynomials of three variables.

The representation of curves and surfaces in implicit form, as opposed to parametric form, has many advantages. In the first place, an implicit curve or surface maintains its implicit form after a change of coordinates, that is, if a set of points can be represented as a subset of an implicit curve or surface in one coordinate system, so can it be in any other coordinate system. That is not the case with data sets represented as graphs of functions of two variables, i.e., as depth maps, the patch descriptors produced by many well known segmentation algorithms. In the second place, the union of two or more implicit curves or surfaces can be represented as a single implicit curve or surface, the set of zeros of the product of the functions which define the individual curves or surfaces,

$$Z(f_1) \cup Z(f_2) \cup \cdots \cup Z(f_n) = Z(f_1 \cdot f_2 \cdots f_n),$$

so that groups of curve or surface patches, or eventually a whole object, can be represented as a subset of a single implicit curve or surface.

For example, the two cylinders in figure 1.4 are the sets of zeros of

$$\{x: x_1^2 + (x_3 - 1)^2 - 4 = 0\}$$
 and $\{x: x_2^2 + (x_3 + 1)^2 - 4 = 0\}$,

respectively. The nonplanar space curve that is the intersection of these two surfaces is the



Figure 1.4: A pair of cylinders represented as a single fourth degree surface.

set of zeros of the column vector

$$\mathbf{f}(x_1, x_2, x_3) = \begin{pmatrix} x_1^2 + (x_3 - 1)^2 - 4 \\ x_2^2 + (x_3 + 1)^2 - 4 \end{pmatrix}.$$

The 3D surface that is the union of the two cylinders is the surface defined by the set of zeros of the product

$$\{x: (x_1^2 + (x_3 - 1)^2 - 4)(x_2^2 + (x_3 + 1)^2 - 4) = 0\}.$$
 (1.1)

Hence, a single fourth degree polynomial can represent a pair of cylinders, and this is true for arbitrary cylinders, e.g., a pair that do not intersect. This property relaxes the requirements on a segmentation algorithm, and it is very important in regard to the matching problem, allowing the matching and aligning of groups of patches at once by using a single analytic function to represent the group of patches.

1.2 Fitting implicit curves and surfaces to data

The first problem that we have to solve is how to fit implicit curves and surfaces to data. Given a family of implicit functions parameterized by a finite number of parameters, and a finite set of points in space, assumed to belong to the same curve or surface, we would like to fit an implicit curve or surface to the data by estimating the parameters which minimize the mean square distance from the data points to the curve or surface defined by those parameters.

Unfortunately, there is no closed form expression for the distance from a point to a generic implicit curve or surface, not even for algebraic curves or surfaces, and iterative methods are required to compute it. We replace the real distance from a point to an implicit curve or surface by a first order approximation. The mean value of this function, on a fixed set of data points, is a smooth nonlinear function of the coefficients, and can be *locally* minimized using well established non-linear least squares techniques. However, since we are interested in the global minimum, and these numerical techniques find local minima, we still need procedures to choose good initial estimates. In the case of algebraic curves and surfaces, we replace the performance function. Instead of the approximate mean square distance we use a new approximation, turning the difficult multimodal optimization problem into a computationally attractive generalized eigenproblem. The curves or surfaces computed by this *generalized eigenvector fit* method usually produces good fits. The fits are often satisfactory, not requiring further improvement, and the required computation is modest and practical.

1.3 Geometric matching procedures

The approximate mean square distance can also be used to test hypotheses supported by several regions of the data set. Since we are dealing with range data, the hypotheses generation is based on a theory of Euclidean invariants of algebraic curves and surfaces. This theory of Euclidean invariants lets us decide whether two curves or surfaces of the same degree match or not. A positive answer to the matching problem would mean that the second curve or surface is almost equal to the first one, but after an unknown Euclidean coordinate transformation, i.e., a rotation and a translation. If two curves or surfaces match, the theory also lets us recover the Euclidean transformation which transforms the first curve or surface into the second one. We also describe new techniques for the efficient computation of projective, affine, and similarity invariants of algebraic curves and surfaces, which find application in other related problems.

For object recognition, we propose a system that involves indexing into a data base of objects represented by features consisting of groups of moderately high degree algebraic surfaces, or algebraic curves, or both. These high degree algebraic curves and surfaces have much more discriminating power than does an individual low degree algebraic surface such as a plane or a quadric surface. Hence, these high degree algebraic curves and surfaces are more powerful discriminatory features than are the quadric surfaces or the high curvature points that are usually used. Two types of representations are presently under consideration. One is the representation of a collection of a few low degree algebraic surfaces by a single algebraic surface of higher degree. For example, representing three planes by the set of zeros of a single third degree polynomial, the product of the three first degree polynomials, each representing one plane, or representing a quadric and a cubic surface by a single fifth degree algebraic surface. The simple low degree primitive surfaces used are those that can be found with modest computation. Exact segmentation is not necessary. Partial occlusion is not a problem; a primitive surface can be estimated from a portion of the primitive surface data. Once the primitives are found in the data, groups are then represented by single higher degree algebraic surfaces. The other type of representation are the interest regions, which are spherical regions in which the data is not well represented by a low degree algebraic surface, such as first or second degree, but is well approximated by an algebraic surface of one degree higher. For example, a region occupied by a portion of two intersecting cylinders would be represented exactly by a fourth degree surface and poorly by a lower degree surface if enough of the surfaces were sensed. More generally, a fourth degree surface might capture a chunk of information useful for recognition purposes on a natural irregular surface such as a face, whereas a lower degree surface might not. Useful interest regions are those having the stability that the polynomial does not depend on the exact placement of the sphere specifying the region of data to be used. For this approach, sphere sizes should be chosen such that most spheres will contain data well approximated by low degree surfaces, and only a few will require representation by higher degree surfaces. These higher degree surfaces then contain considerable discriminatory power for object recognition. Figure 1.5 show an example of an interest region.

In this way we can deal with the occlusion problem. Note that the members of a group of detected patches do not even have to be connected, so that hypotheses of objects and their positions can be generated from more global information, and this procedure can be implemented using a voting scheme, such as a generalized Hough transform or geometric hashing [9, 90, 91, 15, 14, 129].

1.4 Overview of the Thesis

In Chapter 2 we develop several algorithms for fitting implicit curves and surfaces parameterized by a finite number of continuous parameters to data. In Chapter 3 we describe algorithms which make use of the fitting methods for segmenting 2D edge maps and 3D range images. In Chapter 4 we develop a computational theory of algebraic invariants. We introduce the concept of *covariant* matrix, and show how to efficiently compute invariants of



Figure 1.5: Fourth degree interest region.

algebraic forms by performing matrix operations on covariant matrices. We consider invariants with respect to general linear transformations, and orthogonal transformations. All the invariants with respect to the general linear group are invariants with respect to the orthogonal group, but the existence of an invariant linear differential operator, the Laplacian, allows for a special treatment. In Chapter 5 we define the intrinsic frame of reference of an algebraic curve or surface, and show how groups of curves or surfaces can be aligned at once with this method, providing a new way to globally test a hypothesis supported by different regions of the data set. In Chapter 6 we show how the theory of invariants of algebraic forms of Chapter 4 can be applied to the computation of moment invariants. The availability of moments of all the orders at once let us define not only an intrinsic Euclidean coordinate system, but also an intrinsic affine coordinate system. Then the *shape polynomials* and the Δ -*distances* are introduced as tools to measure how well a shape fits as a subset of another shape, and so, obtaining a tool to globally test a hypothesis supported by several subsets of the data set. In Chapter 7 we describe how all these tools can work together to build practical recognition systems based on both algebraic surface or moment vector modeling. Finally, in Chapter 8 we present our conclusions.

Chapter 2 Implicit Curve and Surface Fitting

In this chapter we develop algorithms for fitting implicit curves and surfaces to finite data sets. Given a family of implicit functions parameterized by a finite number of parameters, and a finite set of points in space, assumed to belong to the same curve or surface, we want to estimate the parameters which minimize the mean square distance from the data points to the surface or curve defined by those parameters. Unfortunately, there is no closed form expression for the mean square distance from a data set to a generic curve or surface, and iterative methods are required to compute it.

We develop a first order approximation for the the distance from a point to a curve or surface generalizing some previous results. The mean value of this function, on a fixed set of data points, is a nonlinear function of the coefficients, but since it is a smooth function of these coefficients it can be minimized using well established non-linear least squares techniques. However, since we are interested in the global minimum of the approximate mean square distance, and these numerical techniques find only local minima, methods to chose good initial estimates are required.

In the past other researchers have minimized a different mean square error, the mean sum of squares of the values of the functions which define the curve or surface on the data points, under different constraints. It is well known that this performance function can produce a very biased result. We study the geometric conditions under which the curve or surface produced by the minimization of the mean square error fails to approximate the minimizer of the approximate mean square distance. This analysis leads us to a quadratic constraint, a function of the data, which turns the minimization of the mean square error into a stable and robust generalized eigenvector problem in the linear case, that is, when the admissible functions form a vector space. For example, algebraic curves and surfaces of arbitrary degree can be fitted with this method.

Then we introduce the reweight procedure, which in most of the cases helps to improve the solution produced by the generalized eigenvector fit at a lower cost than the general iterative minimization techniques. Finally, the result of the reweight procedure is fed into the Levenberg-Marquardt algorithm in order to minimize the approximate mean square distance.

In the case of algebraic curves and surfaces, the results of these minimization processes

enjoy the very desirable property of being invariant with respect to similarity transformations of the data set, particularly with respect to rigid body transformations. Hence, these fits are independent of the coordinate system used.

In section 2.3 we derive the approximate distance, the first order approximation to the real distance, from a point to a curve or surface. In section 2.4 we introduce the approximate square distance and develop the constraints for the linear case. In section 2.5 we study the relation between the mean square error and the approximate mean square distance, establishing the relation of our contribution with the previous work. In section 2.6 we introduce the generalized eigenvector fit method for the linear case, in appendix 2.12 we analyze the existence and uniqueness of the solution and in section 2.7 we analyze its complexity. In section 2.8 we show that the curves and surfaces produced by the generalized eigenvector fit and the minimization of the approximate mean square distance are invariant under change of basis in the linear case, and under similarity transformations in the case of algebraic curves and surfaces. In section 2.9 we introduce the reweight procedure and establish its relation with previous work. In section 2.10 we describe several families of parameterized implicit curves and surfaces, including superquadrics, where the methods introduced in this chapter can be applied. Finally, in section 2.11 we survey the previous work on implicit curve and surface fitting, establishing their relation with the methods introduced in this chapter. In the next chapter we describe a variable order algorithm for the segmentation of curves and surfaces in terms of algebraic primitives based on the techniques developed in this chapter.

2.1 Implicit curves and surfaces

Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^k$ be a *smooth* map, a map with continuous first and second order derivatives at every point. We say that the set $Z(\mathbf{f}) = \{x : \mathbf{f}(x) = 0\}$ of zeros of \mathbf{f} is defined by the *implicit equations*

$$f_1(x) = 0$$
, ..., $f_k(x) = 0$.

We are interested in three particular cases for their applications in Computer Vision and Computer Aided Design. They have special names: $Z(\mathbf{f})$ is a *planar curve* if n = 2 and k = 1, it is a *surface* if n = 3 and k = 1, and it is a *space curve* if n = 3 and k = 2. In order to avoid pathological cases, we have to require that the set of points of $Z(\mathbf{f})$ which are *regular points* of \mathbf{f} be dense in $Z(\mathbf{f})$, where a point $x \in \mathbb{R}^n$ is a *regular* point of \mathbf{f} if the Jacobian matrix

$$D\mathbf{f}(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial x_1}(x) & \cdots & \frac{\partial f_k}{\partial x_n}(x) \end{pmatrix} = \begin{pmatrix} \nabla f_1(x)^t \\ \vdots \\ \nabla f_k(x)^t \end{pmatrix}$$

has rank *k*, or equivalently if the matrix $Df(x)Df(x)^t$ is nonsingular. Otherwise, *x* is a *singular* point of **f**.

The intersection of two surfaces is a space curve. However, this representation is not unique, a space curve can be represented as the intersection of many pairs of surfaces. For

example, if $Z(\mathbf{f})$ is the intersection of two cylinders

$$\mathbf{f}(x) = \begin{pmatrix} x_1^2 + (x_3 - 1)^2 - 4 \\ x_2^2 + (x_3 + 1)^2 - 4 \end{pmatrix} = \begin{pmatrix} -3 - 2x_3 + x_3^2 + x_1^2 \\ -3 + 2x_3 + x_3^2 + x_2^2 \end{pmatrix}$$

and

$$\mathbf{g}(x) = \begin{pmatrix} -3 - 2x_3 + x_3^2 + x_1^2 \\ 4x_3 + x_2^2 - x_1^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \mathbf{f}(x) ,$$

the second component of g represents a hyperbolic paraboloid, and the sets of zeros of f and g are exactly the same Z(f) = Z(g). Figure 2.1 shows these two different representations of the same space curve. In general, for any *nonsingular* $k \times k$ matrix **A**, the function g = Af



Figure 2.1: Different representations of the same space curve.

has the same zeros as f does Z(Af) = Z(f). If g = Af for certain nonsingular $k \times k$ matrix A, we say that f and g are two *representations* of the same set Z(f). Particularly, for k = 1, the planar curve or surface Z(f) is identical to $Z(\lambda f)$, for every nonzero λ .

Unions of implicit surfaces are implicit surfaces. For example, the union of the two cylinders of the previous example

$$\{x: x_1^2 + (x_3 - 1)^2 - 4 = 0\} \cup \{x: x_2^2 + (x_3 + 1)^2 - 4 = 0\}$$

is the surface defined by the set of zeros of the product

{
$$x: (x_1^2 + (x_3 - 1)^2 - 4)(x_2^2 + (x_3 + 1)^2 - 4) = 0$$
}.

Hence, a single fourth degree polynomial can represent a pair of cylinders, and this is true for arbitrary cylinders, e.g., a pair that do not intersect. Note that although the two cylinders are regular surfaces, the points which belong to the intersection curve become singular points of

the union, the normal vector to the surface is not uniquely defined on the curve. In general, if $Z(f_1), \ldots, Z(f_k)$ are surfaces, their union is the set of zeros of the product of the functions

$$Z(f_1) \cup \cdots \cup Z(f_k) = Z(f_1 \cdots f_k),$$

with the points which belong to the intersection curve

$$\bigcup_{i \neq j} Z(f_i) \cap Z(f_j)$$

being singular points of the union. The same results hold for planar curves. The case of space curves is more complicated, but, for example, the union of two implicit space curves $Z(\mathbf{f}) \cup Z(\mathbf{g})$ is *included* in the space curve

$$\{x: f_1(x)g_1(x) = 0, f_2(x)g_2(x) = 0\}$$

2.2 Object representation, segmentation and recognition

The boundaries of most manufactured objects can be represented exactly as piecewise smooth surfaces, which in turn can be defined by implicit equations, usually algebraic surfaces.

Given a parameterized family of implicit surfaces, *image segmentation* is the problem of finding a partition of a data set into homogeneous regions, each of them well approximated by a member of the family under certain approximation criteria.

An object \mathcal{O} is a collection of surface patches

$$\mathcal{O} = \bigcup_{i=1}^q \mathcal{O}_i ,$$

where each surface patch is a regular *subset* of an implicit surface

$$\mathcal{O}_i \subseteq Z(f_i) = \{x : f_i(x) = 0\}$$
 $i = 1, \dots, q$.

A subcollection of patches, or even the whole object can be represented as a subset of a *single implicit surface*

$$\mathcal{O} \subseteq Z(f_1 \cdots f_q) = \bigcup_{i=1}^q Z(f_i).$$

The boundaries of many more objects can be *approximated* with piecewise algebraic surfaces. An object O is represented approximately as a subset of a single implicit surface Z(f), under a certain approximation criterion. The two-dimensional parallel of this representation allow us to represent sets of picture edges as subsets of a single implicit planar curve. And the set of space curves corresponding to surface normal discontinuities, or other geometric invariant curves such as the the lines of curvature of surfaces [140, 139], can be approximated by a subset of a single implicit space curve. This single implicit curve or surface will be called a *model* of O in *standard position*. Figures 2.2 and 2.2 show an attempt to recover



Figure 2.2: Degree 5 planar curve generalized eigenvector fit.



Figure 2.3: Degree 5 planar curve fit after reweight procedure.

the boundary of an object with some of the methods to be introduced in subsequent sections. The set of zeros of a single fifth degree polynomial is fitted to the data using the generalized eigenvector fit algorithm of section 2.6, and then the fit is improved with the reweight procedure of section 2.9. Although the curve does not fit the data well, because the degree is not high enough, we can appreciate how the fit is refined by the iterative reweight procedure. Since from a practical point of view, it is not convenient to work with very high degree polynomials, we see this unified representation as a way to represent small groups of smooth patches. Based on this idea, we introduce the concept of *interest region*, and we sketch a tentative approach to object recognition in a cluttered environment. For example in Figure 2.2 we show how the tip of the pliers shown in Figure 2.2 can be well approximated by a fourth degree algebraic curve, and in Figure 2.6 we show the result of fitting a single



Figure 2.4: Regions well approximated by fourth degree algebraic curves. Only the data inside the gray areas have been used in the computations, so, the approximation is good only there.

third degree algebraic surface to two visible surface patches of a pencil sharpener, a planar end and a patch of cylindrical side.

If the data set is an observation of a part of a single and known object \mathcal{O} in an unknown position, and $Z(\mathbf{f})$ is a model of \mathcal{O} in standard position, then *position estimation* becomes the problem of best fitting the data set with a member of the family of implicit curves or surfaces defined as compositions of \mathbf{f} with elements of the admissible family of transformations \mathcal{G} , because if $T \in \mathcal{G}$ is a nonsingular transformation, then

$$T^{-1}[Z(\mathbf{f})] = \{T^{-1}(y) : \mathbf{f}(y) = 0\} = \{x : \mathbf{f}(T(x)) = 0\} = Z(\mathbf{f} \circ T) ,$$



Figure 2.5: Regions well approximated by fourth degree algebraic curves. Only the data inside the gray areas have been used in the computations, so, the approximation is good only there.



Figure 2.6: The original range data, and a single third degree polynomial fit to the planar cap and patch of cylindrical side of a pencil sharpener

that is, the implicit curve or surface $Z(\mathbf{f})$ transformed by T^{-1} is a new implicit curve or surface, the curve or surface defined as the set of zeros of \mathbf{f} composed with the transformation T.

Typical examples of families of transformations are rigid body, similarity, affine and projective transformations. A detailed formulation is given in section 2.10. If the object is unknown, but it is known that it is one of a finite number of known objects modeled in standard position by the curves or surfaces $Z(\mathbf{f}_1), \ldots, Z(\mathbf{f}_q)$, then *object recognition* is equivalent to estimating the position of each object, assuming that the data corresponds to that object, and then associating the data to the object which minimizes the fitting criterion.

If the data set is a view of several objects, object recognition is equivalent to segmentation with curve or surface primitives belonging to the family of compositions of models of known objects in standard position with admissible transformations.

Two different approximation criteria will be considered in different parts of this paper, they are based on the 2-norm

$$\frac{1}{q} \sum_{i=1}^{q} \operatorname{dist}(p_i, Z(\mathbf{f}))^2$$

and the ∞ -norm

$$\sup_{1 \le i \le q} \operatorname{dist}(p_i, Z(\mathbf{f}))$$

where $\{p_1, \ldots, p_q\}$ is a finite data set, and $dist(p_i, Z(\mathbf{f}))$ is the distance from the point p_i to the curve or surface $Z(\mathbf{f})$.

We are primarily interested in fitting curves and surfaces to data under the ∞ -norm, but fitting under the 2-norm is computationally less expensive. In section 3 we describe an algorithm which follows the classical *hypothesize and test* approach, a curve or surface is *hypothesized* by minimizing an approximation to the 2-norm, and then it is *tested* with the ∞ -norm.

2.3 Approximate distance

Since a general formulation lets us study the three cases of interest at once, we will continue our analysis in this way, showing at the same time that it applies to an arbitrary dimension.

In general, the distance from a regular point $x \in \mathbb{R}^n$ of a smooth map $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^k$, to the set of zeros $Z(\mathbf{f})$, can not be computed by direct methods. The case of a linear map is an exception, in which case the Jacobian matrix $\mathbf{D} = D\mathbf{f}(x)$ is constant, and we have the identity

$$\mathbf{f}(y) \equiv \mathbf{f}(x) + \mathbf{D} (y - x) \, .$$

Without loss of generality we will assume that the rank of **D** is k. The unique point \hat{y} that minimizes the distance ||y - x|| to x, constrained by $\mathbf{f}(y) = 0$, is given by

$$\hat{y} = x - \mathbf{D}^{\dagger} \mathbf{f}(x) ,$$

where \mathbf{D}^{\dagger} is the *pseudoinverse* [48, section 9.2][66, chapter 6] of \mathbf{D} . In our case $\mathbf{D}^{\dagger} = \mathbf{D}^{t}[\mathbf{D}\mathbf{D}^{t}]^{-1}$, with the square of the distance from x to $Z(\mathbf{f})$ being

dist
$$(x, Z(\mathbf{f}))^2 = \|\hat{y} - x\|^2 = \mathbf{f}(x)^t [\mathbf{D}\mathbf{D}^t]^{-1} \mathbf{f}(x)$$

In the nonlinear case we approximate the distance from x to $Z(\mathbf{f})$ with the distance from x to the set of zeros of a *linear model* of \mathbf{f} at x, a linear map $\tilde{\mathbf{f}} : \mathbb{R}^n \to \mathbb{R}^k$ such that

$$\mathbf{f}(y) - \tilde{\mathbf{f}}(y) = O(||y - x||^2).$$

Such a map is uniquely defined when x is a regular point of f, and it is given by the truncated Taylor series expansion of f

$$\mathbf{\hat{f}}(y) = \mathbf{f}(x) + \mathbf{D}\mathbf{f}(x) (y - x)$$

Clearly $\mathbf{f}(x) = \mathbf{f}(x)$, $D\mathbf{f}(x) = D\mathbf{f}(x)$, and we have

dist
$$(x, Z(\mathbf{f}))^2 \approx \mathbf{f}(x)^t \left(\mathrm{D}\mathbf{f}(x)\mathrm{D}\mathbf{f}(x)^t \right)^{-1} \mathbf{f}(x)$$
 (2.1)

This normalization generalizes two previous results. For k = 1, the case of planar curves and surfaces, the Jacobian has only one row $Df(x) = \nabla f(x)^t$, and the right hand side member of (2.1) reduces to $f(x)^2/||\nabla f(x)||^2$, the value of the function is scaled down by the rate of growth at the point. Turner [137] and Sampson [118] have independently proposed it for particular cases of curve fitting. For k = n, the Jacobian matrix is square and nonsingular, and so the right hand side member of (2.1) reduces to $||Df(x)^{-1}f(x)||^2$, which is the length of the update of the Newton-Raphson root finding algorithm [42, chapter 5]

$$x' = x - \mathrm{D}\mathbf{f}(x)^{-1}\mathbf{f}(x) \; .$$

Our contribution is the extension to space curves, and in general to curves and surfaces of any dimension.

Since the right hand side member of (2.1) is a nonnegative number, from now on we will call

$$\sqrt{\mathbf{f}(x)^t} \left(\mathrm{D}\mathbf{f}(x)\mathrm{D}\mathbf{f}(x)^t \right)^{-1} \mathbf{f}(x)$$

the *approximate distance* from x to $Z(\mathbf{f})$. Figure 2.7 shows several contours of constant distance, constant function value and constant approximate distance for the simplest case of planar curve with a singular point, the pair of intersecting lines $\{x : x_1x_2 = 0\}$. Figure 2.8 shows the same contours for the regular curve $\{x : 8x_1^2 + (x_2^2 - 4)^2 - 32 = 0\}$. The contours of constant function value tend to be farther from the singular points and closer to the regular points than the real distance. The approximate distance solves these problems.

The approximate distance has several interesting geometric properties. It is independent of the representation of $Z(\mathbf{f})$. If \mathbf{A} is a nonsingular $k \times k$ matrix, and $\mathbf{g}(x) = \mathbf{A}\mathbf{f}(x)$, then

$$\mathbf{g}(x)^{t}[\mathrm{D}\mathbf{g}(x)\mathrm{D}\mathbf{g}(x)^{t}]^{-1}\mathbf{g}(x) = \mathbf{f}(x)^{t}\mathbf{A}^{t}[\mathrm{A}\mathrm{D}\mathbf{f}(x)\mathrm{D}\mathbf{f}(x)^{t}\mathbf{A}^{t}]^{-1}\mathbf{A}\mathbf{f}(x)$$

= $\mathbf{f}(x)^{t}[\mathrm{D}\mathbf{f}(x)\mathrm{D}\mathbf{f}(x)^{t}]^{-1}\mathbf{f}(x)$. (2.2)



Figure 2.7: Contours of constant distance, constant function value and constant approximate distance to the curve $\{x : x_1x_2 = 0\}$ near a singular point



Figure 2.8: Contours of constant distance, constant function value and constant approximate distance to the curve $\{x : 8x_1^2 + (x_2^2 - 4)^2 - 32 = 0\}$ near a regular curve

It is also invariant to rigid body transformations of the space variables, if T(x) = Qx + b is a rigid body transformation, then D[f(Qx + b)] = Df(Qx + b)Q and so

$$D[\mathbf{f}(Qx+b)]D[\mathbf{f}(Qx+b)]^t = D\mathbf{f}(Qx+b)QQ^tD\mathbf{f}(Qx+b)^t = D\mathbf{f}(Qx+b)D\mathbf{f}(Qx+b)^t$$

A similar derivation shows that a scale transformation of the space variables produces the corresponding scaling of the approximate distance.

Since we are interested in fitting curves and surfaces to data in a finite number of steps, we will restrict ourselves to families of maps described by a finite number of parameters. Let us fix a smooth function $\phi : \mathbb{R}^{r+n} \to \mathbb{R}^k$ defined almost everywhere. From now on we will only consider maps $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^k$ which can be written as

$$\mathbf{f}(x) \equiv \phi(\alpha, x) ,$$

for certain $\alpha = (\alpha_1, \ldots, \alpha_r)^t$, in which case we will also write $\mathbf{f} = \phi_{\alpha}$. We will refer to $\alpha_1, \ldots, \alpha_r$ as the *parameters* and to x_1, \ldots, x_n as the *variables*. The family of all such maps will be denoted

$$\mathcal{F} = \{ \mathbf{f} : \exists \alpha \ \mathbf{f} = \phi_{\alpha} \} ,$$

we will say that ϕ is the *parameterization* of the family \mathcal{F} , and we will write \mathcal{F}_{ϕ} when having to differentiate among different parameterizations. The approximate distance from x to $Z(\phi_{\alpha})$ will be denoted

$$\delta(\alpha, x) = \sqrt{\phi_{\alpha}(x)^{t} [\mathrm{D}\phi_{\alpha}(x)\mathrm{D}\phi_{\alpha}(x)^{t}]^{-1} \phi_{\alpha}(x)} ,$$

or $\delta_{\phi}(\alpha, x)$ when needed.

We will give special attention to the *linear parameterization*, where $\phi(\alpha, x)$ is a linear function of α , and so \mathcal{F} is a finite dimensional vector space of smooth maps $\mathbb{R}^n \to \mathbb{R}^k$. We will refer to this case as the *linear case*. In the linear case, a map f belongs to \mathcal{F} if and only if for every nonsingular $k \times k$ matrix \mathbf{A} , \mathbf{Af} also belongs to \mathcal{F} . Vector spaces of polynomials of degree $\leq d$ are typical examples of linearly parameterized families. In section 2.10 we show several nonlinear parameterizations of families of curves and surfaces with applications in Computer Vision.

2.4 Approximate mean square distance

Let $\mathcal{D} = \{p_1, \ldots, p_q\}$ be a set of *n*-dimensional data points, and let $Z(\mathbf{f})$ be the set of zeros of $\mathbf{f} = \phi_{\alpha} : \mathbb{R}^n \to \mathbb{R}^k$. If we assume that α is known and $\delta(\alpha, p_1)^2, \ldots, \delta(\alpha, p_q)^2$, are independent and uniformly distributed as the square of a normal random variable of mean zero and variance σ^2 , the sum

$$\frac{1}{\sigma^2} \sum_{i=1}^q \delta(\alpha, p_i)^2 \tag{2.3}$$

has a χ^2 distribution with *q* degrees of freedom.

If the true α is unknown, it can be estimated minimizing (2.3). Curve or surface fitting corresponds to the minimization of (2.3) with respect to the unknown parameters of $\alpha_1, \ldots, \alpha_r$.

Assuming that the variance σ^2 is known, the problem is equivalent to minimizing the *approximate mean square distance* from the data set \mathcal{D} to the set of zeros of $\mathbf{f} = \phi_{\alpha}$

$$\Delta_{\mathcal{D}}^2(\alpha) = \frac{1}{q} \sum_{i=1}^q \delta(\alpha, p_i)^2$$
(2.4)

Depending on the particular parameterization ϕ , the *r* parameters $\alpha_1, \ldots, \alpha_r$ might not be independent. For example, by (2.2) the sum does not change if we replace **Af** for **f**, where **A** is a nonsingular $k \times k$ matrix, and if **f** minimizes the approximate mean square distance (2.4) so does **Af**. In other words, the parameters might not be *identifiable*. This lack of identifiability is not a problem though, because we are not interested in the function **f**, but in its set of zeros $Z(\mathbf{f})$. For example, in the linear case the symmetric matrix constraint

$$\frac{1}{q} \sum_{i=1}^{q} \mathrm{D}\mathbf{f}(p_i) \mathrm{D}\mathbf{f}(p_i)^t = \mathbf{I}_k , \qquad (2.5)$$

equivalent to k(k + 1)/2 scalar constraints on the parameters, can be imposed on **f** without affecting the set of zeros of a minimizer of (2.4). The reason is that $\Delta_{\mathcal{D}}^2(\alpha)$ is defined only if the symmetric matrices

$$\mathrm{D}\mathbf{f}(p_1)\mathrm{D}\mathbf{f}(p_1)^t$$
, ..., $\mathrm{D}\mathbf{f}(p_q)\mathrm{D}\mathbf{f}(p_q)^t$

are nonsingular, and since all of these matrices are also nonnegative definite, they are positive definite, and their mean

$$\frac{1}{q} \sum_{i=1}^{q} \mathrm{D}\mathbf{f}(p_i) \mathrm{D}\mathbf{f}(p_i)^t$$
(2.6)

is positive definite as well, in which case there exists a nonsingular $k \times k$ matrix **A** such that

$$\mathbf{I}_{k} = \mathbf{A}\left(\frac{1}{q}\sum_{i=1}^{q}\mathbf{D}\mathbf{f}(p_{i})\mathbf{D}\mathbf{f}(p_{i})^{t}\right)\mathbf{A}^{t} = \frac{1}{q}\sum_{i=1}^{q}\mathbf{D}[\mathbf{A}\mathbf{f}](p_{i})\mathbf{D}[\mathbf{A}\mathbf{f}](p_{i})^{t},$$

and Af satisfies the constraint (2.5). We can take A as the inverse of the Cholesky decomposition of (2.6). Since f belongs to \mathcal{F} if and only if Af does, the constraint (2.5) does not impose any restriction on the set of admissible curves or surfaces $\{Z(f) : f \in \mathcal{F}\}$.

The problem of computing a *local minimum* of an expression like (2.4) is known as the nonlinear least squares problem, and it can be solved using several iterative methods [42, chapter 10]. Among them, the Levenberg-Marquardt algorithm [92, 99] is probably the best known, and excellent implementations of it are available in subroutine packages such as MINPACK [100]. A short description of the Levenberg-Marquardt algorithm in the context of our problem is given in Appendix A.

Every local minimization algorithm requires a good starting point. Since we are interested in the *global minimization* of (2.4), even using the Levenberg-Marquardt algorithm we need a method to chose a good initial estimate.

2.5 Mean square error

The study of $\Delta_{\mathcal{D}}^2(\alpha)$ in a particular case will provide us with a good strategy to choose an initial estimate in certain cases, such as, for example, in the linear case. Let us assume that the matrix function $Df(x)Df(x)^t$ is *constant* on the set Z(f), in particular Z(f) does not have singular points. For k = 1 this means that the length of the gradient of the unique component f_1 of f is constant on Z(f), but nothing is said about its orientation. Linear functions obviously have this property because in this case Df(x) is already constant. But circles, spheres and cylinders, among other families of functions, have the same property. If f also satisfies the constraint (2.5) and the data points are close to the set of zeros of f, by continuity of Df we have

$$\mathbf{I}_{k} = \frac{1}{q} \sum_{i=1}^{q} \mathbf{D} \mathbf{f}(p_{i}) \mathbf{D} \mathbf{f}(p_{i})^{t} \approx \mathbf{D} \mathbf{f}(p_{j}) \mathbf{D} \mathbf{f}(p_{j})^{t} \qquad j = 1, \dots, q$$

and the approximate mean square distance $\Delta_{\mathcal{D}}^2(\alpha)$ is approximated by the *mean square error*

$$\xi_{\mathcal{D}}^{2}(\alpha) = \frac{1}{q} \sum_{i=1}^{q} \|\mathbf{f}(p_{i})\|^{2} .$$
(2.7)

In this particular case, when $DfDf^t$ is constant on Z(f), the minimizers of (2.7) and (2.4), both constrained by (2.5) are almost the same, and we will see in the following section that in the linear case the global minimizer of (2.7)-(2.5) can be computed at a much lower cost than a local minimizer of (2.4)-(2.5).

Furthermore, we have observed that, if **f** is the global minimizer of $\Delta_{\mathcal{D}}^2(\alpha)$, and the matrix $D\hat{f}D\hat{f}^t$ is not close to a singular matrix on \mathcal{D} , the minimizer of $\xi_{\mathcal{D}}^2(\alpha)$ constrained by (2.5) is a very good approximation of \hat{f} , and the Levenberg-Marquardt algorithm started from this estimate converges quickly after a few iterations. Geometrically, $D\hat{f}D\hat{f}^t$ not close to a singular matrix means that no point of \mathcal{D} is close to a singular point of \hat{f} . Since \hat{f} is unknown, we can not test beforehand whether $D\hat{f}D\hat{f}^t$ is close to a singular matrix or not. In order to speed up the convergence of the Levenberg-Marquardt algorithm, after computing the minimizer of the mean square error, and before the iterative minimization, we apply the *reweight procedure*, which is described after the analysis of the linear case, in section 2.9.

Most of the previous work on fitting implicit curves and surfaces to data has been based on minimizing the mean square error, but with different constraints. In section 2.11 we give a detailed description of these earlier methods.

2.6 Generalized eigenvector fit

In this section we show that in the linear model, the minimization of the mean square error (2.7) constrained by (2.5) reduces to a generalized eigenvector problem. In section 2.11 we show that this method, introduced by us [127, 128], generalizes several earlier eigenvector fit methods.

Let $X_1(x), \ldots, X_h(x)$ be linearly independent smooth functions, for example polynomials, and let us denote

$$X = (X_1, \dots, X_h)^t : \mathbb{R}^n \to \mathbb{R}^h$$

In this section all the maps can be written as linear combinations of the components of X,

$$\mathbf{f} = FX : \mathbb{R}^n \to \mathbb{R}^k$$

for a $k \times h$ matrix F of real numbers. The parameter vector α has r = hk elements, and it is equal to the concatenation of the rows of F

$$F_{ij} = \alpha_{(i-1)h+j}$$
 $i = 1, ..., k$ $j = 1, ..., h$.

Since differentiation is a linear operation, we have

$$D\mathbf{f} = D[FX] = F[DX],$$

where DX is the Jacobian of X. The constraint (2.5) become a quadratic constraint on the elements of F

$$I_{k} = \frac{1}{q} \sum_{i=1}^{q} F[DX(p_{i})][DX(p_{i})^{t}]F^{t} = FN_{\mathcal{D}}F^{t}$$
(2.8)

where

$$N_{\mathcal{D}} = \frac{1}{q} \sum_{i=1}^{q} [\mathrm{D}X(p_i)\mathrm{D}X(p_i)^t]$$

is symmetric nonnegative definite. The approximate mean square distance $\Delta_{\mathcal{D}}^2(\alpha)$ does not have any special form, but the mean square error (2.7) becomes

$$\begin{aligned} \xi_{\mathcal{D}}^{2}(\alpha) &= \frac{1}{q} \sum_{i=1}^{q} \|FX(p_{i})\|^{2} \\ &= \frac{1}{q} \sum_{i=1}^{q} \operatorname{trace} \left(F[X(p_{i})X(p_{i})^{t}]F^{t} \right) \\ &= \frac{1}{q} \operatorname{trace} \left(FM_{\mathcal{D}}F^{t} \right) , \end{aligned}$$

$$(2.9)$$

where

$$M_{\mathcal{D}} = \frac{1}{q} \sum_{i=1}^{q} [X(p_i)X(p_i)^t] ,$$

the covariance matrix of X over the data set D. This matrix is classically associated with the normal equations of the least square method [66, chapter 6]. And several researchers have introduced linear or quadratic constraints on F to fit implicit curves or surfaces, k = 1, to data [4, 13, 20, 33, 36, 37, 64, 104, 103, 110, 118] minimizing (2.9). However, all of these constraints do not take into account the data, they are fixed and predetermined constraints on the coefficients, and most of them introduce singularities in parameter space, i.e., certain parameters are *never* solutions of the corresponding method. A detailed description of these methods is given in section 2.11 below. Our contribution is the introduction of the quadratic

constraint (2.8) which is function of the data, and the handling of space curves within the same framework. The generalized eigenvector fit algorithm is more robust than most of the previous direct methods, except perhaps for Pratt's *simple fit* algorithm [110], explained in section 2.11, which seems to be equivalent both in computational cost and robustness. We plan to carry out a detailed comparison of the generalized eigenvector fit and the simple fit algorithms in the near future.

Note that if $M_{\mathcal{D}}$ is singular and a row F_j of the matrix F belongs to the null space of $M_{\mathcal{D}}$, then

$$0 = F_j M_{\mathcal{D}} F_j^t = \frac{1}{q} \sum_{i=1}^q |F_j X(p_i)|^2$$

and the function $f_j = F_j X$ is identically zero on \mathcal{D} , in which case f_j *interpolates* all the data set. If

$$1 = F_j N_{\mathcal{D}} F_j^t = \frac{1}{q} \sum_{i=1}^{q} \|\nabla f_j(p_i)\|^2$$

as well, then F_j has to be a row of the minimizer matrix \hat{F} of (2.9)-(2.8), when such a minimizer exists. In appendix 2.12 we analyze the existence and uniqueness of solution and show that, if N_D is positive definite and $\hat{F}_1, \ldots, \hat{F}_k$ are the eigenvectors of the symmetric-positive pencil $M_D - \lambda N_D$ corresponding to the least k eigenvalues $0 \le \lambda_1 \le \cdots \lambda_k$,

$$F_i M_{\mathcal{D}} = \lambda_i F_i N_{\mathcal{D}} \qquad i = 1, \dots, k$$
$$\hat{F}_i N_{\mathcal{D}} \hat{F}_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad i, j = 1, \dots, k$$

then \hat{F} , the matrix with rows $\hat{F}_1, \ldots, \hat{F}_k$, is a minimizer of (2.9)-(2.8).

In general, the matrix N_D is rank-deficient, but the problem can be reduced to an equivalent positive-definite one of size equal to the rank of N_D . A minimizer exists if and only if the rank of N_D is at least k, and the solution is unique if $k = \operatorname{rank}(N_D)$ or, if $k < \operatorname{rank}(N_D)$ and $\lambda_k < \lambda_{k+1}$.

We have already shown examples of generalized eigenvector planar curve fit in figures 2.2,2.2, 2.2, and 2.2. Figure 2.6 shows an example of generalized eigenvector nonplanar space curve fit, the intersection of two cylinders. Figure 2.10 shows an example of algebraic fourth degree surface fit. The original surface is the surface of revolution generated by the curve of figure 2.8, and the solution is the set of zeros of a general fourth degree polynomial with 35 coefficients. Figure 2.11 shows another example of generalized eigenvector space curve fit, again the intersection of two surfaces, where the solution fits the data very well, but it is very different from the original curve elsewhere. In general, even if the solution fits the data well, we can not expect to reconstruct the same original curve or surface, the curve or surface the data is a sample of. Depending on the amount of noise and the extent of the original curve or surface sampled by the data, the generalized eigenvector fit can produce this kind of solution or not. It is less likely to occur if the reweight procedure and the Levenberg-Marquardt algorithms are used, and the solution is tested after each of the three steps.



Figure 2.9: Generalized eigenvector space curve fit. The solution is the intersection of two unconstrained quadrics.



Figure 2.10: Fourth degree algebraic surface fit.



Figure 2.11: Generalized eigenvector space curve fit. The solution is the intersection of two unconstrained quadrics.
2.7 Computational cost

The complexity of evaluating the matrices M_D and N_D depends on the functions which constitute the vector X. In the case of polynomials, when X is the vector of monomials of degree $\leq d$, the elements of M_D are moments of degree $\leq 2d$ and the elements of N_D are integral combinations of at most n moments of degree $\leq 2(d-1)$. For example, for polynomials of degree two in three variables we have

Other basis vectors, such as the Bernstein basis are more expensive to evaluate, but are potentially more stable [54, 53]. The solution is independent of the basis though, as we explain in section 2.8. We circumvent the stability problem by using centered and scaled monomials, also explained in section 2.8. The *center* of the data set is its mean value, and the *scale* is the square root of the mean square distance to the center. The computation of the center and scale only require the moments of degree ≤ 2 , the moments used for the eigenvector line or planar fit.

Computing the matrices from their definitions require more operations than computing the vector of moments of degree $\leq 2d$, and then filling the matrices by table look up. The tables are computed off-line. The vector of monomials of degree $\leq 2d$ has $s = \binom{2d+n}{n} = O(h)$ components, and can be evaluated by exactly s - n - 1 multiplications, where $h = \binom{d+n}{n} = O(d^n)$ is the number of components of X, the size of the matrices. It follows that q(s-n-1) multiplications and (q-1)(s-n-1) additions are required to evaluate the moments, and then h(h+1)/2 operations to fill M_D and at most nh(h+1) operations to fill N_D . The total number of operations required to build the matrices is $O(qh + nh^2)$.

An algorithm which computes all the eigenvalues and eigenvectors is given by Golub and Van Loan [66], requiring about $7h^3$ *flops*, where *h* is the order of the matrices, and a *flop* roughly constitutes the effort of doing a floating point add, a floating point multiply, and a little subscripting. That algorithm uses the symmetric *QR* algorithm to compute all the eigenvalues and corresponding eigenvectors. When all the eigenvectors are computed the symmetric *QR* algorithm requires about $5h^3$ flops.

Since we only need to compute a few eigenvalues and eigenvectors, we use the alternative methods implemented in EISPACK [122, 60], that is, tridiagonalization, the QR algorithm and inverse iteration. Tridiagonalization of a symmetric matrix requires about $2/3h^3$ flops. The computation of each eigenvalue using the QR algorithm without computing eigenvectors requires about 5h flops per eigenvalue. When the eigenvalues are well separated from each other, as is generally the case in our matrices due to measurement errors and noise, only one inverse iteration is required to compute an eigenvector. Each inverse iteration requires about 4h flops. From all this analysis we conclude that the proposed algorithm requires about $3h^3$ flops. The alternative implicit curve and surface fitting algorithms all have the same order of complexity [33, 110].

2.8 Independence and invariance

The solution produced by the generalized eigenvector fit method is independent of the basis. In particular, it is independent of the order of the elements of a particular basis. If Y_1, \ldots, Y_h is another basis of the vector space spanned by X_1, \ldots, X_h , and we denote $Y = (Y_1, \ldots, Y_h)^t$, then there exists a nonsingular $h \times h$ matrix A such that X = AY. Every admissible function can be written as $\mathbf{f}(x) = FX = [FA]Y$, and

$$FM_{\mathcal{D}}F^{t} = F\left(\frac{1}{q}\sum_{i=1}^{q} [X(p_{i})X(p_{i})^{t}]\right)F^{t} = [FA]\left(\frac{1}{q}\sum_{i=1}^{q} [Y(p_{i})Y(p_{i})^{t}]\right)[FA]^{t}$$

with the corresponding identity for $N_{\mathcal{D}}$, because, by linearity of differentiation

$$D[FX] = FD[AY] = [FA]DY$$

It follows that F solves the generalized eigenvector fit problem with respect to the basis X if and only if [FA] solves the same problem but with respect to the basis Y. Since the approximate distance, and a fortiori the approximate mean square distance, are clearly independent of the basis, the minimizer of the approximate mean square distance is independent of the basis too.

If **f** is a solution of the generalized eigenvector fit or a minimizer of the approximate mean square distance to the data set $\mathcal{D} = \{p_1, \ldots, p_q\}$, i.e., the curve or surface $Z(\mathbf{f})$ best fits the data set \mathcal{D} , we want to know for which families of nonsingular transformations Tthe transformed curve or surface $T^{-1}[Z(\mathbf{f})] = Z(\mathbf{f} \circ T)$ best fits the transformed data set $T^{-1}[\mathcal{D}] = \{T^{-1}(p_1), \ldots, T^{-1}(p_q)\}$.

If the vector space spanned by X is the space of polynomials of maximum degree d, then the solution of the generalized eigenvector fit is invariant with respect to similarity transformations. If T(x) = Ax + b is an nonsingular affine transformation and f(x) is a polynomial, the composition f(T(x)) is a polynomial of the same degree, whose coefficients are polynomials in A, b, and the coefficients of f(x). If the components of X form a basis of the vector space of polynomials of degree $\leq d$, then so do the components of Y(x) = X(T(x)), because the transformation T is nonsingular. It follows that there exists an $h \times h$ nonsingular matrix T^* , whose coefficients are polynomials of degree $\leq d$ in A and b such that $X(T(x)) \equiv T^*X(x)$ is a polynomial identity [141, chapter III,§4]. Furthermore, the map $T \mapsto T^*$ defines a faithfull representation of the n-dimensional affine group in the h-dimensional general linear group, a 1 - 1 homomorphism of groups, which in particular satisfies $[T^{-1}]^* = [T^*]^{-1}$. We study in deeper detail a closely related representation in the plane

$$T(x) = \begin{pmatrix} x_1 + b_1 \\ x_2 + b_2 \end{pmatrix} ,$$

and X is the vector of monomials of degree ≤ 2 in two variables

$$X = (1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_1 x_2 \quad x_2^2)^t ,$$

then

$$X(T(x)) = \begin{pmatrix} 1 \\ x_1 + b_1 \\ x_2 + b_2 \\ (x_1 + b_1)^2 \\ (x_1 + b_1)(x_2 + b_2) \\ (x_2 + b_2)^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ b_1 & 1 & 0 & 0 & 0 & 0 \\ b_2 & 0 & 1 & 0 & 0 & 0 \\ b_1^2 & 2b_1 & 0 & 1 & 0 & 0 \\ b_1b_2 & b_2 & b_1 & 0 & 1 & 0 \\ b_2^2 & 0 & 2b_2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{pmatrix} = T^*X(x)$$

If $\mathbf{f} = FX$ is a solution of the generalized eigenvector fit for the data set \mathcal{D} , and T is a similarity transformation, that is, $A = \lambda Q$, with Q orthogonal and λ a nonzero constant, then $\mathbf{g} = \mathbf{f} \circ T = [FT^*]X$ is the solution of the same problem for the data set $T^{-1}[\mathcal{D}]$, because

$$M_{T^{-1}[\mathcal{D}]} = \frac{1}{q} \sum_{i=1}^{q} [X(T^{-1}(p_i))X(T^{-1}(p_i))^t]$$

= $\frac{1}{q} \sum_{i=1}^{q} [[T^{-1}]^*X(p_i)][[T^{-1}]^*X(p_i)]^t = [T^*]^{-1}M_{\mathcal{D}}[T^*]^{-t}$

which implies that

$$[FT^{\star}]M_{T^{-1}[\mathcal{D}]}[FT^{\star}]^{t} = FM_{\mathcal{D}}F^{t}$$

and with respect to the matrix $N_{\mathcal{D}}$, by the chain rule

$$\mathrm{D}Y \ = \ \mathrm{D}[X(T(x)] \ = \ [\mathrm{D}X](T(x)) \cdot \mathrm{D}[T(x)] \ = \ T^{\star} \,\mathrm{D}X \,A \ = \ \lambda T^{\star} \,\mathrm{D}X \,Q,$$

which, since Q is orthogonal, implies that

$$N_{T^{-1}[\mathcal{D}]} = \lambda^2 [T^{\star}]^{-1} N_{\mathcal{D}} [T^{\star}]^{-t} ,$$

and so

$$[FT^{\star}]N_{T^{-1}[\mathcal{D}]}[FT^{\star}]^{t} = \lambda^{2} F N_{\mathcal{D}} F^{t} .$$

The factor λ^2 in the constraint equation does not change the solution of the problem.

A similar result holds for the minimizer of the approximate mean square distance (2.7), but we omit the proof.

Figures 2.8 and 2.8 shows an example of generalized eigenvector fit of sixth degree algebraic curves to two observations of the same object in different positions. The curves are slightly different because so are the data sets.

The matrix M_D could be very badly conditioned when X is the vector of monomials of degree $\leq d$. The matrix N_D is generally better conditioned than M_D , but not too much. When the matrices are poorly conditioned the results are not accurate, or even totally wrong. Even worse, the eigenvalue computation can diverge. Based on the results of the current section, we would like to find a new basis Y = AX for the polynomials of degree $\leq d$, such that the matrices computed with respect to this new basis are well conditioned. Such a basis has to be function of the data set. One of such basis is the Bernstein basis [54, 53]. Since the evaluation of the Bernstein basis has a higher complexity than the power basis, we



Figure 2.12: Invariance to translation and rotation. (a): Data set A. (b): Degree 6 generalized eigenvector fit to data set A.



Figure 2.13: Invariance to translation and rotation. (a): Data set B. (b): Degree 6 generalized eigenvector fit to data set B.

follow an alternative approach that is usually found in the statistical literature [38, 25]. The *center* of the data set is its mean value

$$\bar{p} = \frac{1}{q} \sum_{i=1}^{q} p_i$$

and its *scale* is the square root of the mean square distance to the center

$$\sigma^2 = \frac{1}{q} \sum_{i=1}^{q} \|p_i - \bar{p}\|^2 = \operatorname{trace} \left(\frac{1}{q} \sum_{i=1}^{q} [p_i \, p_i^t] - \bar{p} \, \bar{p}^t \right) \,.$$

If T is the similarity transformation $T(x) = \sigma \cdot x + \bar{p}$, then solving the generalized eigenvector fit or minimizing the approximate mean square distance with respect to the original data set \mathcal{D} is equivalent to solving the corresponding problem using the power basis with respect to the transformed set $T^{-1}[\mathcal{D}]$, and this is the proper way to implement this algorithm. First the center and scale are computed, then the data set is transformed and the corresponding fitting problem is solved using the power basis, finally the coefficients are backtransformed using T^* . It is even better not to backtransform the polynomial \mathbf{g} , the solution of the transformed problem, and evaluate the solution of the original problem \mathbf{f} in two steps

1.
$$x' = \frac{1}{\sigma}(x-\bar{p})$$

2. $f(x) = g(x')$

when such an evaluation becomes necessary.

2.9 The reweight procedure

Let $\mathbf{f} = \phi_{\alpha}$ and w_1, \ldots, w_q be positive numbers. Let us call

$$\xi_{\mathcal{D},w}^{2}(\alpha) = \frac{1}{q} \sum_{i=1}^{q} w_{i} \|\mathbf{f}(p_{i})\|^{2}$$
(2.10)

the weighted mean square error. For planar curves and surfaces, k = 1, and $w_i = 1/||\nabla \mathbf{f}(p_i)||^2$, the weighted mean square error reduces to the approximate mean square distance. The point p_i is given a heavy weight only if it is close to a singular point of $Z(\mathbf{f})$. Note that if $\hat{\mathbf{f}}$ is a minimizer of the approximate mean square distance constrained by (2.5), and p_i is close to a singular point of $Z(\hat{\mathbf{f}})$, then both $||\hat{\mathbf{f}}(p_i)||^2 \approx 0$ and $||\nabla \hat{\mathbf{f}}(p_i)||^2 \approx 0$, and the contributions of p_i to both the mean square error (2.7) and the constraint (2.5) are negligible. Minimizing the mean square error, instead of the approximate mean square distance, is like fitting a curve or surface to the data set, but without considering the points which are close to singularities of $Z(\hat{\mathbf{f}})$. If the minimizer of the approximate mean square distance has singular points, we can not expect them to be well approximated by the set of zeros of the curve or surface solution of the generalized eigenvector fit problem.

In general we take

$$w_i(\alpha) = \operatorname{trace}\left([\operatorname{Df}(p_i)\operatorname{Df}(p_i)^t]^{-1}\right) ,$$

and based on the same arguments used in section 2.4, we impose the constraint

$$\frac{1}{q} \sum_{i=1}^{q} w_i \left(\mathrm{D}\mathbf{f}(p_i) \mathrm{D}\mathbf{f}(p_i)^t \right) = \mathbf{I}_k$$
(2.11)

without modifying the set of zeros of the minimizer of the approximate mean square distance.

Now, if we keep $w = (w_1, \ldots, w_q)^t$ fixed, and we stay within the linear model $\mathbf{f}(x) = FX(x)$, the minimization of (2.10) constrained by (2.11) is a generalized eigenproblem, as in the previous section. In this case we minimize $\operatorname{trace}(FM_{\mathcal{D},w}F^t)$ constrained by $FN_{\mathcal{D},w}F^t = 1$, where

$$M_{\mathcal{D},w} = \frac{1}{q} \sum_{i=1}^{q} w_i \left(X(p_i) X(p_i)^t \right)$$
$$N_{\mathcal{D},w} = \frac{1}{q} \sum_{i=1}^{q} w_i \left(DX(p_i) DX(p_i)^t \right)$$

If **f** is a minimizer of the approximate mean square distance, and the weighs w_1, \ldots, w_q are close to $w_1(\hat{\mathbf{f}}), \ldots, w_q(\hat{\mathbf{f}})$, by continuity the minimizer of the weighted linear problem is close to $\hat{\mathbf{f}}$. This property suggests the *reweight procedure*, described in Figure 2.14, where ϵ is a small positive constant which controls when to stop the loop.

```
procedure Reweight (F, D)

F' := F

do

F := F'

w := w(F)

F' := minimizer of trace(F^t M_{D,w}F)

constrained by F^t N_{D,w}F = I_k

while \Delta^2_D(F') < (1 - \epsilon)\Delta^2_D(F)

if \Delta^2_D(F') < \Delta^2_D(F) then

return(F')

else

return(F)
```

Figure 2.14: The reweight procedure for the generalized eigenvector fit.

In the linear model the initial value of F will be the solution produced by the generalized eigenvector fit with uniform weights, the case of the previous section. In a practical implementation, and in order to save time, the reweight procedure would be called only if this initial value of F does not pass a *goodness of fit* test. At each iteration we solve a generalized eigenproblem and then recompute the weights. We continue doing this while the approximate mean square distance decreases. Then, if the value of F returned by the reweight

procedure does not pass the same goodness of fit test we call the Levenberg-Marquardt algorithm.

The reweight procedure is similar in spirit to Sampson's algorithm [118] and those iterative weighted least squares algorithms that appear in the regression literature [111]. There is no certainty of convergence though, and according to Sampson, the system of equations is so complex that it would be extremely difficult to determine the conditions under which such a scheme converges [84]. These statement agrees with what we have observed during the course of this study.

We have already shown in figures 2.2 and 2.2 how the reweight procedure improves the result of the generalized eigenvector fit, even though the final result would not be accepted by the goodness of fit test. In figure 2.15 we show how the reweight procedure improves the fitting curve in a case where the result provides an acceptable approximation of the data.



Figure 2.15: Reweight procedure improving generalized eigenvector fit. (a): generalized eigenvector fit. (b): after reweight procedure.

2.10 Other families of curves and surfaces

In this section we consider several potential applications of the methods introduced in this paper.

In certain cases, such as in the family of cylinders or superquadrics, the parameters can be divided in two groups : *shape* and *positional* parameters. The radius of a cylinder is a shape parameter and all the other parameters, which describe its axis, are positional parameters. In the family of polynomials of a given maximum degree all the parameters are shape parameters, because the composition of a polynomial with a nonsingular affine transformation is a polynomial of the same degree.

It is particularly important to consider the case in which all the parameters are positional parameters, the case of a family of compositions of a fixed map g with a parameterized

family of transformations G

$$\mathcal{F} = \{ \mathbf{f} : \exists T \in \mathcal{G} \ \mathbf{f}(x) \equiv \mathbf{g}(T(x)) \} ,$$

for its applications to object recognition and position estimation.

2.10.1 Transformations of a curve or surface

A family of transformations, for example rigid body, affine or projective transformations, can be given in parametric form

$$\mathcal{G} = \{T_{\alpha} : \alpha \in \mathbb{R}^r\}$$

in which case the parameterization of the admissible functions is

$$\phi(\alpha, x) = \mathbf{g}(T_{\alpha}(x)) ,$$

where $\mathbf{g}(x)$ is a fixed map whose set of zeros we will call the *model in standard position*. Explicit parameterizations of some of these families are discussed below in this section. The problem of fitting a member of the family $\mathcal{F} = \{\mathbf{f} : \exists \alpha \ \mathbf{f}(x) \equiv \mathbf{g}(T_{\alpha}(x))\}$ to a set of data points $\mathcal{D} = \{p_1, \ldots, p_q\}$ is position estimation or *generalized template matching* : we assume that the data belong to a single object, and that this object is an observation of the model not necessarily in the standard position, and possibly partially occluded. We minimize $\Delta_{\mathcal{D}}^2(\alpha)$, the approximate mean square distance from the data points to the model $Z(\mathbf{g})$ transformed to the position and orientation defined by the transformation T_{α}

$$Z(\mathbf{g} \circ T_{\alpha}) = \{ x : \mathbf{g}(T_{\alpha}(x)) = 0 \} = \{ T_{\alpha}^{-1}(y) : \mathbf{g}(y) = 0 \} = T_{\alpha}^{-1}[Z(\mathbf{g})]$$

This method was used by Cooper, Hung and Taubin [35] to locate surfaces matching curves in the context of the stereo problem. Figure 2.16 shows a simple example of planar curve position estimation within this framework. How to choose initial estimates for the minimization of the approximate mean square distance is the remaining problem of this formulation, particularly when we have to deal with very complex objects or occlusions, and it is the subject of subsequent chapters.

2.10.2 Projection of space curves onto the plane

The methods for the minimization of the approximate mean square distance introduced in this paper can be used to improve Ponce and Kriegman's algorithm [109] for object recognition and positioning of 3D objects from image contours.

In their formulation object models consist of collections of parametric surface patches and their intersection curves. The image contours considered are the projections of surface discontinuities and occluding contours. They use elimination theory [47, 65, 97, 117, 120, 27] for constructing the implicit equation of the image contours of an object observed under orthographic, weak perspective or perspective projection. The equation is parameterized by the position and orientation of the object with respect to the observer.



Figure 2.16: Simple position estimation.

Note that object models defined by implicit curves and surfaces can be handled using the same techniques. For example, under orthographic projection the implicit equation of the projection of an algebraic curve $Z(\mathbf{f})$ onto the plane can be computed eliminating the variable x_3 from the pair $(f_1(x), f_2(x))$, and since the occluding contour of a surface Z(f) is the curve defined as the intersection of Z(f) with $Z(\partial f/\partial x_3)$, the projection of the occluding contour is in this case the *discriminant* of f with respect to z_3 [141, chapter I, §9].

They use more complex iterative methods to compute the distance from a point to the hypothesized curve than the approximate distance, and so, it becomes much more expensive to minimize their approximation to the mean square distance than in our case.

2.10.3 Parameterizations of some transformation groups

The general rigid body transformation is T(x) = Qx+b, where Q is a rotation matrix, an orthogonal matrix of unitary determinant, and b is a n-dimensional vector. The general affine transformation can be written as usual as T(x) = Ax + b, where A is a $n \times n$ nonsingular matrix and b is a n-dimensional vector. Alternatively, we can write it as T(x) = LQx + b, where L is a nonsingular lower triangular matrix and Q is a rotation matrix.

The usual parameterization of a general rotation matrix is trigonometric

$$\begin{pmatrix} \cos\theta_1 & \sin\theta_1 & 0\\ -\sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & 0 & \sin\theta_2\\ 0 & 1 & 0\\ -\sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0\\ 0 & \cos\theta_3 & \sin\theta_3\\ 0 & -\sin\theta_3 & \cos\theta_3 \end{pmatrix} ,$$

where θ_1 , θ_2 , θ_3 are the so called Euler angles. Since we have to iterate on the parameters, we propose the alternative rational parameterization, originally due to Cayley [143, chapter II, B.10], and which is valid in any dimension.

A square matrix Q is called *exceptional* if

$$|I+Q| = 0,$$

and *non-exceptional* otherwise, where I is the identity matrix. That is, a matrix es exceptional if it has -1 as an eigenvalue. If the matrix Q is non-exceptional, let us consider the matrix

$$U = (I - Q)(I + Q)^{-1} = (I + Q)^{-1}(I - Q).$$

The matrix U so defined is non-exceptional as well, because

$$|I + U| = |(I + Q)^{-1} ((I + Q) + (I - Q))| = 2 |I + Q|^{-1},$$

and the matrix Q can be recovered from U in the same way, because

$$I - U = (((I + Q) - (I - Q))(I + Q)^{-1}) = 2Q(I + Q)^{-1}$$
$$I + U = (((I + Q) + (I - Q))(I + Q)^{-1}) = 2I(I + Q)^{-1}$$
$$(I - U)(I + U)^{-1} = (2Q(I + Q)^{-1})(2I(I + Q)^{-1})^{-1} = Q$$

Furthermore, it is not difficult to see that Q is a rotation if and only if U is skew-symmetric.

For n = 2, a skew-symmetric matrix can be written as

$$U = \begin{pmatrix} 0 & u \\ -u & 0 \end{pmatrix} ,$$

and no real 2×2 skew-symetric is exceptional, because

$$|I + U| = 1 + u^2 \ge 1$$
.

The rational parameterization of the 2×2 non-exeptional rotation matrices is defined by the map $\mathbb{R}^2 \to O(2)$ given by

$$Q(u) = \begin{pmatrix} 1 & -u \\ u & 1 \end{pmatrix} \begin{pmatrix} 1 & u \\ -u & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1-u^2}{1+u^2} & \frac{2u}{1+u^2} \\ \frac{2u}{1+u^2} & \frac{1-u^2}{1+u^2} \end{pmatrix}.$$
 (2.12)

The only exceptional two-dimensional rotation is the matrix

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

which corresponds to a rotation of π radians.

For n = 3, we can write the general skew-symmetric matrix in the following way

$$U = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} ,$$

because, if $u = (u_1, u_2, u_3)^t$ and v are two three-dimensional vectors, then Uv is equal to the vector product $u \times v$. Again, no skew-symmetric 3×3 matrix is exceptional, because $|I + U| = 1 + u_1^2 + u_2^2 + u_3^2 \ge 1$, and the only exceptional rotation matrices are those corresponding to rotations of π radians. The rational parameterization of the 3×3 rotation matrices is defined by the map $\mathbb{R}^3 \to O(3)$ given by

$$Q(u) = \begin{pmatrix} \frac{1+u_1^2-u_2^2-u_3^2}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_1u_2-u_3)}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_1u_3-u_2)}{1+u_1^2+u_2^2+u_3^2} \\ \frac{2(u_1u_2+u_3)}{1+u_1^2+u_2^2+u_3^2} & \frac{1-u_1^2+u_2^2-u_3^2}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_2u_3-u_1)}{1+u_1^2+u_2^2+u_3^2} \\ \frac{2(u_1u_3-u_2)}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_2u_3+u_1)}{1+u_1^2+u_2^2+u_3^2} & \frac{1-u_1^2-u_2^2+u_3^2}{1+u_1^2+u_2^2+u_3^2} \end{pmatrix} .$$
(2.13)

Note that this function maps a neighborhood of the origin u = 0 continuously onto a neighborhood of the identity matrix Q(0) = I. Clearly, if Q_0 is a rotation, then, the map

 $u \mapsto Q_0 Q(u)$ maps a neighborhood of the origin onto a neighborhood of the matrix Q_0 , and in this way we can deal with the exceptional orthogonal matrices. Also note that the three parameters have explicit geometrical meaning. If $u_2 = u_3 = 0$, the matrix Q(u) becomes

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1-u_1^2}{1+u_1^2} & \frac{-2u_1}{1+u_1^2} \\ 0 & \frac{2u_1}{1+u_1^2} & \frac{1-u_1^2}{1+u_1^2} \end{pmatrix} ,$$

a rotation about the x_1 axis. More generally, if v(t) describes the trajectory of a point which rotates with constant angular velocity, the motion equations can be written as $\dot{v} = u \times v = Uv$, where \dot{v} is the time derivative of v(t). In the language of Lie groups, the skew-symmetric matrices are the infinitesimal generators of the group of rotations.

2.10.4 Cylinders, algebraic curves and surfaces

The implicit equations of a straight line can be written as

$$\begin{cases} Q_1^t x - \alpha_4 &= 0\\ Q_2^t x &= 0 \end{cases}$$

where Q_1, Q_2, Q_3 are the columns of a general rotation matrix $Q(\alpha_1, \alpha_2, \alpha_3)$ parameterized as in (2.13), with $\alpha_1, \alpha_2, \alpha_3$ real parameters, in which case the general implicit representation of a cylinder is given as the set of zeros of

$$(Q_1^t x - \alpha_4)^2 + (Q_2^t x)^2 - \alpha_5^2 = x^t (I - Q_3 Q_3^t) x - 2\alpha_4 Q_1^t x + \alpha_4^2 - \alpha_5^2, \qquad (2.14)$$

the set of points at a distance $|\alpha_5|$ from the straight line. By homogeneity, the set of zeros of (2.14) does not change if we multiply it by the constant $\kappa^2 = (1 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2)^2 \neq 0$, so that, the cylinder can also be represented as the set of zeros of

$$f(x) = x^{t} (\kappa^{2} I - q_{3} q_{3}^{t}) x - 2\alpha_{4} \kappa q_{1}^{t} x + \kappa^{2} (\alpha_{4}^{2} - \alpha_{5}^{2}) , \qquad (2.15)$$

where

$$q_{1} = \kappa Q_{1} = \begin{pmatrix} 1 + \alpha_{1}^{2} - \alpha_{2}^{2} - \alpha_{3}^{2} \\ 2(\alpha_{1}\alpha_{2} + \alpha_{3}) \\ 2(\alpha_{1}\alpha_{3} - \alpha_{2}) \end{pmatrix} \text{ and } q_{3} = \kappa Q_{2} = \begin{pmatrix} 2(\alpha_{1}\alpha_{3} + \alpha_{2}) \\ 2(\alpha_{2}\alpha_{3} - \alpha_{1}) \\ 1 - \alpha_{1}^{2} - \alpha_{2}^{2} + \alpha_{3}^{2} \end{pmatrix}.$$

This representation is particularly attractive because the coefficients of f(x) are *polynomials* in the five unconstrained parameters $\alpha_1, \ldots, \alpha_5$, and so, much less expensive to evaluate than the trigonometric functions. Explicitly,

$$f(x) = \begin{pmatrix} (1+\alpha_1^2+\alpha_2^2+\alpha_3^2)(\alpha_4^2-\alpha_5^2) \\ -2\,\alpha_4((1+\alpha_1^2)^2-(\alpha_2^2+\alpha_3^2)^2) \\ -4\,\alpha_4(1+\alpha_1^2+\alpha_2^2+\alpha_3^2)(\alpha_1\alpha_2+\alpha_3) \\ -4\,\alpha_4(1+\alpha_1^2+\alpha_2^2+\alpha_3^2)(\alpha_1\alpha_3-\alpha_2) \\ (1+\alpha_1^2+\alpha_2^2+\alpha_3^2)^2-4\,(\alpha_1\alpha_3+\alpha_2)^2 \\ -8\,(\alpha_1\alpha_3+\alpha_2)(\alpha_2\alpha_3-\alpha_1) \\ -4\,(\alpha_1\alpha_3+\alpha_2)(1-\alpha_1^2-\alpha_2^2+\alpha_3^2) \\ (1+\alpha_1^2+\alpha_2^2+\alpha_3^2)-4(\alpha_2\alpha_3-\alpha_1) \\ -4\,(\alpha_2\alpha_3-\alpha_2)(1-\alpha_1^2-\alpha_2^2+\alpha_3^2) \\ 4\,(1+\alpha_1^2+\alpha_2^2)(1+\alpha_3^2)-4 \end{pmatrix}^t \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_1^2 \\ x_1x_2 \\ x_1x_3 \\ x_2^2 \\ x_2x^3 \\ x_3^2 \end{pmatrix}$$

The same procedure that we applied above, the multiplication of the implicit equation of the cylinder by a power of κ to convert the rational parameterization of the coefficients into a polynomial one can be applied to the composition of a fixed polynomial with a rigid body transformation to obtain a polynomial parameterization of all the members of the family, even if some shape parameters are present, as in the cylinder.

2.10.5 Superquadrics

According to Solina [124], superquadrics were discovered by the Danish designer Peit Hein [61] as an extension of basic quadric surfaces and solids. They have been used as primitives for shape representation in Computer Graphics by Barr [10], in Computer Aided Design by Elliot [51], and in Computer Vision by Pentland [107], Bajcsy and Solina [8], Solina [124] and Boult and Gross [22, 23, 73].

Barr divides the superquadrics into superellipsoids, superhyperboloids and supertoroids. We will only consider the case of superellipsoids here as an example. The usual representation of a superellipsoid is the set of zeros of f(x) = g(x) - 1, where

$$g(x) = \left(\left| \frac{u^t x}{a_1} - b_1 \right|^{\frac{2}{\epsilon_1}} + \left| \frac{v^t x}{a_2} - b_2 \right|^{\frac{2}{\epsilon_1}} \right)^{\frac{\epsilon_1}{\epsilon_2}} + \left| \frac{w^t x}{a_3} - b_3 \right|^{\frac{2}{\epsilon_2}}$$

and [u v w] is a general rotation matrix parameterized by the three Euler angles $\theta_1, \theta_2, \theta_3$, and $a_1, a_2, a_3, b_1, b_2, b_3, \epsilon_1, \epsilon_2$ are unconstrained parameters. Solina [124] proposes an alternative parameterization with

$$f(x) = \sqrt{a_1 a_2 a_3} (|g(x)|^{\epsilon_2} - 1)$$

to overcome the bias problem associated with the minimization of the mean square error, and Gross and Boult [73] compare the performance of these two representations with other two, one of them proposed by themselves.

We propose the following parameterization, a generalization of the parameterization of cylinders (2.15), which simplifies the computation of the partial derivatives with respect to α , required by the Levenson-Marquardt algorithm

$$\phi(\alpha, x) = \left(\alpha_7^2 (q_1^t x - \alpha_4 \kappa)^{2\alpha_{10}} + \alpha_8^2 (q_2^t x - \alpha_5 \kappa)^{2\alpha_{10}}\right)^{\frac{\alpha_{11}}{\alpha_{10}}} + \alpha_9^2 (q_3^t x - \alpha_6 \kappa)^{2\alpha_{11}} - \kappa^{2\alpha_{11}},$$

where q_1 , q_2 and q_3 are the three columns of the parameterized rotation matrix (2.13) multiplied by κ^2 . Superhyperboloids and supertoroids can be parameterized in a similar way.

2.11 Related work on implicit curve and surface fitting

According to Duda and Hart [48], the earliest work on fitting a line to a set of points was probably motivated by the work of the experimental scientist; the easiest way to explain a set of observations is to relate the dependent and independent variables by means of the equation of a straight line. Minimum-squared-error line fitting [48, section 9.2.1] and eigenvector line fitting [48, section 9.2.2] are the two classical solutions to this classical problem. With our notation, in both cases the mean square error (2.5) is minimized, with $F = (F_1 \ F_2 \ F_3)$ and $X = (1 \ x_1 \ x_2)$. In the first case the linear constraint $F_3 = 1$ is imposed, while the quadratic constraint $F_2^2 + F_3^2 = 1$ is imposed in the second case. Early work on eigenvector line fitting can be traced back to Pearson [106] and Hotelling [82] on principal components analysis. Jollife [85, chapter 1] gives a brief history of principal components analysis.

Pratt [110] affirms that there appears to be relatively little written about fitting planar algebraic curves to data [104, 103, 13, 4, 137, 36, 37, 20, 64, 118, 33], and none whatsoever of least-squares fitting of nonplanar algebraic surfaces. We can add to this statement that we have been unable to locate any previous work on fitting space curves defined by implicit functions, except for Ponce and Kriegman [109] who fit the *projection* of a space curve to two-dimensional data points, and only a few references on fitting quadric surfaces to data [62, 75, 57, 31, 16, 17] However, there is an extensive literature on fitting parametric curves and surfaces to scattered data, for example [41, 59, 63, 96, 105].

The first extensions of the line fitting algorithms concentrated on fitting conics to data minimizing the mean square error. Since the implicit equations are homogeneous, a constraint has to be imposed on the coefficients to obtain a nontrivial solution, and either linear or quadratic constraints were considered. With our notation a conic is the set of zeros of a polynomial f(x) = FX, where $F = (F_1, \ldots, F_6)$, and $X = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_2^2)^t$. Biggerstaff [13], Albano [4] and Cooper and Yalabik [36, 37] impose the constraint $F_1 = 1$. This constraint has the problem that no conic that pass through the origin satisfies it, and so it can not be a solution of the problem. Paton [104, 103] imposes the constraint $F_1^2 + \cdots + F_6^2 = 1$, and Gnanadesikan [64] uses $F_2^2 + \cdots + F_6^2 = 1$. These two constraints are not invariant with respect to translation and rotation of the data, while Bookstein's constraint [20] $F_4^2 + F_5^2/2 + F_6^2$ is. However, the previous two quadratic constraints allow the solution to have $F_4 = F_5 = F_6 = 0$, a straight line, while Bookstein does not. Pratt [110] shows that this is a great disadvantage, and proposes a new quadratic constraint for the case of circles. Cernuschi [31] derives a quadratic constraint which is invariant under the action of the Euclidean group for quadric surfaces, generalizing Bookstein's constraint.

In general, linear constraints turn the minimization of the mean square error into a linear regression problem, while quadratic constraints convert it into an eigenvalue problem. It is important to note that in all the previous cases, the constraints are independent of the data. They have been chosen beforehand. In our generalized eigenvector fit the quadratic

constraint is a function of the data.

More recently Chen [33] fitted general linearly parameterized planar curves to data. With our notation an admissible function is $f(x) = FX = F_1X_1 + \cdots F_hX_h$. He imposes the constraint $F_h = 1$ turning the minimization into a linear regression problem, and he solves the normal equations with the QR algorithm [66, chapter 6]. This constraint looks arbitrary because it rules out all the curves with $F_h = 0$ or close to zero. A different ordering of the basis vector would produce a different solution. In the case of fitting circles to data shown as an application in Chen's paper, where $X = (1 x_1 x_2 x_1^2 + x_2^2)^t$, a straight line can not be a valid solution as in Bookstein's algorithm, and the same kind of behavior has to be expected.

Chen apparently developed his algorithm independently of the earlier work of Pratt [110], whose *simple fit* algorithm is very close to Chen's, except for the fact that he solves the ordering problem in a clever way. Pratt presents his algorithm for algebraic curves and surfaces, but it can be formulated for the general linear case with the same effort. Let $\mathcal{D} = \{p_1, \ldots, p_q\}$ be the set of data points. Let us first assume that q = h - 1, where *h* is the number of elements of *X*, and the $h \times q$ design matrix

$$X_{\mathcal{D}} = [X(p_1) \cdots X(p_q)]$$

has maximal rank h - 1. The function

$$f(x) = \det\left(\left[X_{\mathcal{D}} X\right]\right)$$

is a linear combination of the elements of X, with the coefficients being the cofactors of the elements of the *h*-th column. Note that for every i = 1, ..., h - 1

$$f(p_i) = \det\left(\left[X_{\mathcal{D}} X(p_i)\right]\right) = 0$$

because the matrix has two identical columns. The function f(x) interpolates the data points. Pratt calls this technique, which generalizes the classical Vandermonde determinant [40, chapter II], exact fit. The coefficients can be efficiently computed by first triangularizing X_D via column operations at cost $O(h^3)$, and then computing the cofactors at an additional cost $O(h^2)$. In the general case he applies the Cholesky decomposition to the square matrix $X_D X_D^t = q M_D$ obtaining the unique square lower triangular matrix L with positive diagonal elements such that $X_D X_D^t = LL^t$, then he deletes the last column of L and treats the result as though it were the $h \times (h - 1)$ matrix X_D of the exact fit case. In this procedure the coefficient of X_h is never zero and corresponds to the constraint $F_h = 1$, producing the same solution as Chen's algorithm. The way Pratt overcomes the ordering problem is by performing the Cholesky decomposition with full pivoting, permuting the elements of X during the decomposition. No one coefficient is singled out as having to be nonzero. Pratt calls this procedure simple fit.

With respect to iterative methods, Sampson [118] introduced a reweight procedure to improve Bookstein's algorithm in the case of "very scattered" data. He explains that this reweighting scheme does not necessarily converge, but he does not make use of further optimization techniques.

Solina [124] and Gross and Boult [73] fit superquadrics [10] to data minimizing the mean square error (2.7) using the Levenberg-Marquardt algorithm.

Ponce and Kriegman [109] introduce two iterative methods for estimating the distance from a point to an implicit curve, and then use the Levenberg-Marquardt algorithm to minimize the sum of the squares of these estimates. Both methods are obviously more expensive than minimizing the approximate mean square distance.

2.12 Analysis of the generalized eigenvector fit

Let *M* and *N* be $r \times r$ symmetric matrices, *N* nonnegative definite, and let *F* be a $k \times r$ matrix. Let us consider the problem of minimizing

trace
$$(FMF^t)$$
 (2.16)

constrained by

$$FNF^t = \mathbf{I}_k . (2.17)$$

Let $h = \operatorname{rank}(N)$, then $h \ge k$, otherwise the problem has no solution, because from (2.17)

 $h = \operatorname{rank}(N) \ge \operatorname{rank}(FNF^t) = k$.

We will see that this condition is also sufficient for the existence of solution. Also $k = \operatorname{rank}(F)$, because

$$k = \min\{k, h\} \ge \operatorname{rank}(F) \ge \operatorname{rank}(FNF^t) = k$$

Let F_i denote the *i*-th row of F, and D_{F_i} the Jacobian with respect to the *r* variables of F_i . We can rewrite (2.16) as

$$\sum_{i=1}^{k} F_i M F_i^t$$

and (2.17) as

$$F_i N F_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad i, j = 1, \dots, k .$$

Let us consider the function

$$\varphi(F,\lambda) = \sum_{l=1}^{k} F_l M F_l^t - \sum_{l=1}^{k} \sum_{j=1}^{k} \lambda_{lj} [F_l N F_j^t - \delta_{lj}].$$

where λ is a $k \times k$ matrix. If $\hat{F}_1, \ldots, \hat{F}_k$ are the rows of a minimizer \hat{F} of (2.16)-(2.17), by the Lagrange multipliers theorem [6, Theorem 7-10], there exist a matrix $\hat{\lambda}$, such that

$$0 = \frac{1}{2} D_{F_i}[\varphi](\hat{F}, \hat{\lambda}) = \hat{F}_i M - \sum_{j=1}^k \hat{\lambda}_{ij} \hat{F}_j N \qquad i = 1, \dots, k ,$$

or in matrix form

$$\hat{F}M = \hat{\lambda}\hat{F}N$$

Since the matrix *M* is symmetric and nonnegative definite, so is FMF^t for every matrix *F*, and for certain orthogonal $k \times k$ matrix *Q*

$$Q[FMF^t]Q^t = [QF]M[QF]^t$$

is diagonal. Since $Q^t = Q^{-1}$ and (2.16) and (2.17) are invariant with respect to similarity transformations

trace
$$(FMF^t)$$
 = trace $([QF]M[QF]^t)$

and

$$FNF^t = I_k \quad \Leftrightarrow \quad [QF]N[QF]^t = I_k$$

the matrix F is a solution of the minimization problem if and only if QF is. We may assume without loss of generality, that $\hat{F}M\hat{F}^t$ is diagonal. In this case $\hat{\lambda}$ is nonnegative definite and diagonal because

$$\hat{\lambda} = \hat{\lambda} \operatorname{I}_k = \hat{\lambda} \hat{F} N \hat{F}^t = \hat{F} M \hat{F}^t ,$$

or equivalently

$$\hat{F}_i M = \hat{\lambda}_{ii} \hat{F}_i N$$
 $i = 1, \dots, k$

 F_i is a generalized eigenvector of the symmetric-positive pencil $M - \lambda N$ corresponding to the eigenvalue $\hat{\lambda}_{ii}$. It is clear that if $0 \leq \lambda_1 \leq \cdots \leq \lambda_h$ are the eigenvalues of $M - \lambda N$, the choice $\hat{\lambda}_{11} = \lambda_1, \ldots, \hat{\lambda}_{kk} = \lambda_k$ defines a solution of the minimization problem, and a solution clearly exists. The solution is unique if and only if $\lambda_k < \lambda_{k+1}$, and it is a subspace of dimension $s \leq h - k$ if $\lambda_k = \lambda_{k+1} = \cdots = \lambda_{k+s}$ and, s = h - k or $\lambda_{k+s} < \lambda_{k+s+1}$.

The pencil $M - \lambda N$ is *symmetric-positive* if N is also positive definite. Algorithms to solve a symmetric-positive eigenproblem are well known [66, Sec. 8.6.], and very good subroutine packages are available such as EISPACK [122, 60]. The following is a brief description of such an algorithm.

Since *N* is symmetric and positive definite, it has a Cholesky decomposition $L^{-1}NL^{-t} = I_r$, where *L* is nonsingular and lower triangular. Let $0 \le \lambda_1 \le \cdots \le \lambda_k$ be the least eigenvalues of $H = L^{-1}ML^{-t}$, and let U_1, \ldots, U_k be corresponding orthogonal row eigenvectors. The eigenvalues can be computed with the QR algorithm, and then the eigenvectors can be computed one by one by Inverse Iteration. If *U* is the $k \times r$ matrix with rows U_1, \ldots, U_k , $\lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_k)$, and $F = UL^{-1}$, then

$$FM = UHL^{t} = \lambda UL^{t} = \lambda FN ,$$

$$FNF^{t} = UU^{t} = I_{k} ,$$

and *F* is the solution of the original problem. *F* can be computed by backsubstitution .

If *N* is rank-deficient, as it is for example in the case of polynomials, we can reduce the original problem to a symmetric-positive one, generalizing the previous algorithm [67].

Since *N* is nonnegative definite, using the Cholesky decomposition with full pivoting , we can find an $r \times h$ matrix L_1 and an $r \times (r - h)$ matrix L_2 such that $N = L_1 L_1^t$, $NL_2 = 0$, and the square matrix $L = [L_1 L_2]$ is nonsingular. The matrix *L* is a product of transpositions and elementary triangular transformations. In this case the matrix $L^{-1}NL^{-t}$

is not the identity matrix, but a diagonal matrix with *h* ones followed by r - h zeros on the diagonal. The matrix $H = L^{-1}ML^{-t}$, can be decomposed in the following way

$$H = \begin{pmatrix} H_3 + H_2 H_1 H_2^t & H_2 H_1 \\ H_1 H_2^t & H_1 \end{pmatrix} = \begin{pmatrix} I_h & H_2 \\ 0 & I_{(r-h)} \end{pmatrix} \begin{pmatrix} H_3 & 0 \\ 0 & H_1 \end{pmatrix} \begin{pmatrix} I_h & H_2 \\ 0 & I_{(r-h)} \end{pmatrix}^t$$

The matrices H_1, H_2, H_3 are computed in this order from the corresponding blocks of H. Let U_1 be a $k \times h$ matrix, U_2 a $k \times (r - h)$ matrix, and $U = [U_1 \ U_2]$. Since L is nonsingular, we can write $F = UL^{-1}$ for certain matrix U. Then

trace
$$(FMF^t)$$
 = trace (UHU^t)
= trace $(U_1H_3U_1^t)$ + trace $([U_1H_2 + U_2]H_1[U_1H_2 + U_2]^t)$.

and

$$FNF^t = \mathbf{I}_k \qquad \Leftrightarrow \qquad U_1U_1^t = \mathbf{I}_k \ .$$

Since U_2 is not included in the constraint, and the matrix $[U_1H_2+U_2]H_1[U_1H_2+U_2]^t$ is nonnegative definite, if F is a solution of the minimization problem, then $U_2 = -U_1H_2$. The original problem has been reduced to minimizing trace $(U_1H_3U_1^t)$ constrained by $U_1U_1^t = I_k$, which is a particular case of the symmetric positive problem.

Chapter 3

Segmentation Algorithms

As an application of the ideas introduced in preceding chapters, we have implemented two versions of the same segmentation algorithm. The first one segments planar edge maps into algebraic planar curve patches, and the second one segments range images into algebraic surface patches. In both cases the maximum degree is a parameter that can be chosen by the user, but due to numerical and complexity considerations, it is not appropriate to use patches of degree higher than four or five. There is a straightforward extension of these algorithms to the segmentation of space edge maps, composed of surface discontinuities, surface normal discontinuities, occluding boundaries, or lines of curvature, into algebraic space curve patches, which we are currently implementing.

These algorithms are partially based on Besl and Jain's variable-order surface fitting algorithm [12, 11], Silverman and Cooper's surface estimation clustering algorithm [121], and they are also related to Chen's planar curve reconstruction algorithm [33].

Both versions have the same structure with minor differences at the implementation level. The basic building blocks are: noise variance estimation, region growing, and merging.

Our philosophy is that this segmentation is most useful when the appropriate segmentation is well defined, i.e., when there are range or surface normal discontinuities between regions, each of which is well represented by a single polynomial.

3.1 Noise variance estimation

Since the algorithm follows an *hypothesize and test* approach, tests for accepting or rejecting a hypothesized curve or surface as a good approximation for a given set of data points have to be chosen. Also, good subsets of the data set to start the region growing process, the *seeds*, have to be located. These two subjects are closely related.

Our tests are based on modeling a smooth region of the data set as samples of an implicit curve or surface plus additive perturbations in the orthogonal direction to the curve or surface. These orthogonal perturbations are assumed to be independent Gaussian random variables of zero mean. The square of the noise variance at one data point $\hat{\sigma}(x)$ is estimated by fitting a straight line or plane to the data in a small neighborhood of the point, in our implementation a circle or ball of radius equal to a few pixels, using the eigenvector fit method, and then computing the approximate mean square distance to the fitted line or plane. This estimator can be biased if the curvature of the curve or surface is large at the point.

Although a much better method to estimate the noise variance would be to fit a circle or ellipsoid to the data in a neighborhood of the point and then measure the approximate mean square distance to this curve or surface, we have experienced very good results with the former method.

The first step of the algorithm is to estimate the square of the noise variance at every data point, and to store these values in an array for latter usage. In the case of surfaces, we only compute the estimated noise variances on a subsampled image, to save computational time.

The second step of the algorithm is to build a histogram of the square noise variance. The data points with square noise variance in the top ten percent of the histogram are marked as outliers. This points correspond to mixed regions, corners, or surface normal discontinuities. All the remaining points are left available to start growing regions from the small neighborhoods of them used to estimate the noise variances, because a straight line or plane well approximates the data in that neighborhood.

3.2 Goodness of fit tests

During the region growing process, curve or surfaces are hypothesized as good approximations for a given subset of data. These hypotheses have to be tested because the criterion used for fitting is different from the desired one. We have chosen the thresholds used to test the validity of a hypothesized curve or surface approximation for a set of data points as functions of the noise variance estimates for the individual points of the set, instead of choosing global thresholds for the full data set, as in [12]. The reasons are two. In the first place, we have observed that in the case of surfaces the point noise variance estimated with the method described in the previous section is non-stationary, varying slowly according to the angle of the normal with respect to the vertical, that is, the viewing direction. In the second place, local thresholds allow parallel implementations of the algorithms, because distant regions become independent of each other. The data set can be spatially divided into blocks of approximately the same size, the region growing algorithm can be run in parallel for each block, and then pairs of neighboring blocks can also be merged in parallel. A pyramid architecture can be used to implement such an algorithm. The two implementations that we present in this paper are sequential, and are only intended to show the usefulness of the previous fitting techniques. We will implement the corresponding parallel versions in the near future.

Let $S = \{p_1, \ldots, p_q\}$, be a set of data points, and let α be a parameter vector which defines an implicit curve or surface.

The *mean noise variance estimate* on the set S will be denoted

$$\hat{\sigma}_{\mathcal{S}}^2 = \frac{1}{q} \sum_{i=1}^q \hat{\sigma}(p_i) ,$$

where $\hat{\sigma}(p_i)$ is the noise variance at the point p_i estimated with the method of the previous section. The *maximum approximate square distance* from the set S to the curve or surface defined by α will be denoted

$$\delta_{\mathcal{S}}^2(\alpha) = \max_{1 \le i \le q} \delta(\alpha, p_i)^2.$$

Two tests are performed on the parameter vector α to decide whether to accept the curve or surface define be α as a good approximation of the set S or not. The first test is related to the χ^2 statistic. The parameter vector α passes the first test if the approximate mean square distance satisfies the inequalities

$$\epsilon_1 \, \hat{\sigma}_{\mathcal{S}}^2 < \Delta_{\mathcal{S}}^2(\alpha) < \epsilon_2 \, \hat{\sigma}_{\mathcal{S}}^2 ,$$

where $0 < \epsilon_1 < 1 < \epsilon_2$ are test constants. If α satisfies the first test, then we perform the second test. The parameter vector α passes the second test if

$$\delta_{\mathcal{S}}^2(\alpha) < \epsilon_3 \Delta_{\mathcal{S}}^2(\alpha) ,$$

where $\epsilon_3 > 1$ is another test constant.

If the parameter vector α does not satisfy the first test with $\Delta_{\mathcal{S}}^2(\alpha) \leq \epsilon_1 \hat{\sigma}_{\mathcal{S}}^2$, that is, with the approximate mean square distance to small, the curve or surface defined by α is *overaproximating* the data set, and the hypothesis has to be rejected. For example, if we try to test a pair of very close parallel lines as an approximation of a noisy straight line segment. In a variable-order algorithm, this means that the order has to be *decreased*. If the parameter vector α does not satisfy the first test with $\epsilon_1 \hat{\sigma}_{\mathcal{S}}^2 \leq \Delta_{\mathcal{S}}^2(\alpha)$, that is, with the approximate mean square distance to large, too many points of \mathcal{S} are too far away from the curve or surface defined by α , and it can not possibly pass the second test. The hypothesis has to be rejected. Finally, our goal is to accept the parameter α only if the curve or surface defined by α approximates well every point of \mathcal{S} , that is, the maximum approximate distance is not to large with respect to the variance estimate for the set. The second test takes care of this situation.

3.3 Region growing

The basic structure of the variable-order region growing algorithm can be described as follows. An increasing sequence $\mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_{order_{MAX}}$ of families of functions is given. A *region* is a data structure $\mathcal{R} = (S, f, order)$, where S is a connected subset of data points, and f is an element of \mathcal{F}_{order} which well approximates every point of S. A point x is well approximated by the curve or surface defined by the parameter vector α if the approximate distance satisfies the inequality

$$\delta(\alpha, x)^2 < \epsilon_2 \,\hat{\sigma}_{\mathcal{S}}^2 \,,$$

where ϵ_2 is the same constant of the previous paragraph. In the case of surfaces, we also test surface normal continuity. When the noise variance is estimated at a point, the equation of the fitting line or plane also provides an estimate for the curve or surface normal at the

point. If the angle between the gradient of the surface defined by α at a point under test and the estimate for the surface normal at the same point is larger than a certain value, the point is rejected.

The region growing starts by finding a *seed region* $\mathcal{R} = (S, \mathbf{f}, 1)$, where S is a subset of data points and \mathbf{f} is an element of \mathcal{F}_1 , whose set of zeros $Z(\mathbf{f})$ approximates well every point of S. In our case \mathcal{F}_1 is the family of polynomials of first degree, and a seed region is the subset of data points in the neighborhood of a point not marked as an outlier, which was used to estimate the noise variance at the point, together with the fitted straight line or plane.

Then, given a current region $\mathcal{R} = (S, \mathbf{f}, order)$, the following loop is repeated until no further grow in S is observed. The maximal connected region S' of points well approximated by \mathbf{f} and which intersects the initial seed set is computed. If S' does not have more points than S, then neither S nor \mathbf{f} are changed and the loop is exited. Otherwise a new member \mathbf{f}' of \mathcal{F}_{order} is fitted to S', and if it satisfies the *goodness of fit* test, the region \mathcal{R} is replaced by $\mathcal{R} = (S', \mathbf{f}', order)$, and the loop repeated. If \mathbf{f}' does not satisfy the test, the loop is exited. When the loop is exited, if *order* is equal to the maximum order $order_{MAX}$ the region growing is finished returning the current region $\mathcal{R} = (S, \mathbf{f}, order)$. Otherwise a member \mathbf{f}' of $\mathcal{F}_{order+1}$ is increased to order + 1, that is, \mathcal{R} is replaced by $\mathcal{R} = (S, \mathbf{f}', order + 1)$, and the loop is traversed once more. If \mathbf{f}' does not satisfy the test, the region growing the test, the region $\mathcal{R} = (S, \mathbf{f}, order + 1)$.

In our implementation of the algorithm for planar curve segmentation, the second order corresponds to circles, the third to general second degree polynomials, the fourth to third degree polynomials, and so on. In the case of surfaces the second order corresponds to spheres and cylinders, the third order to second degree polynomials, the fourth to third degree polynomials, and so on. The reasons for introducing an extra family between first and second degree polynomials are two. Firstly, we want to locate large and smooth curve or surface patches without curve or surface normal discontinuities, singularities. Since any pair of straight line or planar patches can be well approximated by the set of zeros of a valid second degree algebraic curve or surface, the set of zeros of the product of the two linear equations, without the new family, it is equally likely that a region grow across a curve or surface normal discontinuity than along a smooth curved area. The second reason is that slightly curved areas of the data set which are almost well approximated by a straight line are usually best approximated by second degree curves or surfaces with many branches, such as pairs of lines or planes, hyperbolas or hyperboloids, and ellipses or ellipsoids with very unequal mean axes.

The reason for introducing the merging process is that once a region has grown large enough, very few new points compatible with the current hypothesized curve or surface can be found, and the fit corresponding to the next order is generally rejected due to the same problem that we just explained for the case of almost flat data sets approximated by second degree curves or surfaces. The approximate mean square distance is too small. With the merging algorithm a larger proportion of points is included in the current region, and there exist a possibility of a successful fit of higher order. We have restricted the curve or surface types used in the region growing process to those that can be computed only with the generalized eigenvector fit algorithm, without the need of improving them with the reweight heuristic and the iterative Levenson-Marquardt algorithm, saving time in the computation. For planar curves we use straight lines and circles. For surfaces, the parallel of circles are spheres and cylinders, but since fitting cylinders require the Levenberg-Marquardt algorithm and a strategy to compute an initial estimate, we also moved the spheres and cylinders to the merging phase, leaving a single order planar region growing.

Figures 3.1 and 3.2 provide a description of our implementation of the region growing algorithm for planar curves. The corresponding surface algorithms are simplified

```
procedure GrowRegionFromPoint (p)
   S := SeedSet (p)
   order := Line
   f := FitImplicit (order, S)
   \mathcal{R} := (\mathcal{S}, \mathbf{f}, order)
   CopyRegion (\mathcal{R}, \mathcal{R}')
    order_{MAX} := Circle
   LOOP:
   GrowRegion (\mathcal{R}, \mathcal{R}')
   TEST:
   if order' = Circle or order = Line
       CopyRegion (\mathcal{R}', \mathcal{R})
       if order = Circle then
           order := Line
          f := FitImplicit (order, S)
          if TestOrder (\mathcal{R}) \neq Increase then
              CopyRegion (\mathcal{R}, \mathcal{R}')
               order_{MAX} := Line
              goto LOOP
       else if order = Line
          if order_{MAX} = Circle
               order := Circle
              f := FitImplicit (order, S)
              goto LOOP
   return (\mathcal{R}')
```

Figure 3.1: Region growing algorithm for planar curves.

versions of the planar curve algorithms described above. **TestOrder** (\mathcal{R}) is a procedure which implements the goodness of fit test described in the previous section for the region $\mathcal{R} = (\mathcal{S}, \mathbf{f}, order)$, and returns one of three values : *Decrease, Accept,* or *Increase*. It is clear

procedure GrowRegion $(\mathcal{R}, \mathcal{R}')$ do $\Delta_1 :=$ **CompatiblePoints** (\mathcal{R}) $\delta_1 := |\Delta_1|$ $\mathcal{S} := \mathcal{S} \cup \Delta_1$ f := FitImplicit (order, S) $\Delta_2 :=$ **CompatiblePoints** (\mathcal{R}) $\delta_2 := |\Delta_2|$ $\mathcal{S}'' := \mathcal{S} \cup \Delta_2$ S := LargestComponent (p, S'')if $|\mathcal{S}| < |\mathcal{S}''|/2$ then if order = Circle then order := Line return $\delta_3 := |\mathcal{S}''| - |\mathcal{S}|$ if $|\mathcal{S}| > |\mathcal{S}'|$ then **CopyRegion** $(\mathcal{R}, \mathcal{R}')$ $\delta_4 := \delta_1 + \delta_2 + \delta_3$ if TooFewPoints (order, S) then if order = Circle then order := Line return while $\max\{|\delta_1|, |\delta_2|, |\delta_3|, |\delta_4|\} > \delta_{MIN}$ return

procedure CompatiblePoints (\mathcal{R}) $B := \{x : ||x|| \le radius\}$ S' := Sdo S := S' $S' := \{x \in S \oplus B : IsCompatible (x, f, order)\}$ while $\mathcal{R}' \ne \mathcal{R}$ return (\mathcal{R})

Figure 3.2: Core of region growing algorithm for planar curves.

what the procedure **FitImplicit** (*order*, S) does. With respect to **GrowRegion** (\mathcal{R} , \mathcal{R}'), the procedure **CompatiblePoints** (\mathcal{R}) computes a not necessarily connected region intersecting the current region by sequentially dilating the current region and checking for compatible points. **IsCompatible** (x, \mathbf{f} , *order*) just checks that the approximate distance from x to the set $Z(\mathbf{f})$ be within the limits imposed by the second part of the goodness of fit test. In the case of surfaces it also checks the angular deviation between the estimated normal and the gradient of f at the point. Without this check some surface patches grow along very thin strips of neighboring transversal patches. **LargestComponent** (p, S) computes the largest connected component of the set S which is adjacent or contains the point p. Finally, **TooFewPoints** (*order*, S) rejects a region if it has shrinked to much with respect to the given order.

3.4 Merging

The variable-order merging tries to find *maximal* subsets of the smooth regions or, depending on the application, groups of smooth regions, each of them represented as a subset of the set of zeros of a single implicit curve or surface. The merging process can be seen as a generalization of region growing, where regions grow not by single points, but by groups of points.

At every step the variable order merging produces the best possible merge of two neighboring regions. At the beginning all the neighboring pairs which are well fitted by a surface of the current order are computed and inserted into a priority queue according to the value of the maximum relative approximate square distance, the number

$$\frac{\delta_{\mathcal{S}}^2(\alpha)}{\Delta_{\mathcal{S}}^2(\alpha)} ,$$

where S is the union of the two sets being considered for merging. The priority queue is implemented with a binary heap. Every node of this heap consists of a tern of regions $(\mathcal{R}, \mathcal{R}', \mathcal{R}'')$, where \mathcal{R}' and \mathcal{R}'' are two neighboring regions, the set S of the region \mathcal{R} is the union of the sets S' and S'' corresponding to the region \mathcal{R}' and \mathcal{R}' , and the element f of \mathcal{R} , is an acceptable fit of the current *order* for S.

Then, and while the queue is not empty, the pair corresponding to the minimum value is deleted from the queue and merged, creating a new region and deleting two, all the pairs which are still in the queue and involve either one of the two merged regions are deleted, and all the pairs which involve the new region are recomputed and reinserted in the queue. The procedure **Merging** (*order*, *list*), described in figure 3.3, is called sequentially for increasing values of *order*, until the maximum one, with *list* equal to the current list of regions. The procedure **MergeAndFit** (*order*, \mathcal{R} , \mathcal{R}'), fits a surface of the requested order to the union of the two regions, and returns a nonempty data structure only if a fit of the given order successfully passes the goodness of fit test. Finally, the procedures **MinOrder** and **MaxOrder** impose preliminary limits before the fitting precess. For example, if two regions are currently approximated by quadrics, it does not make sense to fit a plane to the union, nor an

```
procedure Merging (order, list)
     heap := \emptyset
     for \mathcal{R}' \in list
          for \mathcal{R}'' \in list \setminus \{\mathcal{R}'\}
               node := MergeAndFit (order, \mathcal{R}', \mathcal{R}'')
               if node \neq \emptyset then
                    Insert (node, heap)
     while heap \neq \emptyset
          (\mathcal{R}, \mathcal{R}', \mathcal{R}'') := DeleteMin (heap)
          \mathcal{R}' := \mathcal{R}
          list := list\{\mathcal{R}''}
          for \mathcal{R} \in list \setminus \{\mathcal{R}'\}
               if S \cap S' \neq \emptyset then
                    Delete (\mathcal{R}, heap)
          for \mathcal{R}'' \in list \setminus \{\mathcal{R}'\}
               if \mathcal{S}'' \cap \mathcal{S}' \neq \emptyset then
                    node := MergeAndFit (order, (\mathcal{R}', \mathcal{R}''))
                    if \mathcal{R} \neq \emptyset then
                          Insert (node, heap)
     return (list)
```

procedure MergeAndFit (order, $\mathcal{R}', \mathcal{R}''$) if order < MinOrder (order', order'') then return (\emptyset) if order > MaxOrder (order', order'') then return (\emptyset) $\mathcal{S} := \mathcal{S}' \cup \mathcal{S}''$ f := FitImplicit (order, \mathcal{S}) if TestOrder (\mathcal{R}) \neq Accept then $\mathcal{R} := \emptyset$ return (($\mathcal{R}, \mathcal{R}', \mathcal{R}''$))

Figure 3.3: Merging procedure.

algebraic surface of degree higher than four, because the product of the two quadrics already approximates the union well.

3.5 Experimental results

Figures 3.4, 3.5 and 3.6 show the results of our segmentation algorithm applied to contours obtained by thresholding gray level images taken by a standard TV quality CCD camera. Figures 3.8 and 3.7 show the corresponding results for the segmentation of range images taken with a White Scanner model 100 laser range finding system. And finally, figure 3.9 shows the segmentation of one range image from the NRCC data base [113], the file "jet4".



Figure 3.4: Segmentation of planar curve. (a): Data Set. (b): Segmentation after region growing with straight line and circle primitives. (c): After merging (b) using general second degree algebraic curve primitives (conics). (d): After merging (c) using general third degree algebraic curve primitives.



Figure 3.5: Segmentation of planar curve. (a): Data Set. (b): Segmentation after region growing with straight line and circle primitives. (c): After merging (b) using general second degree algebraic curve primitives (conics). (d): After merging (c) using general fourth degree algebraic curve primitives.



Figure 3.6: Segmentation of planar curve. (a): Data Set. (b): Segmentation after region growing with straight line and circle primitives. (c): After merging (b) using general second degree algebraic curve primitives (conics). (d): After merging (c) using general third degree algebraic curve primitives.



Figure 3.7: Segmentation of 3D scene : sphere, cylinder and box. (a): Original range image represented as a gray level image. (b): The same range image from a perspective view. (c): White areas represent regions occluded by shadows. (d): Result of region growing with planar patches reconstructed as a gray level image. (e): The same result from a perspective view. (f): The segmentation after the region growing. (g): Result of merging (d)-(e)-(f) with general second degree algebraic surface patches (quadrics), reconstructed as a gray level image. (h): The same result from a perspective view. (f): The segmentation after the merging.



Figure 3.8: Segmentation of 3D scene : spray bottle and cylinders. (a): Original range image represented as a gray level image. (b): The same range image from a perspective view. (c): White areas represent regions occluded by shadows. (d): Result of region growing with planar patches reconstructed as a gray level image. (e): The same result from a perspective view. (f): The segmentation after the region growing. (g): Result of merging (d)-(e)-(f) with general second degree algebraic surface patches (quadrics), reconstructed as a gray level image. (h): The same result from a perspective view. (f): The segmentation after the merging.



Figure 3.9: Segmentation of 3D scene : NRCC "jet4" file. (a): Original range image represented as a gray level image. (b): The same range image from a perspective view. (c): White areas represent regions occluded by shadows and the background. (d): Result of region growing with planar patches reconstructed as a gray level image. (e): The same result from a perspective view. (f): The segmentation after the region growing. (g): Result of merging (d)-(e)-(f) with general second degree algebraic surface patches (quadrics), reconstructed as a gray level image. (h): The same result from a perspective view. (f): The segmentation after the merging.

Chapter 4 Invariant theory of algebraic forms

In previous chapters we have described methods for fitting algebraic curves and surfaces to data points. In this chapter we develop the tools for efficiently solving the algebraic curve and surface matching problem. This problem can be described as follows. We start with a *database* of curves or surfaces, a finite set of algebraic curves or surfaces, the sets of zeros of the polynomials f_1, \ldots, f_r , These curves or surfaces describe complex regions of object boundaries, regions which uniquely determine the position of the corrersponding object. Then, for every other curve or surface extracted from the data set, the zeros of a polynomial **f**, we have to decide whether or not there exist a coordinate transformation x' = T(x), and an element f_i of the database, such that the transformed polynomial $f'(x') = f(T^{-1}(x'))$ defines almost the same curve or surface as the one defined by f_i . The transformation will be Euclidean for the recognition and positioning of objects from range data, but we will also consider similarity, affine, and projective transformations for other applications. If we obtain an affirmative answer, then we want to recover the unknown transformation from the coefficients of the matching pair of polynomials. This transformation would correspond to the hypothetical presence of the object, that the curve or surface defined by f_i represents a region of, in the data set.

Our approach will be based on computing and comparing invariants. An *invariant* of an algebraic curve or surface is a function of the coefficients of the defining polynomials which does not change after a change of coordinates. Every curve or surface will be represented as a point in a multidimensional *invariant space*, and the classification will be carried out using an appropriate distance measure, or a statistical classifier, in invariant space,

By introducing homogeneous coordinates, every curve or surface described in Euclidean space by a polynomial in n indeterminates, can be described in projective space by its associated *homogeneous* polynomial in n + 1 indeterminates. A polynomial is homogeneous, or a *form*, if every one of its terms is of the same degree. If $\phi(x_0, \ldots, x_n)$ is a form of degree d in n + 1 variables, and $\psi(u_1, \ldots, u_n)$ is a regular polynomial of degree $\leq d$ in n variables, the following equations define the processes of *homogeneization* and *dehomogeneization*.

$$\psi(u_1, \dots, u_n) \quad \mapsto \quad \phi(x_0, \dots, x_n) = x_0^d \psi\left(\frac{x_1}{x_0}, \dots, \frac{x_n}{x_0}\right)$$

$$\phi(x_0, \dots, x_n) \quad \mapsto \quad \psi(u_1, \dots, u_n) = \phi(1, u_1, \dots, u_n)$$

For example, the quadratic form

$$\phi(x_0, x_1) = \frac{1}{2} \Phi_{(2,0)} x_0^2 + \Phi_{(1,1)} x_0 x_1 + \frac{1}{2} \Phi_{(0,2)} x_1^2$$
(4.1)

is the homogeneous version of the one dimensional quadratic polynomial

$$\psi(u_1) = \frac{1}{2} \Phi_{(2,0)} u_1^2 + \Phi_{(1,1)} u_1 + \frac{1}{2} \Phi_{(0,2)}$$

Algebraic invariants of polynomials have to be defined with respect to a linear group of transformations. The groups of Euclidean, similarity, and affine transformations, are subgroups of the projective group, and every invariant with respect to one of these group, is also invariant with respect to all the previous groups, because every one of these four groups contain all the previous groups as subgroups. It is natural to start studying invariants of forms with respect to the projective group, i.e., homogeneous linear transformations of the homogeneous coordinates x' = Ax, where A is a nonsingular $n \times n$ matrix, an element of the general linear group GL(n). Projective invariants of algebraic forms also have other applications in Computer Vision, in the recognition and classification of three-dimensional shapes from the two-dimensional projections of their occluding contours [142, 58]. This applications make a systematic treatment of the subject interesting by itself, independently of our intended application.

When invariants with respect to a subgroup of affine transformations are considered, such as Euclidean transformations, it is better to work with regular polynomials. A general affine transformation x'' = Ax + b can be decomposed as a homogeneous transformation x' = Ax, followed by a translation x'' = x' + b. Every regular polynomial can be written in a unique way as a sum of homogeneous polynomials of different degrees, and these homogeneous parts transform independently of each other under homogeneous coordinate transformations, i.e., coordinate transformations without translation part. In this chapter we will consider Euclidean transformations without translation part. The general case will be solved by defining the *center* of an algebraic curve or surface in Chapter 5. A general Euclidean transformation will be homogeneous when written with respect to coordinates systems with the origins in the centers of both curves or surfaces. Euclidean invariants are in one-to-one correspondence with joint Cartesian invariants of the homogeneous terms with respect to the curve or surface center. A Cartesian invariant is an invariant with respect to the orthogonal group. Since we are primarily interested in Euclidean matching of algebraic curves and surfaces, it is extremely important to compute Cartesian invariants of forms. Every invariant of a form with respect to the general linear group, is also invariant with respect to the orthogonal group, but we will be able to exploit the structure of the orthogonal transformations and construct other Cartesian invariants.

According to Dieudonné [45, 44], by the middle of the nineteenth century it was known that, if in the quadratic form (4.1) we make a linear change of variables x' = Ax, where A is a nonsingular 2×2 matrix, we obtain a new quadratic form

$$\phi'(x'_1, x'_2) = \frac{1}{2} \Phi'_{(2,0)} {x'_1}^2 + \Phi'_{(1,1)} x'_1 x'_2 + \frac{1}{2} \Phi'_{(0,2)} {x'_2}^2 ,$$

and the function

$$\mathcal{I}(\phi) = \Phi_{(2,0)} \Phi_{(0,2)} - \Phi_{(1,1)}^2 \tag{4.2}$$
satisfies the following identity

$$\mathcal{I}(\phi') = |A|^{-2} \, \mathcal{I}(\phi) \,, \tag{4.3}$$

where |A| is the determinant of A. A function $\mathcal{I}(\phi)$ of the coefficients of a form ϕ , is a *relative invariant of weight two*, if the identity (4.3) is satisfied. More generally, a relative invariant of weight w of a form, $\phi(x)$ of degree d in n variables is a homogeneous function $\mathcal{I}(\phi)$ of the coefficients, such that, if x' = Ax is a nonsingular homogeneous coordinate transformation, and $\phi'(x') = \phi(A^{-1}x')$ is the polynomial which describes the set zeros of ϕ in the new coordinate system, then $\mathcal{I}(\phi') = |A|^{-w} \mathcal{I}(\phi)$. An absolute invariant is a relative invariant of weight zero.

The classical invariant theory of algebraic forms was developed in the nineteenth century by Boole [21], Cayley [30], Clebsh [34], Elliot [50], Gordan [68], Grace and Young [69], Hilbert [77, 78], Sylvester [126], and others [43, 117], to solve the problem of classification of projective algebraic varieties, i.e., sets of common zeros of several homogeneous polynomials. In this century, the main contributions have been by Weyl [143], Mumford [101] and others [74, 125]. The projective coordinate transformations define a relation of equivalence in the family of algebraic varieties, with two varieties being equivalent if one of them can be transformed into the other by a projective transformation. The projective classification of algebraic varieties is the description of the geometric properties which characterize the members of the classes of equivalence, and for a given curve or surface, the determination of its class of equivalence. For example, the algebraic varieties defined by binary quadratic forms like (4.1) are sets of one or two points, so that, there are two classes of equivalence, and the value of the invariant (4.2) determines the class. The form corresponds to the one point class if $\mathcal{I}(\phi) = 0$, and it corresponds to the two points class if $\mathcal{I}(\phi) \neq 0$. If we consider polynomials with real coefficients and real projective coordinate transformations, instead of complex polynomials and transformations, there are three classes corresponding to sets with zero, one and two points, respectively. In the real case, the form corresponds to the zero points class if $\mathcal{I}(\phi) < 0$, it corresponds to the two points class if $\mathcal{I}(\phi) > 0$, and it corresponds to the one point class if $\mathcal{I}(\phi) = 0$. The invariant (4.2) is called the *discriminant* of (4.1), because it discriminates among the different classes.

The classical approach to the classification problem, as for example the classification of planar algebraic curves defined by a single form $\phi(x)$ of degree d in three variables, is to find a set of relative or absolute invariants, $\{\mathcal{I}_1(\phi), \mathcal{I}_2(\phi), \ldots\}$ whose values determine the class that the form belongs to. One naturally tries to find a minimal family, and Hilbert [77, 78, 3] proved that there exist a finite number of *polynomial* invariants, a *fundamental system* of invariants, such that every other polynomial invariant is equal to an algebraic combination of the members of the fundamental system. But Hilbert's proof is not constructive, and the problem is then, how to compute a fundamental system of polynomial invariants. Algorithms exist, such as the Straightening Algorithm [114], but they are computationally expensive [144].

Due to the finiteness of the database, and the numerical and measurement errors involved, the classification problem that we have to solve is slightly different. We would like to use a fundamental system of polynomial invariants for this purpose, but to achieve a low computational cost is more important. We only need a sufficiently large family { $\mathcal{I}_1(\phi), \ldots, \mathcal{I}_s(\phi)$ } of invariants with the *separation* property, i.e., for every pair of members \mathbf{f}_i and \mathbf{f}_j of the database, there must exist an invariant \mathcal{I}_k such that $\mathcal{I}_k(\mathbf{f}_i) \neq \mathcal{I}_k(\mathbf{f}_j)$, such that members of different calsses are mapped to different points in invariant space. These invariants do not have to be functionally independent. Some of the techniques that we will describe for the computation of invariants will produce vectors of invariants which will not necessarily be functionally independent. However, finding an independent subset will be computationally more expensive than using all the invariants as a feature vector.

In this chapter we describe several techniques for efficiently *computing* invariants and covariants of forms, both with respect to general linear transformations, and with respect to orthogonal transformations. We want to emphasize the computational aspect. For example, the complexity of numerically computing the determinant of a square $n \times n$ matrix A is in the order n^4 arithmetic operations, because, in the order of n^3 operations are needed for computing the QR decomposition of A, and exactly n - 1 multiplications to compute the determinant of the triangular matrix of the decomposition [66]. However, the analytic expression of the determinant |A| as a polynomial of degree n in the n^2 elements of the matrix, has n! terms. Some of these techniques have been well known for a century, but our emphasis on structuring the algorithms for the efficient numerical computation of invariants based on matrix computations is new.

In section 4.1 we introduce the notation and the basic tools to analyze how the coefficients of forms and the partial derivatives of forms transform when homogeneous linear transformations are applied to the space variables. This analysis corresponds to both the projective transformation of forms, and the affine or Euclidean transformations of the homogeneous terms of regular polynomials. In section 4.2 we formally define the concepts of invariant and covariant, and give several methods for constructing and computing projective invariants of forms. In section 4.3 we analyze, and give new methods for the construction and computation of invariants with respect to the orthogonal group. Finally, in section 4.4 we prove all the lemmas and corollaries.

The problem of recovering the unknown transformation which transforms a curve or surface into an equivalent one will be solved in the next chapter, and the relation of algebraic invariants to moment invariants will be covered in the following chapter.

4.1 Polynomials, forms, and linear transformations

Every quadratic form can be written as a product of three terms, a row vector of indeterminates, a matrix of coefficients, and column vector of indeterminates. For example, the form (4.1) can be written as

$$2\phi(x_1, x_2) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^t \begin{pmatrix} \Phi_{(2,0)} & \Phi_{(1,1)} \\ \Phi_{(2,0)} & \Phi_{(1,1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} .$$
(4.4)

If a nonsingular linear transformation is applied to the space variables

$$x' = Ax \qquad : \qquad \begin{cases} x'_1 = a_{11}x_1 + a_{12}x_2 \\ x'_2 = a_{21}x_1 + a_{22}x_2 \end{cases}$$

the transformed polynomial $\phi'(x') = \phi(A^{-1}x')$ can be written in the same way

$$2\phi'(x_1', x_2') = \begin{pmatrix} x_1' \\ x_2' \end{pmatrix}^t \begin{pmatrix} \Phi'_{(2,0)} & \Phi'_{(1,1)} \\ \Phi'_{(2,0)} & \Phi'_{(1,1)} \end{pmatrix} \begin{pmatrix} x_1' \\ x_2' \end{pmatrix} , \qquad (4.5)$$

and the matrices are related by the following formula

$$\Phi'_{[1,1]} = A^{-t} \Phi_{[1,1]} A^{-1} , \qquad (4.6)$$

where $A^{-t} = (A^{-1})^t = (A^t)^{-1}$, $\Phi_{[1,1]}$ is the square matrix of coefficients in (4.4), and $\Phi'_{[1,1]}$ is the corresponding matrix in (4.5). Note that these matrices can be obtain from the original form by differentiation,

$$D_{[1,1]}\phi = \begin{pmatrix} \frac{\partial^2 \phi}{\partial x_1^2} & \frac{\partial^2 \phi}{\partial x_1 x_2} \\ \frac{\partial^2 \phi}{\partial x_1 x_2} & \frac{\partial^2 \phi}{\partial x_2^2} \end{pmatrix} \,.$$

This matrix, the *Hessian matrix* of ϕ , is well defined not only for a quadratic form, but a form ϕ of arbitrary degree. If we denote $D'_{[1,1]}\phi'$ the Hessian matrix of $\phi'(x')$ with respect to the variables $(x'_1, x'_2)^t$, then, we also have the relation

$$D'_{[1,1]}\phi' = A^{-t} D_{[1,1]}\phi A^{-1} ,$$

with (4.6) as a particular case. When ϕ is a quadratic form, the determinant of $D_{[1,1]}\phi$ is an invariant, but in general it is a new form which satisfies the relation

$$|D'_{[1,1]}\phi'(x')| = |A|^{-2} |D_{[1,1]}\phi(x)|.$$
(4.7)

This is a *covariant*. New invariants of a form can be computed from its covariants. As we pointed out in the introduction of this chapter, the computation of $|\Phi_{[1,1]}|$ can be performed inexpensively using numerical techniques, but if the degree of ϕ is higher than two, the computation of the covariant form $|D_{[1,1]}\phi(x)|$ has to be performed *symbolically*, at a much higher computational cost.

The operator defined by the previous construction, $\phi \mapsto |D_{[1,1]}\phi(x)|$, from the set of all binary forms into the same set, is a *polynomial differential operator*. It maps a form ϕ into another form which is computed by evaluating a polynomial of many variables in some partial derivatives of ϕ . For example, for the Hessian matrix of the binary quadratic form (4.1), we can write

$$\begin{cases} p(\theta_1, \theta_2, \theta_3) &= \theta_1 \theta_3 - \theta_2^2 , \\ D_{[1,1]} \phi &= p\left(\frac{\partial^2 \phi}{\partial x_1^2}, \frac{\partial^2 \phi}{\partial x_1 x_2}, \frac{\partial^2 \phi}{\partial x_2^2}\right) . \end{cases}$$

And it is invariant, in the sense that for *every* binary form ϕ , the relation (4.7) is satisfied. In fact, all the invariants and covariants of algebraic forms can be constructed as invariant polynomial differential operators. In the case presented above, the operator is *homogeneous*, in the sense that it is a polynomial evaluated in partial derivatives of ϕ , but all of the same order. In such a case, we obtain an invariant by restricting the operator to forms of degree equal to the order of the derivatives.

In this section we will generalize this construction to forms of higher degree, and will present several methods for the construction of covariants and invariants. Certain matrices of partial derivatives of higher order will be our basic building blocks for the construction of covariants and invariants. Our first task is to define these matrices and to analyze their transformation rules under linear coordinate transformations.

4.1.1 Polynomials

From now on, polynomials will be written expanded in Taylor series at the origin

$$f(x) = \sum_{\alpha} \frac{1}{\alpha!} F_{\alpha} x^{\alpha} , \qquad (4.8)$$

where the vector of nonnegative integers $\alpha = (\alpha_1, \ldots, \alpha_n)^t$ is a *multiindex* of *size* $|\alpha| = \alpha_1 + \cdots + \alpha_n$, $\alpha! = \alpha_1! \cdots \alpha_n!$ is a multiindex factorial, $\{F_\alpha : \alpha\}$ is the set of coefficients of f, F_α is a coefficient of degree $|\alpha|$, and $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ is the *monomial* of degree $|\alpha|$ associated with the multiindex α . There are exactly

$$h_d = \binom{n+d-1}{n-1} = \binom{n+d-1}{d}$$

different multiindices of size d, and so, that many monomials of degree d. A polynomial of degree d has

$$h_d + h_{d-1} + \dots + h_0 = \binom{n+d}{n}$$

coefficients. The coefficients of f are equal to the partial derivatives of order d evaluated at the origin

$$F_{\alpha} = \frac{\partial^{\alpha_1 + \dots + \alpha_n}}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}} f(0)$$

and only finitely many coefficients are different from zero.

4.1.2 Forms, homogeneous coordinates, and projective transformations

A homogeneous polynomial, or *form*, is a polynomial with all the coefficients of the same degree d

$$\phi(x) = \sum_{|\alpha|=d} \frac{1}{\alpha!} \Phi_{\alpha} x^{\alpha} .$$
(4.9)

For example, a fourth degree form in three variables is

$$\phi(x_1, x_2, x_3) = \frac{1}{24} \Phi_{(4,0,0)} x_1^4 + \frac{1}{6} \Phi_{(3,1,0)} x_1^3 x_2 + \frac{1}{6} \Phi_{(3,0,1)} x_1^3 x_3 + \frac{1}{4} \Phi_{(2,2,0)} x_1^2 x_2^2 + \frac{1}{2} \Phi_{(2,1,1)} x_1^2 x_2 x_3 + \frac{1}{4} \Phi_{(2,0,2)} x_1^2 x_3^2 + \frac{1}{6} \Phi_{(1,3,0)} x_1 x_2^3 + \frac{1}{2} \Phi_{(1,2,1)} x_1 x_2^2 x_3 + \frac{1}{2} \Phi_{(1,1,2)} x_1 x_2 x_3^2 + \frac{1}{6} \Phi_{(1,0,3)} x_1 x_3^3 + \frac{1}{24} \Phi_{(0,4,0)} x_2^4 + \frac{1}{6} \Phi_{(0,3,1)} x_2^3 x_3 + \frac{1}{4} \Phi_{(0,2,2)} x_2^2 x_3^2 + \frac{1}{6} \Phi_{(0,1,3)} x_2 x_3^3 + \frac{1}{24} \Phi_{(0,0,4)} x_3^4 .$$
(4.10)

The vector of indeterminates $x = (x_1, ..., x_n)^t$ can be seen as the coordinates of a point in Euclidean *n*-dimensional space, or as the homogeneous coordinates of a point $u = (u_1, ..., u_{n-1})^t$ in Euclidean space of dimension n - 1, if $x_n \neq 0$. The correspondence given by

$$u_1 = \frac{x_1}{x_n}$$
, ..., $u_{n-1} = \frac{x_{n-1}}{x_n}$,

which is clearly not one to one, because for every nonzero constant $\lambda \neq 0$, $\lambda x = (\lambda x_1, \dots, \lambda x_n)^t$ also represents the same point $u = (u_1, \dots, u_{n-1})$. Homogeneous polynomials in n homogeneous coordinates correspond to regular polynomials in n-1 Euclidean coordinates.

A projective transformation is an homogeneous linear transformation x' = Ax on the homogeneous coordinates of a point. Every nonsingular matrix A defines a projective transformation, but the correspondence is not one-to-one. Two nonsingular matrices A and B define the same projective transformation if $A = \lambda B$ for certain constant $\lambda \neq 0$. Affine and Euclidean transformations can be seen as particular cases of projective transformations. An affine transformation u' = Au + b in *n*-1-dimensional space, corresponds to the projective transformation

$$x' = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} x \, .$$

There is a one-to-one correspondence between regular polynomials in $u = (u_1, \ldots, u_{n-1})^t$ of degree $\leq d$, and homogeneous polynomials in $x = (x_1, \ldots, x_n)^t$ of degree d. A planar curve is represented by a polynomial in two variables, or by a homogeneous polynomial in three variables. A surface is represented by a polynomial in three variables or a homogeneous polynomial in four variables. For example, the polynomial (4.10), is associated with the regular polynomial

$$\begin{array}{l} \frac{1}{24} \, \Phi_{(4,0,0)} \, \, u_1^4 \, + \, \frac{1}{6} \, \Phi_{(3,1,0)} \, \, u_1^3 u_2 \, + \, \frac{1}{4} \, \Phi_{(2,2,0)} \, \, u_1^2 u_2^2 \, + \, \frac{1}{6} \, \Phi_{(1,3,0)} \, \, u_1 u_2^3 \, + \, \frac{1}{24} \, \Phi_{(0,4,0)} \, \, u_2^4 \, + \\ \frac{1}{6} \, \Phi_{(3,0,1)} \, \, u_1^3 \, + \, \frac{1}{2} \, \Phi_{(2,1,1)} \, \, u_1^2 u_2 \, + \, \frac{1}{2} \, \Phi_{(1,2,1)} \, \, u_1 u_2^2 \, + \, \frac{1}{6} \, \Phi_{(0,3,1)} \, \, u_3^3 \, + \\ \frac{1}{4} \, \Phi_{(2,0,2)} \, \, u_1^2 \, + \, \frac{1}{2} \, \Phi_{(1,1,2)} \, \, u_1 u_2 \, + \, \frac{1}{4} \, \Phi_{(0,2,2)} \, \, u_2^2 \, + \\ \frac{1}{6} \, \Phi_{(1,0,3)} \, \, u_1 \, + \, \frac{1}{6} \, \Phi_{(0,1,3)} \, \, u_2 \, + \\ \frac{1}{24} \, \Phi_{(0,0,4)} \, \end{array}$$

4.1.3 Vectors and matrices

Now we will introduce vectors and matrices of monomials, and their duals, the corresponding vectors and matrices of partial derivative operators, and will analyze their transformation rules under linear coordinate transformations. In subsequent sections, these vectors and matrices will constitute the basic building blocks for the construction of invariants and covariants.

The lexicographical order for the multiindices is defined as usual,

$$\alpha < \beta \quad \Leftrightarrow \quad \begin{cases} \text{either} & |\alpha| < |\beta| \\ \text{or} & |\alpha| = |\beta| \\ & \text{and for certain } 1 < k \le n : \\ & \alpha_1 = \beta_1, \dots, \alpha_{k-1} = \beta_{k-1}, \alpha_k > \beta_k \end{cases}$$

For example, for multiindices of size 2 in three variables, the lexicographical order is

$$(2,0,0) < (1,1,0) < (1,0,1) < (0,2,0) < (0,1,1) < (0,0,2)$$
.

From now on, all the sets of objects subindicated with multiindices will be implicitly ordered according to the lexicographical order, and they will be considered vectors or matrices according to whether one or two subindices are used.

Monomials

The set of monomials of degree d lexicographically ordered

$$\left\{\sqrt{\frac{1}{\alpha!}} x^{\alpha} : |\alpha| = d\right\} , \qquad (4.11)$$

is an h_d -dimensional vector, which we will denote $X_{[d]}(x)$. For example,

$$X_{[3]}(x_1, x_2, x_3) = \begin{pmatrix} \frac{1}{\sqrt{6}} x_1^3 \\ \frac{1}{\sqrt{2}} x_1^2 x_2 \\ \frac{1}{\sqrt{2}} x_1^2 x_3 \\ \frac{1}{\sqrt{2}} x_1 x_2^2 \\ x_1 x_2 x_3 \\ \frac{1}{\sqrt{2}} x_1 x_3^2 \\ \frac{1}{\sqrt{2}} x_2^2 x_3 \\ \frac{1}{\sqrt{2}} x_2^2 x_3 \\ \frac{1}{\sqrt{2}} x_2 x_3^2 \\ \frac{1}{\sqrt{6}} x_3^3 \end{pmatrix}.$$

.

For every pair of nonnegative integers, k and j, the set of monomials

$$\left\{\sqrt{\frac{1}{\alpha!\,\beta!}}\,x^{\alpha+\beta}\,:\,|\alpha|=k,|\beta|=j\right\}$$

lexicographically ordered, defines an $h_k \times h_j$ matrix which we will denote $X_{[k,j]}(x)$. For example,

$$X_{[2,1]}(x_1, x_2, x_3) = \begin{pmatrix} \frac{1}{\sqrt{2}} x_1^3 & \frac{1}{\sqrt{2}} x_1^2 x_2 & \frac{1}{\sqrt{2}} x_1^2 x_3 \\ x_1^2 x_2 & x_1 x_2^2 & x_1 x_2 x_3 \\ x_1^2 x_3 & x_1 x_2 x_3 & x_1 x_3^2 \\ \frac{1}{\sqrt{2}} x_1 x_2^2 & \frac{1}{\sqrt{2}} x_2^3 & \frac{1}{\sqrt{2}} x_2^2 x_3 \\ x_1 x_2 x_3 & x_2^2 x_3 & x_2 x_3^2 \\ \frac{1}{\sqrt{2}} x_1 x_3^2 & \frac{1}{\sqrt{2}} x_2 x_3^2 & \frac{1}{\sqrt{2}} x_3^3 \end{pmatrix}$$

$$(x) = X_{[d]}(x), \text{ and } X_{[i,k]}(x) = X_{[k]}(x) X_{[i]}^t(x) = X_{[k,i]}^t(x).$$

Note that $X_{[d,0]}(x) = X_{[d]}(x)$, and $X_{[j,k]}(x) = X_{[k]}(x)X_{[j]}^t(x) = X_{[k,j]}^t(x)$.

Partial differential operators

The space of forms of degree d in n variables is an h_d -dimensional vector space, and the set of monomials (4.11), i.e., the elements of $X_{[d]}(x)$, form a basis of it. Now, we will show that its dual space is the space of homogeneous *linear* differential operators of the same degree, and will show that the dual basis of (4.11) is the corresponding set of partial derivatives of the same order.

Let $D = (\partial/\partial x_1, \dots, \partial/\partial x_n)^t$ be the vector of first order partial derivatives, and for every multiindex α , let D^{α} be the linear partial differential operator

$$D^{\alpha} = \left(\frac{\partial}{\partial x_1}\right)^{\alpha_1} \cdots \left(\frac{\partial}{\partial x_n}\right)^{\alpha_n} = \frac{\partial^{\alpha_1 + \dots + \alpha_n}}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}$$

For every form of degree d

$$\psi(x) = \sum_{|\eta|=d} \frac{1}{\eta!} \Psi_{\eta} x^{\eta}$$

there is a corresponding homogeneous linear differential operator

$$\psi(D) = \sum_{|\eta|=d} \frac{1}{\eta!} \Psi_{\eta} D^{\eta} ,$$

and every homogeneous linear differential operator of degree d can be written in this form in a unique way, i.e., the vector space of linear differential operators of order d is a vector space of the same dimension h_d , and the map $\psi(x) \mapsto \psi(D)$ defines an isomorphism of vector spaces.

Note that for every pair of multiindices, η of size k, and α of size d, we have

$$D^{\eta}(x^{\alpha}) = \begin{cases} \frac{\alpha!}{\beta!} x^{\beta} & \text{if } \alpha_1 \ge \eta_1, \dots, \alpha_n \ge \eta_n \text{ and } \alpha = \beta + \eta \\ 0 & \text{otherwise }, \end{cases}$$
(4.12)

so that, if $|\eta| = |\alpha| = d$

$$\sqrt{\frac{1}{\eta!}} D^{\eta} \left(\sqrt{\frac{1}{\alpha!}} x^{\alpha} \right) = \begin{cases} 1 & \text{if } \eta = \alpha \\ 0 & \text{otherwise} \end{cases}$$

and the set of homogeneous partial differential operators

$$\left\{\sqrt{\frac{1}{\alpha!}} D^{\alpha} : |\alpha| = d\right\} ,$$

i.e., the elements of the vector $X_{[d]}(D)$, is a dual basis of (4.11). Furthermore, the map $f(x) \mapsto f(D)$, which transforms an arbitrary polynomial into the corresponding linear differential operator, is an isomorphism of rings, and every polynomial identity has a corresponding identity for linear differential operators [3, chapter B]. That is, in order to prove an identity for linear differential operators, we only have to prove the associated polynomial identity.

Coefficients

Consistently with the notation introduced for vectors and matrices of monomials, and in order to simplify what follows, we will write $D_{[d]}$ for the vector of operators $X_{[d]}(D)$, and $D_{[k,j]}$ for the matrix of operators $X_{[k,j]}(D)$. If ϕ is a form of degree d and $0 \le k \le d$, then $D_{[k]}\phi(x)$ is a vector of forms of degree d - k. In particular, $D_{[d]}\phi$ is a vector of constants, the coefficients of ϕ

$$\left\{\sqrt{\frac{1}{\alpha!}}\,\Phi_{\alpha}\,:\,|\alpha|=d\right\}$$

which we will denote $\Phi_{[d]}$. In this way, a form ϕ of degree *d* can be written in vector form

$$\phi(x) = \Phi_{[d]}^t X_{[d]}(x) . \tag{4.13}$$

The matrix $D_{[k,d-k]}\phi$, a matrix of constants, will be denoted $\Phi_{[k,d-k]}$. For example, for the fourth degree form in three variables (4.10) we have

$$\Phi_{[2,2]} = \begin{pmatrix} \frac{1}{2}\Phi_{(4,0,0)} & \frac{1}{\sqrt{2}}\Phi_{(3,1,0)} & \frac{1}{\sqrt{2}}\Phi_{(3,0,1)} & \frac{1}{2}\Phi_{(2,2,0)} & \frac{1}{\sqrt{2}}\Phi_{(2,1,1)} & \frac{1}{2}\Phi_{(2,0,2)} \\ \frac{1}{\sqrt{2}}\Phi_{(3,1,0)} & \Phi_{(2,2,0)} & \Phi_{(2,1,1)} & \frac{1}{\sqrt{2}}\Phi_{(1,3,0)} & \Phi_{(1,2,1)} & \frac{1}{\sqrt{2}}\Phi_{(1,1,2)} \\ \frac{1}{\sqrt{2}}\Phi_{(3,0,1)} & \Phi_{(2,1,1)} & \Phi_{(2,0,2)} & \frac{1}{\sqrt{2}}\Phi_{(1,2,1)} & \Phi_{(1,1,2)} & \frac{1}{\sqrt{2}}\Phi_{(1,0,3)} \\ \frac{1}{2}\Phi_{(2,2,0)} & \frac{1}{\sqrt{2}}\Phi_{(1,3,0)} & \frac{1}{\sqrt{2}}\Phi_{(1,2,1)} & \frac{1}{2}\Phi_{(0,4,0)} & \frac{1}{\sqrt{2}}\Phi_{(0,3,1)} & \frac{1}{2}\Phi_{(0,2,2)} \\ \frac{1}{\sqrt{2}}\Phi_{(2,1,1)} & \Phi_{(1,2,1)} & \Phi_{(1,1,2)} & \frac{1}{\sqrt{2}}\Phi_{(0,3,1)} & \Phi_{(0,2,2)} & \frac{1}{\sqrt{2}}\Phi_{(0,1,3)} \\ \frac{1}{2}\Phi_{(2,0,2)} & \frac{1}{\sqrt{2}}\Phi_{(1,1,2)} & \frac{1}{\sqrt{2}}\Phi_{(1,0,3)} & \frac{1}{2}\Phi_{(0,2,2)} & \frac{1}{\sqrt{2}}\Phi_{(0,1,3)} & \frac{1}{2}\Phi_{(0,0,4)} \end{pmatrix}$$

Finally, we can generalize (4.13), and write the partial derivatives with this matrix notation. The following result shows that writing a polynomial expanded in Taylor series, the computation of partial derivatives of a form becomes a shift operation.

Lemma 1 Let $\phi(x) = \Phi_{[d]}^t X_{[d]}(x)$ be a form of degree d, and let k be a nonnegative integer such that $k \leq d$. Then, for every multiindex η of size k, we have

$$D^{\eta}\phi(x) = \sum_{|\beta|=d-k} \frac{1}{\beta!} \Phi_{\beta+\eta} x^{\beta},$$

or equivalently, in matrix form

$$D_{[k]}\phi(x) = \Phi_{[k,d-k]}X_{[d-k]}(x)$$
.

4.1.4 Euler's theorem

As we mentioned earlier, it is well known how to write a quadratic form as a product of three terms, a row vector of variables, a matrix of coefficients, and a column vector of variables. In this section we will generalize this construction to higher degree forms.

Euler's theorem is a classical result which says how to reconstruct a form of degree d from its partial derivatives

$$d \phi(x) = \sum_{i=1}^{n} \frac{\partial \phi(x)}{\partial x_i} x_i = X_{[1]}^t(x) D_{[1]} \phi(x)$$

According to Lemma 1, for d = 2 this last expression can be rewritten as

$$2\phi(x) = X_{[1]}(x)^t \Phi_{[1,1]} X_{[1]}(x) .$$

More generally, as Walker [141, theorem 10.3] points out, Euler's theorem can be enunciated as follows

Lemma 2 (Euler's theorem) Let $\phi(x) = \Phi_{[d]}^t X_{[d]}(x)$ be a form of degree d, and let k be a non-negative integer such that $k \leq d$. Then,

$$\binom{d}{k}\phi(x) = \sum_{|\eta|=k} \frac{1}{\eta!} D^{\eta}\phi(x) x^{\eta}$$

or equivalently, in matrix form

$$\begin{pmatrix} d \\ k \end{pmatrix} \phi(x) = X_{[k]}^t(x) D_{[k]} \phi(x) = X_{[k]}^t(x) \Phi_{[k,d-k]} X_{[d-k]}(x) .$$

4.1.5 Transformation rules

In this section we study how the vectors and matrices

$$X_{[d]}(x) \quad X_{[k,j]}(x) \quad D_{[d]}\phi(x) \quad D_{[k,j]}\phi(x) \quad \Phi_{[d]} \quad \Phi_{[k,d-k]}$$

transform when the space variables undergo either a nonsingular linear coordinate transformation, or an orthogonal transformation.

If x' = Ax is a nonsingular linear transformation, for every form $\psi(x)$, the polynomial $\psi(Ax)$ is a form of the same degree. In particular, every component of $X_{[d]}(Ax)$ can be written in a *unique way* as a linear combination of the elements of $X_{[d]}(x)$, or in matrix form

$$X_{[d]}(Ax) = A_{[d]}X_{[d]}(x) ,$$

where $A_{[d]}$ is a nonsingular $h_d \times h_d$ matrix. We will call the map $A \mapsto A_{[d]}$ the *k*-th. degree representation map, and the matrix $A_{[d]}$ the *k*-th. degree representation matrix of A. Furthermore,

- **Lemma 3** 1. The map $A \mapsto A_{[d]}$ defines a faithful linear representation, $a \ 1 1$ homomorphism of groups, of the group of nonsingular $n \times n$ matrices GL(n) into the group of nonsingular $h_d \times h_d$ matrices $GL(h_d)$, i.e., for every pair of nonsingular matrices A, and B, we have
 - (a) The map preserves products : $(AB)_{[d]} = A_{[d]}B_{[d]}$.
 - (b) The map is one to one : if $A_{[d]} = B_{[d]}$, then A = B.
 - (c) The matrix $A_{[d]}$ is nonsingular, and $(A_{[d]})^{-1} = (A^{-1})_{[d]}$.
 - 2. The map $A \mapsto A_{[d]}$ preserves transposition, i.e., for every nonsingular matrix A, we have $(A^t)_{[d]} = (A_{[d]})^t$. In particular,

- (a) If A is symmetric, so is $A_{[d]}$.
- (b) If A is symmetric positive definite, so is $A_{[d]}$.
- (c) If A is orthogonal, so is $A_{[d]}$.

We can say more about this family of representations, and we will need the following three results in the sequel.

Lemma 4 Let d be a nonnegative integer, $m = \binom{n+d-1}{n}$, $A = (a_{ij})$ a lower (upper) triangular $n \times n$ matrix, and $a = (a_{11}, \ldots, a_{nn})^t$ the diagonal of A. Then,

- 1. The matrix $A_{[d]}$ is lower (upper) triangular.
- 2. For every multiindex α of size d, the α -th element of the diagonal of $A_{[d]}$ is a^{α} .
- 3. The determinant of $A_{[d]}$ is equal to $|A|^m$.

Corollary 1 Let d be a nonnegative integer, and let $m = \binom{n+d-1}{n}$. Then, for every $n \times n$ matrix A we have

$$|A_{[d]}| = |A|^m$$

Corollary 2 Let d be a nonnegative integer, let A be a diagonal $n \times n$ matrix, and let $a = (a_{11}, \ldots, a_{nn})^t$ be its diagonal. Then,

- 1. The matrix $A_{[d]}$ is diagonal.
- 2. For every multiindex α of size d, the α -th element of the diagonal of $A_{[d]}$ is a^{α} .

The transformation rules for the matrices of monomials clearly follow from the corresponding rules for the vectors of monomials. If x' = Ax is a nonsingular linear transformation, then

$$X_{[k,j]}(Ax) = X_{[k]}(Ax)X_{[j]}(Ax)^{t} = A_{[k]}X_{[k]}(x)X_{[j]}(x)^{t}A_{[j]}^{t} = A_{[k]}X_{[k,j]}(x)A_{[j]}^{t}$$

Now we can determine the transformation rules for the vectors and matrices of partial derivatives.

Lemma 5 Let x' = Ax be a nonsingular linear transformation, D' the vector of partial derivatives with respect to the new coordinate system, $\phi(x)$ a form of degree d, $\phi'(x') = \phi(A^{-1}x')$ the unique form of degree d such that $\phi'(x') = \phi(x)$, and k and j two nonnegative integers. Then,

1. $D'_{[k]} = A^{-t}_{[k]} D_{[k]}$.

2.
$$D'_{[k,j]} = A^{-t}_{[k]} D_{[k,j]} A^{-1}_{[j]}$$
.

Or equivalently,

1.
$$D'_{[k]}\phi'(x') = A^{-t}_{[k]}D_{[k]}\phi(x)$$
.

2. $D'_{[k,j]}\phi'(x') = A^{-t}_{[k]} \left(D_{[k,j]}\phi(x) \right) A^{-1}_{[j]}$.

The transformation rules for the vectors and matrices of coefficients of a form constitute particular cases of this result.

Corollary 3 Let $\phi = \Phi_{[d]}^t X_{[d]}$ be a form of degree d, x' = Ax a nonsingular linear transformation, $\phi'(x') = \phi(A^{-1}x) = \Phi_{[d]}'^t X_{[d]}(x')$ the unique form of degree d such that $\phi'(x') = \phi(x)$, and k a nonnegative integer such that $k \leq d$. Then,

- 1. $\Phi'_{[d]} = A^{-t}_{[d]} \Phi_{[d]}$.
- 2. $\Phi'_{[k,d-k]} = A^{-t}_{[k]} \Phi_{[k,d-k]} A^{-1}_{[d-k]}$.

Finally, since for an orthogonal matrix $A^{-t} = A$, we obtain

Corollary 4 Let x' = Ax be an orthogonal transformation, D' the vector of partial derivatives with respect to the new coordinate system, $\phi(x)$ a form of degree d, $\phi'(x') = \phi(A^t x') = \Phi_{[d]}'^t X_{[d]}(x')$ the unique form of degree d such that $\phi'(x') = \phi(x)$, and k a nonnegaive integer such that $k \leq d$. Then,

- 1. $D'_{[k]} = A_{[k]} D_{[k]}$.
- 2. $D'_{[k,j]} = A_{[k]} D_{[k,j]} A^t_{[j]} = A_{[k]} D_{[k,j]} A^{-1}_{[j]}$.

3.
$$\Phi'_{[d]} = A_{[d]} \Phi_{[d]}$$

4. $\Phi'_{[k,d-k]} = A_{[k]} \Phi_{[k,d-k]} A^t_{[d-k]} = A_{[k]} \Phi_{[k,d-k]} A^{-1}_{[d-k]}$

4.2 Algebraic invariants and covariants

A relative *covariant* of weight w of a form ϕ of degree d, is classically defined as a function

$$\mathcal{C}(\phi, x)$$

of the coefficients of the form and a vector of indeterminates, homogeneous in each variable

$$\mathcal{C}(\theta_1 \phi, \theta_2 x) = \theta_1^{k_1} \theta_2^{k_2} \mathcal{C}(\phi, x) ,$$

for certain rational constants k_1 and k_2 , and for every value of θ_1 and θ_2 , and such that, if x' = Ax is a nonsingular linear transformation, then

$$\mathcal{C}(\phi', x') = |A|^{-w} \mathcal{C}(\phi, x)$$

where $\phi'(x') = \phi(A^{-1}x')$. If w = 0 the covariant is *absolute*, and if the covariant is independent of the indeterminates, it is an *invariant*.

In other words, if k_1 and k_2 are nonnegative integers, a covariant of the form $\phi(x)$ is another form $\psi(x) = C(\phi, x)$ of degree k_2 , with coefficients forms of degree k_1 in the coefficients of ϕ , and such that, if x' = Ax is a coordinate transformation, $\phi'(x') = \phi(A^{-1}x')$, and $\psi'(x') = \psi(A^{-1}x')$, then $\psi'(x') = |A|^w C(\phi', x')$, i.e., the coefficients of ψ' can be computed, except for a multiplicative constant, either by transforming its coefficients according to the coordinate transformation, as for any form, or by evaluating the same homogeneous function of the coefficients of ϕ in the coefficients of the transformed form ϕ' . The most general covariants can be written as quotients of these.

One immediately proves that covariants of covariants are covariants, invariants of covariants are invariants, and homogeneous functions of invariants are invariants. If the set of forms has two relative invariants, \mathcal{I}_1 of weight $w_1 \neq 0$, and \mathcal{I}_2 of weight $w_2 \neq 0$, one can easily construct an absolute invariant

$${\cal I} \;=\; rac{{\cal I}_1^{w_2}}{{\cal I}_2^{w_1}} \;.$$

The same construction applies to a covariant and an invariant. If C_1 is a relative covariant of weight $w_1 \neq 0$, and \mathcal{I}_2 is an invariant of weight $w_2 \neq 0$, then

$$\mathcal{C} = rac{\mathcal{C}_1^{w_2}}{\mathcal{I}_2^{w_1}}$$

is an absolute invariant.

The trivial example of a covariant of a form ϕ is the identity, the form itself $C(\phi, x) \equiv \phi(x)$. Less trivial covariants are given by polynomials of one variable evaluated in the form ϕ

$$\mathcal{C}(\phi, x) = p(\phi(x))$$
 where $p(\theta) = \sum_{i=0}^{a} p_i \theta^i$.

For example, $C(\phi, x) \equiv \phi(x)^2$. And the most general covariant of a form ϕ , which is a homogeneous *polynomial* of the coefficients, can be constructed by evaluating a polynomial of many variables in the form ϕ and some of its partial derivatives

$$\mathcal{C}(\phi, x) = p\left((D^{\alpha}\phi(x))_{\alpha}\right),$$

where $p((\theta_{\alpha})_{\alpha})$ is a polynomial in an infinite number of variables, one for each multiindex α [3]. Ratios of two of these yield rational covariants. However, not every polynomial of this kind defines a covariant, and here is where the vectors and matrices of partial derivatives, and their transformation rules, come into play. We will construct covariants and invariants using certain matrix operations which are known to be covariant, or even invariant, with respect to the matrix transformation rules, such as the determinant and the characteristic polynomial of a square matrix.

Joint covariants

This definition of covariant of a form given at the beginning of this section, can be generalized to joint covariants of several forms, each of them function not only of the vector of indeterminates x, but of several vectors of indeterminates, in a natural fashion. This is another instance of the more general notion of covariant described above. A relative *covariant* of weight w of r forms ϕ_1, \ldots, ϕ_r and s vectors of indeterminates x^1, \ldots, x^s is a function

$$\mathcal{C}(\phi_1, \dots, \phi_r, x^1, \dots, x^s) \tag{4.14}$$

of the coefficients of the forms and the vectors, homogeneous in each variable, both in the forms and the indeterminates, and such that, if x' = Ax is a nonsingular linear transformation, then

$$\mathcal{C}(\phi'_1, \dots, \phi'_r, x^{1\prime}, \dots, x^{s\prime}) = |A|^{-w} \mathcal{C}(\phi_1, \dots, \phi_r, x^1, \dots, x^s) .$$
(4.15)

The covariant is *absolute* if w = 0, and it is an *invariant* if s = 0. In general, we will just write C for (4.14), and C' for the left side of (4.15).

A general definition of covariants

We can give a more general definition of covariant, which applies not only in the context of forms [45]. An *action* of a group \mathbb{G} on a set *E* is a map

$$\begin{cases} \mathbb{G} \times E & \to & E \\ (\sigma, e) & \mapsto & \sigma \cdot e \end{cases}$$

which satisfies the following two properties

1.
$$1_{\mathbb{G}} \cdot e = e$$

2. $\sigma \cdot (\tau \cdot e) = (\sigma \tau) \cdot e$,

where $1_{\mathbb{G}}$ is the identity of the group \mathbb{G} , and the concatenation $\sigma\tau$ denotes the operation of the group. The action is called *trivial* if $\sigma \cdot e = e$ for every element $\sigma \in \mathbb{G}$, and $e \in E$. If E and F are two sets with corresponding actions of the same group \mathbb{G} , a *covariant* of E in F with respect to \mathbb{G} , is a function $\xi : E \to F$ which satisfies the following equation

$$\xi(\sigma \cdot e) = \sigma \cdot \xi(e) ,$$

for every element $\sigma \in \mathbb{G}$, and $e \in E$. The covariant is called *invariant* if the action of \mathbb{G} in F is trivial. For example, in the case of invariants of a form ϕ of degree d, the group \mathbb{G} is GL(n), the group of nonsingular $n \times n$ matrices, the set E is the set of forms of degree d in n variables, and the set F is the set of homogeneous functions in the same n variables, modulo the relation of equivalence which identifies two functions which differ by a multiplicative constant. In both sets, the action is defined by $(A, f(x)) \mapsto f'(x') = f(A^{-1}x')$. The fitting algorithms described in previous sections are also covariants, acording to this definition. In this case the group \mathbb{G} is the group of similarity transformations, the set E is the family of finite subsets of n-dimensional data points, and the set F is the set of algebraic curves or surfaces of a given maximum degree.

4.2.1 Covariant and contravariant matrices

If an $h_k \times h_j$ matrix $C_{[k,j]}$ of functions of r forms and s vectors transforms as the matrix of monomials $X_{[k,j]}$, we will say that it transforms *covariantly* if

$$C'_{[k,j]} = A_{[k]} C_{[k,j]} A^t_{[j]}.$$

and *contravariantly* if it follows the transformation rules of the matrix of partial derivatives $D_{[k,j]}$

 $C'_{[k,j]} = A^{-t}_{[k]} C_{[k,j]} A^{-1}_{[j]}.$

We can clearly extend these definitions to matrices which transform covariantly on the right side and contravariantly on the left side, or viceversa. Vectors which transform covariantly or contravariantly are just special cases $C_{[k]} = C_{[k,0]}$.

For example, if ϕ is a form of even degree d = 2k, the square matrix $\Phi_{[k,k]}$ is a contravariant matrix. If this matrix is nonsingular, then $\Phi_{[k,k]}^{-1}$ is a covariant matrix. If ϕ is a form of odd degree d, we cannot construct a square matrix in the same way, but if $\psi(x) = \phi(x)^2$, then the square matrix $\Psi_{[d,d]}$ is contravariant, and $\Psi_{[d,d]}^{-1}$ is covariant.

If we restrict ourselves to orthogonal transformations, all these concepts coincide, and we only talk about matrices which transform covariantly.

The representation matrices of a covariant matrix

Let $C_{[1,1]}$ be a $n \times n$ covariant matrix, and let us consider the coordinate transformation $x' = C_{[1,1]} x$ associated with it. For every positive integer k, the k-th. degree representation map $A \mapsto A_{[k]}$ defines a new $h_k \times h_k$ matrix $C_{[1,1][k]}$. According to Lemma 3, if $C_{[1,1]}$ is nonsingular, symmetric, positive definite, triangular, or diagonal, so is $C_{[1,1][k]}$. But this new matrix is also covariant.

Lemma 6 If $C_{[1,1]}$ is a $n \times n$ covariant, contravariant, covariant on the left side and contravariant on the right side, or viceversa, so is the $h_k \times h_k$ matrix $C_{[1,1][k]}$, for every positive integer k.

For example, if ϕ is a form of degree $d \ge 2$, we can apply this construction to the Hessian matrix of ϕ , obtaining the contravariant matrices

$$D_{[1,1][k]}\phi = (D_{[1,1]}\phi)_{[k]}$$
 $2 \le k \le d/2$

If the Hessian of ϕ is not identically zero, then, the inverses of these matrices are covariant matrices. If any case, the coefficients of the homogeneous characteristic polynomials

$$\theta_1 D_{[k,k]} \phi + \theta_2 D_{[1,1][k]} \phi$$
 $2 \le k \le d/2$

define new covariants.

4.2.2 Construction of algebraic invariants and covariants

Now we will describe several methods for the construction of covariants and invariants, based on matrix operations on covariant and contravariant matrices. The formulation is particularly attractive because we are interested in *computing* invariants, and not necessarily in giving explicit analytic expressions as functions of the form or forms. For example, computing the value of the Hessian of a quadratic form using numerical techniques is much less expensive than symbolically expanding the determinant of a square matrix.

The k-Jacobian

The first method for the construction of covariants is based on the fact that the Jacobian of n functions of n variables is a covariant of weight one. If $\phi = (\phi_1, \dots, \phi_n)$ is a row vector of forms, The Jacobian of the n forms is the determinant of the Jacobian matrix

$$D_{[1]}\phi = (D_{[1]}\phi_1 \mid \ldots \mid D_{[1]}\phi_n)$$

If x' = Ax is a change of coordinates, and we denote $\phi' = (\phi'_1, \dots, \phi'_n)$, where $\phi'_i(x') = \phi(A^{-1}x')$, we have

$$D'_{[1]}\phi' = (D'_{[1]}\phi'_1 | \dots | D'_{[1]}\phi'_n) = (A^{-t}D_{[1]}\phi_1 | \dots | A^{-t}D_{[1]}\phi_n) = A^{-t}D_{[1]}(\phi_1,\dots,\phi_n),$$

and so, the Jacobian satisfies the following relation

$$|D_{[1]}\phi| = |A|^{-1} |D'_{[1]}\phi'|.$$

This procedure can be generalized in the following way. Let us define the *k*-Jacobian matrix of a row vector $\phi = (\phi_{\alpha})_{|\alpha|=k}$ of h_k forms, not all of them necessarily of the same degree, as the $h_k \times h_k$ matrix

$$D_{[k]}\phi = (D_{[k]}\phi_{\alpha})_{|\alpha|=k} ,$$

and the *k-Jacobian* of ϕ as the determinant of its *k*-Jacobian matrix. The same argument used for the 1-Jacobian shows that

$$|D_{[k]}\phi| = |A_{[k]}|^{-1} |D'_{[1]}\phi'|,$$

and so, based on Corollary 1 we obtain

Lemma 7 The k-Jacobian of a row vector $\phi = (\phi_{\alpha})_{|\alpha|=k}$ of h_k forms is a covariant of weight $m = \binom{n+k-1}{n}$.

This results clearly extends to the most general case

Lemma 8 Let k be a nonnegative integer, let $C = (C_{\alpha})_{|\alpha|=k}$ be a row vector of h_k covariants, each of them function of the vector of indeterminates x. Then, the k-Jacobian $|D_{[k]}C|$ is a covariant.

In particular, when the covariants C_{α} of the last lemma are forms of degree k in x, the matrix $D_{[k]}C$ has as columns the vectors of coefficients of the forms, and the k-Jacobian $|D_{[k]}C|$ becomes an invariant which can be computed inexpensively by numerical techniques.

The determinant of a square covariant matrix

The second method to construct covariants and invariants in based on computing determinant of a square covariant or contravariant matrix.

Lemma 9 Let k be a nonnegative integer, and let $C_{[k,k]}$ be a square covariant matrix. Then, the determinant $|C_{[k,k]}|$, is a covariant.

For example, for every form ϕ of degree d, and every nonnegative integer k such that $k \leq d/2$ we have the covariant

$$|D_{[k,k]}\phi(x)| \tag{4.16}$$

of weight $2\binom{n+k-1}{n}$, a form of degree $(d-2k)h_k = (d-2k)\binom{n+k-1}{n-1}$ in x. For k = 1 it is the well known *Hessian* of the form ϕ . In general, we will call (4.16) the *k*-*Hessian* of ϕ . If d = 2k, the *k*-Hessian of ϕ is an invariant $|\Phi_{[k,k]}|$, which can be computed numerically. The discriminant (4.2) of a binary quadratic form is a particular case of this invariants. The next step is to apply this construction to linear combinations of square covariant matrices of the same size.

The characteristic polynomial

If ϕ and ψ are forms of degree $d \ge 2k$, and the determinant $|D_{[k,k]}\phi|$ is not identically zero, the matrix

$$\mathcal{C}_{[k,k]} = \left(D_{[k,k]} \phi \right)^{-1} \left(D_{[k,k]} \psi \right)$$

is covariant on the left side, and contravariant on the right side. New invariants can be computed from square matrices which are left covariant and right contravariant, or viceversa, because the transformation rules of these matrices correspond to conjugation

$$C'_{[k,k]} = A_{[k]} C_{[k,k]} A^{-1}_{[k]} ,$$

and the characteristic polynomial of a square matrix is invariant under conjugation

$$\left| \theta I - \mathcal{C}'_{[k,k]} \right| = \left| A_{[k]} \left(\theta I - \mathcal{C}_{[k,k]} \right) A_{[k]}^{-1} \right| = \left| \theta I - \mathcal{C}_{[k,k]} \right|.$$

The h_k nonconstant coefficients of the characteristic polynomial

$$\left| \theta I - \mathcal{C}'_{[k,k]} \right| = \theta^{h_k} + \sum_{i=1}^{h_k} (-1)^i \, \mathcal{C}_i \, \theta^{h_k - i}$$

are new covariants. In particular, the trace C_1 , and the determinant C_{h_k} of $C_{[k,k]}$ are absolute covariants. Equivalently, if the coefficients of the square matrix are independent of the indeterminates, the h_k principal values of C_{h_k} are absolute invariants. If C_{h_k} is also symmetric, its h_k eigenvalues are absolute invariants.

For example, if ϕ and ψ are forms of even degree d = 2k and the square matrix $\Phi_{[k,k]}$ is nonsingular, then, the coefficients of the characteristic polynomial

$$\left| \theta I - \Phi_{[k,k]}^{-1} \Psi_{[k,k]} \right| ,$$

are joint absolute invariants of the pair. In particular, the trace and the determinant

trace
$$(\Phi_{[k,k]}^{-1}\Psi_{[k,k]})$$
 $|\Phi_{[k,k]}^{-1}\Psi_{[k,k]}|$

are absolute invariants of the pair. For k = 1 this construction gives us the well known n absolute invariants of a pair of quadratic forms [58, 142].

The homogeneous characteristic polynomial

The method for computing covariants using the characteristic polynomial does not apply to matrices which are covariant or contravariant on both sides. Also, the previous example of the two forms of degree d = 2k is asymmetric, in the sense that it requires $|\Psi_{[k,k]}| \neq 0$, while no constraint is imposed on $\Phi_{[k,k]}$. However, if we multiply the characteristic polynomial of the matrix $\Phi_{[k,k]}\Psi_{[k,k]}^{-1}$ by the determinant $|\Psi_{[k,k]}|$, we obtain

$$\left| \theta I - \Phi_{[k,k]} \Psi_{[k,k]}^{-1} \right| \left| \Psi_{[k,k]} \right| = \left| \theta \Psi_{[k,k]} - \Phi_{[k,k]} \right|,$$

The roots of this polynomial are in one to one correspondence with the roots of the *homogeneous characteristic polynomial*

$$\left| \theta_1 \Psi_{[k,k]} + \theta_2 \Phi_{[k,k]} \right| , \qquad (4.17)$$

and this expression is totally symmetric in $\Psi_{[k,k]}$ and $\Phi_{[k,k]}$. Furthermore, no constraint is imposed on the matrices, and the coefficients of (4.17), as polynomials in (θ_1, θ_2) , are homogeneous in the elements of the two matrices. The characteristic polynomial (4.17) is identically zero if and only if the null spaces of the two square matrices have nonzero intersection, otherwise it has h_k roots in the projective line, which are absolute invariants of the pair of forms. These are *generalized eigenvalues* of the pair of matrices, and can be computed efficiently in the order of $(h_k)^3$ operations [66].

This process admits the following generalization.

Lemma 10 If $C_{[k,k]1}, \ldots, C_{[k,k]r}$ are covariant or contravariant matrices, and we write the homogeneous polynomial

$$\left| \theta_1 \, \mathcal{C}_{[k,k]1} + \cdots + \theta_r \, \mathcal{C}_{[k,k]r} \right|$$

of degree h_k in $\theta = (\theta_1, \ldots, \theta_r)^t$ in the normal form

$$\sum_{|\alpha|=h_k} rac{1}{lpha!} \, \mathcal{C}_{lpha} \, heta^{lpha} \; .$$

then, the coefficients C_{α} , which are polynomials in the components of the matrices, are new covariants. Explicitly, for every multiindex α , the coefficient C_{α} is an homogeneous polynomial of degree α_i in the components of the matrix $C_{[k,k]i}$.

Note that, if the entries of the matrices $C_{[k,k]1}, \ldots, C_{[k,k]r}$ are independent of space variables, for every choice of $\theta = (\theta_1, \ldots, \theta_r)^t$, the matrix $\theta_1 C_{[k,k]1} + \cdots + \theta_r C_{[k,k]r}$ is also covariant

and independent of space variables. Let $1 \le j < r$, and let us consider the following two new covariant matrices

$$\begin{array}{lll} \mathcal{C}_{[k,k]r+1} &=& \theta_1 \, \mathcal{C}_{[k,k]1} + \dots + \theta_j \, \mathcal{C}_{[k,k]j} \\ \mathcal{C}_{[k,k]r+2} &=& \theta_{j+1} \, \mathcal{C}_{[k,k]j+1} + \dots + \theta_r \, \mathcal{C}_{[k,k]r} \end{array}$$

The h_k generalized eigenvalues of this pair of matrices, the roots (λ_1, λ_2) of the binary form

$$\left| \lambda_1 \mathcal{C}_{[k,k]r+1} + \lambda_2 \mathcal{C}_{[k,k]r+2} \right|$$

are joint invariants of the *r* matrices, and can be computed efficiently in order of $(h_k)^3$ operations.

Polarization

The determinant of a square covariant or contravariant matrix is a covariant. We have seen that the determinant of an homogeneous combination of covariant or contravariant matrices of the same size defines many new covariants. More generally, if $C(\phi, x)$ is a covariant of a form ϕ of degree d, and it is an homogeneous polynomial of degree k in the coefficients of ϕ , then, for every set of the forms ϕ_1, \ldots, ϕ_r , the function

$$\mathcal{C}(\theta_1\phi_1+\cdots+\theta_r\phi_r,x)$$

is an homogeneous polynomial in $\theta = (\theta_1, \dots, \theta_r)^t$, and so, it can be written in normal form

$$\mathcal{C}(\theta_1\phi_1 + \dots + \theta_r\phi_r, x) = \sum_{|\alpha|=k} \frac{1}{\alpha!} \mathcal{C}_{\alpha} \theta^{\alpha} ,$$

where the coefficients C_{α} are functions of the coefficients of the forms and the vector of indeterminates. Furthermore,

Lemma 11 The coefficients $C_{\alpha}(\phi_1, \ldots, \phi_r, x)$ are joint covariants of the r forms ϕ_1, \ldots, ϕ_r , with the same weight, and for each $i = 1, 2, \ldots, r$, C_{α} is homogeneous of degree α_i in the coefficients of the form ϕ_i .

4.2.3 Construction of covariant and contravariant matrices

The basic tools for the construction of contravariant matrices are the partial differential operator matrices $D_{[k,j]}$. For every form ϕ of degree $d \ge k + j$, the matrix $D_{[k,j]}\phi$ is a covariant matrix.

If $C(\phi, x)$ is an absolute covariant of the form ϕ , which is homogeneous of degree $d \ge k + j$ in the variable x, then, the matrix of functions of the form ϕ and the vector x defined by

$$\mathcal{C}_{[k,j]} = D_{[k,j]}\mathcal{C} \tag{4.18}$$

is a contravariant matrix. But if the weight of the covariant is $w \neq 0$, then, the matrix (4.18) is not a contravariant matrix, because, if x' = Ax is a coordinate transformation, then

$$\mathcal{C}'_{[k,j]} = |A|^{-w} A^{-t}_{[k]} \left(D'_{[k,j]} \mathcal{C}' \right) A^{-1}_{[j]}$$

However, this matrix can be transformed into a covariant matrix by replacing the relative covariant with an absolute one. We can transform the relative covariant into an absolute covariant by dividing it by a relative invariant $\mathcal{I}(\phi)$ of the same weight w. For example, if ϕ is a form of even degree d = 2k, we can consider

$$\mathcal{I}_1(\phi) = \left| \Phi_{[k,k]}^2 \right|,$$

which is an invariant of weight $w_1 = 4\binom{n+k-1}{n}$. $\mathcal{I}_2(\phi) = \mathcal{I}_1(\phi)^{w/w_1}$ is an invariant of weight w, and so the matrix

$$\mathcal{C}_{[k,j]} = \frac{1}{\mathcal{I}_2(\phi)} D_{[k,j]} \mathcal{C}$$

is a contravariant matrix.

With respect to covariant matrices, if $C_{[k,k]}$ is a contravariant matrix, and $|C_{[k,k]}|$ is not identically zero, then $C_{[k,k]}^{-1}$ is a covariant matrix.

Now, we can construct new covariant and contravariant matrices alternating covariant and contravariant matrices of the proper sizes. For example, If ψ is a form of degree $\geq 2k$ such that $|D_{[k,k]}\psi(x)|$ is not identically zero, and ϕ and ξ are forms of degrees $\geq j+k$, then

$$\left(D_{[j,k]}\phi(x)\right)\left(D_{[k,k]}\psi(y)\right)^{-1}\left(D_{[k,j]}\xi(z)\right)$$

is a contravariant matrix, and

$$\mathcal{C}(\phi,\psi,\xi,x,y,z) = \left| \left(D_{[j,k]}\phi(x) \right) \left(D_{[k,k]}\psi(y) \right)^{-1} \left(D_{[k,j]}\xi(z) \right) \right|$$

is a joint rational covariant of the three forms and the three covariant vectors x, y, z. As a particular case of this, if ϕ is a form of degree d, and we set $\psi(x) = \phi(x)^2$, a form of degree 2d, if the matrix $\Psi_{[d,d]}$ is nonsingular, then

$$\mathcal{I}(\phi) = \Phi^{t}_{[d]} \Psi^{-1}_{[d,d]} \Phi_{[d]}$$

is an absolute invariant of the form ϕ , which can be computed using numerical methods.

4.2.4 Some examples of algebraic invariants

As applications of the techniques described above for the construction of covariants and invariants, we will show here several invariants of one form, and joint invariants of two forms, which can be computed efficiently. We are primarily interested in absolute invariants.

Joint invariants of two forms

If ϕ_1 and ϕ_2 are two forms of the same degree d = 2k we have already shown that the h_k generalized eigenvalues, roots of the homogeneous polynomial

$$\left| \theta_1 \Phi_{1[k,k]} + \theta_2 \Phi_{2[k,k]} \right| ,$$

are joint absolute invariants of the pair. The number of operations required to compute the matrices $\Phi_{1[k,k]}$ and $\Phi_{2[k,k]}$ is in the order of $(h_k)^2$ multiplications, and the generalized eigenvalues can be computed with in the order of $(h_k)^3$ operations.

If the degree is odd, or more generally, for every degree d, we can perform the same computation for the squares of the two polynomials, $\psi_1(x) = \phi_1(x)^2$, $\psi_2(x) = \phi_2(x)^2$ obtaining h_d generalized eigenvalues, the roots of the homogeneous polynomial

$$\left| \theta_1 \Phi_{1[k,k]} + \theta_2 \Phi_{2[k,k]} \right|$$

In this case, we have the extra cost of computing the square of the original polynomials, which is less expensive than the computation of eigenvalues. In the order of $(h_d)^2$ operations are sufficient.

We can also extend the previous procedure to the case of two forms ϕ_1 and ϕ_2 of different degrees d_1 and d_2 . If k is the minimum common multiple of d_1 and d_2 , $k_1 = k/d_1$, and $k_2 = k/d_2$, the two new forms $\psi_1(x) = \phi_1(x)^{2k_1}$ and $\psi_2(x) = \phi_2(x)^{2k_2}$ are forms of the same degree d = 2k. Now, we can compute the h_k generalized eigenvalues, roots of

$$\left| \theta_1 \Phi_{1[k,k]} + \theta_2 \Phi_{2[k,k]} \right|$$

which are absolute invariants of the pair of forms.

Absolute invariants of one form

Absolute invariants of one form ϕ of degree d can be computed, using the methods described in the previous paragraph, as joint invariants of ϕ and an absolute covariant $\psi(x) = C(\phi, x)$ of ϕ . The two forms ϕ and ψ have to be algebraically independent to produce nontrivial results. For example, choosing $\psi(x) = \phi(x)^k$ will not work. An absolute covariant can be obtained as a ratio of a relative covariant divided by a relative invariant of the same weight. The simplest covariants that we can construct with the methods described in this chapter are the *k*-Hessians (4.16). The *k*-Hessian $|D_{[k,k]}\phi|$ is a form of degree $(n - 2k)\binom{n+k-1}{n-1}$ if $1 \le k < d/2$, independent of x for k = d/2, and identically zero for k > d/2. We will only consider the cases $1 \le k < d/2$, because we need a form of positive degree. As an invariant we can take $|\Phi_{1[j,j]}|$, where $\phi_1 = \phi$, if d = 2j is even, and $\phi_1 = \phi^2$ if d is odd, in which case j = d. If one of the k-Hessians has even degree, we can also take ϕ_1 equal to this k-Hessians are computationally expensive to obtain. The absolute invariant will be defined only for those forms ϕ of degree d such that $|\Phi_{1[j,j]}| \neq 0$. The absolute

covariant will be

$$\psi = \frac{D_{[k,k]}\phi}{|\Phi_{1[j,j]}|^{w_1}}$$

with the constant w_1 properly choosen to make ψ an absolute covariant.

4.3 Cartesian invariants and covariants

Cartesian invariants of forms are defined in the same way as algebraic invariants of forms, but the group of transformations is restricted to be the orthogonal group.

The fundamental difference between algebraic and Cartesian invariants, is that there exist no invariant form under the action of the projective group, while there are plenty under the subgroup of orthogonal transformations. That is, there exists no nonzero form ϕ , such that $\phi(Ax) \equiv \phi(x)$ for every nonsingular matrix A, but $\phi(x) = ||x||^2$, and any function $f(||x||^2)$, is invariant under orthogonal transformations.

This means that there exists no linear differential operator invariant under the general linear group, but there are many invariant under the orthogonal group. In particular, there exists no linear invariant of a form under projective transformations, i.e., one which is a linear function of the coefficients of the form, but we will show many linear invariants under orthogonal transformations.

The *Laplacian* operator Δ , the differential operator corresponding to the form $||x||^2$ is a generator of the algebra of invariant linear differential operators. That is, every other invariant linear differential operator can be written as a polynomial in the Laplacian [76].

4.3.1 The invariant inner product of forms

If $\phi = \Phi_{[d]}^t X_{[d]}$ and $\psi = \Psi[d]^t X_{[d]}$ are two forms of degree d, the expression

$$\langle \phi, \psi \rangle = \Phi^t_{[d]} \Psi_{[d]} = \sum_{|\alpha|=d} \frac{1}{\alpha!} \Phi_{\alpha} \Psi \alpha$$

defines an *inner product* in the vector space of forms of degree d, which is invariant under orthogonal transformations of the space variables. It is a joint Cartesian invariant of two forms of degree d. If x' = Ax is an orthogonal transformation, then

$$\langle \phi', \psi' \rangle = \Phi'_{[d]}{}^{t} \Psi'_{[d]} = \Phi^{t}_{[d]} [A^{t}_{[d]} A_{[d]}] \Psi_{[d]} = \Phi^{t}_{[d]} \Psi^{t}_{[d]} = \langle \phi, \psi \rangle$$

because the matrix $A_{[d]}$ is orthogonal. In particular, the norm of ϕ

$$\|\phi\|^2 = \langle \phi, \phi \rangle = \sum_{\alpha=d} \frac{1}{\alpha!} \Phi_{\alpha}^2 = \Phi_{[d]}^t \Phi_{[d]} = \|\Phi_{[d]}\|^2.$$
(4.19)

is invariant under orthogonal transformations of the space variables. It is a Cartesian invariant of one form of degree d. For example, the norm of a second degree form in three variables

$$\phi(x_1, x_2, x_3) = \frac{1}{2} \Phi_{(2,0,0)} x_1^2 + \Phi_{(1,1,0)} x_1 x_2 + \Phi_{(1,0,1)} x_1 x_3 + \frac{1}{2} \Phi_{(0,2,0)} x_2^2 + \Phi_{(0,1,1)} x_2 x_3 + \frac{1}{2} \Phi_{(0,0,2)} x_3^2$$

$$\|\phi\|^2 = \frac{1}{2}\Phi_{(2,0,0)}^2 + \Phi_{(1,1,0)}^2 + \Phi_{(1,0,1)}^2 + \frac{1}{2}\Phi_{(0,2,0)}^2 + \Phi_{(0,1,1)}^2 + \frac{1}{2}\Phi_{(0,0,2)}^2.$$

4.3.2 Covariant matrices and the characteristic polynomial

As we have mentioned earlier, there is only one kind of covariant matrix with respect to orthogonal transformations. Furthermore, the transformation rules of the square covariant matrices correspond to conjugation, and so the characteristic polynomial of a square covariant matrix yields new absolute covariants

$$\left| \theta I - \mathcal{C}_{[k,k]} \right| = \theta^{h_k} + \sum_{i=1}^{h_k} (-1)^i \mathcal{C}_i \theta^{h_k - i} .$$

In particular, if the elements of the matrix are independent of the indeterminates, the h_k eigenvalues of the matrix are absolute invariants. For example, if ϕ is a matrix of even degree d = 2k, then, the h_k eigenvalues of the square matrix $\Phi_{[k,k]}$ are Cartesian invariants of the form ϕ . More generally, for every form ϕ of degree d, if we set $\psi(x) = \phi(x)^2$, the $h_{[d]}$ eigenvalues of the matrix $\Psi_{[d,d]}$ are Cartesian invariants of ϕ , but they are functionally dependent. The maximum number of functionally independent invariants of a form of degree d can not be larger than the number of coefficients minus the number of parameters of the orthogonal group O(n), i.e., not larger than $h_d - n(n-1)/2$.

As we did in the projective case, new covariant matrices can be constructed by multiplying two or more covariant matrices of matching sizes. For example, for every form ϕ of degree d = k + j, the following symmetric and nonnegative definite matrix is covariant

$$\Phi_{[k,j]} \Phi_{[j,k]} = \Phi^t_{[j,k]} \Phi_{[j,k]} ,$$

and its h_k nonnegative eigenvalues are invariants. For different values of k and j we do not obtain independent invariants though. For example, for k = 0 and j = d we obtain the invariant norm of ϕ , which can be seen to be equal to the trace of $\Phi_{[k,j]}\Phi_{[j,k]}$, for every value of j and k.

Joint invariants of two or more forms can be constructed in the same way. For example, if ϕ is a form of degree $d_1 = k + j$ and ψ is a form of degree $d_2 = j + l$, then, the matrix

$$\Phi_{[k,j]}\Psi_{[j,l]}$$

is covariant, and the maximum number of joint invariants is obtained by choosing j equal to the maximum between $\lceil d_1/2 \rceil$ and $\lceil d_2/2 \rceil$.

Note that the Laplacian is a particular case of the construction described in this section

$$\Delta \phi \; = \; \sum_{i=1}^n rac{\partial^2 \phi}{\partial x_i^2} \; = \; {
m trace} \; \left(D_{[1,1]} \phi
ight)$$

which is the coefficient corresponding to t^{n-1} of the characteristic polynomial

 $\left| \theta I - D_{[1,1]} \phi \right|$.

is

And also because

$$||x||^2 = \text{trace}(X_{[1,1]}) = \text{trace}(xx^t).$$

Similarly, it is not difficult to see that the consecutive powers of the Laplacian can be obtained in the same way

$$\frac{1}{k!} \Delta^k \phi = \operatorname{trace} \left(D_{[k,k]} \phi \right) \;,$$

because

trace
$$(X_{[k,k]}) = ||X_{[k]}^t||^2 = \frac{1}{k!} ||x||^{2k}$$
,

and every other invariant linear differential operator is a linear combination of these

$$\sum_{k=0}^{\infty} \frac{1}{k!} p_k \,\Delta^{2k} \phi \;,$$

where only finitely many coefficients are nonzero.

4.3.3 The harmonic decomposition

The inner product defined above can also be written in the following way

$$\langle \phi, \psi \rangle = \phi(D)\psi = \psi(D)\phi$$
, (4.20)

where $\phi(D)$ and $\psi(D)$ are the linear differential operators associated with the forms $\phi(x)$ and $\psi(x)$ of degree $d \ge 0$, because, by linearity

$$\phi(D)\psi = \sum_{\alpha=d} \frac{1}{\alpha!} \Phi_{\alpha} D^{\alpha} \left(\sum_{\beta=d} \frac{1}{\beta!} \Psi_{\beta} x^{\beta} \right)$$
$$= \sum_{\alpha=d} \sum_{\beta=d} \frac{1}{\alpha!\beta!} \Phi_{\alpha} \Psi_{\beta} D^{\alpha}(x^{\beta}) = \sum_{\alpha=d} \frac{1}{\alpha!} \Phi_{\alpha} \Psi_{\alpha} = \langle \phi, \psi \rangle .$$

And since the composition of two linear differential operators corresponds to the product of the corresponding polynomials, we obtain the following fundamental property of the Laplacian

Lemma 12 Let ψ be a form of degree $d \ge 2$ and ξ a form of degree d - 2. Then,

$$\langle \psi, \|x\|^2 \xi \rangle = \langle \Delta \psi, \xi \rangle,$$

where the left side is the invariant inner product of forms of degree d, and the right side is the invariant inner product of forms of degree d - 2.

If the Laplacian of a form is identically zero, the form is called *harmonic*. Since the Lalacian of a form is a covariant, the property of being harmonic is clearly invariant under orthogonal coordinate transformations, and the set of all harmonic forms of degree d determines a *stable* subspace of the vector space of forms of degree d, with respect to the action

of the orthogonal group. The subspace of harmonic forms is also *irreducible*, meaning that it cannot be written as the sum of two stable subspaces, unless one of them is trivial, i.e., the zero subspace. A total decomposition of the vector space of forms of degree d as a sum of stable and irreducible subspaces would yield new invariants, and in the rest of this section we will show how to efficiently compute this harmonic decomposition. Towards this goal, the first result is the following

Lemma 13 The subspaces of harmonic forms of degree d and of forms of degree d divisible by $||x||^2$, are complementary with respect to the invariant inner product, i.e., for every form ϕ of degree d there exist two unique forms, ψ harmonic of degree d, and ξ of degree d - 2, such that

$$\phi(x) = \psi(x) + ||x||^2 \xi(x) .$$

And if we apply this lemma recursively, we obtain the following result

Corollary 5 Every form ϕ of degree d has a unique decomposition

$$\phi(x) = \sum_{i=0}^{\lfloor d/2 \rfloor} \|x\|^{2i} \phi_i(x) ,$$

where ϕ_i is an harmonic form of degree d - 2i.

Furthermore, the subspaces determined by this decomposition are orthogonal to each other, because

Lemma 14 Let ψ be a harmonic form of degree $d \ge 2$, ξ a form of degree d - 2, and k a nonnegative integer. Then,

$$\langle ||x||^{2k}\psi, ||x||^{2k+2}\xi \rangle = 0.$$

Finally, these subspaces are known to be irreducible [76, Introduction, theorem 3.1], but we omit the proof because it is not necessary for our purposes.

After computing the harmonic decomposition, many new absolute invariants can be obtained using the other techniques described above. For example, the norms of the harmonic components

$$\|\phi_i\|^2$$
 $i = 0, 1, \dots, \lfloor d/2 \rfloor$

of Corollary 5 are absolute invariants.

Our last task in this chapter is to present an efficient algorithm for the computation of this harmonic decomposition.

4.3.4 Computing the harmonic decomposition

Let us denote \mathcal{H}_d and \mathcal{H}_d^0 , the space of forms of degree d, and its subspace of harmonic forms, respectively. We start by presenting a recursive algorithm based on Corollary 5. We rewrite the decomposition as a recursion

$$\begin{cases} \psi_0 &= \phi \\ \Delta \psi_i &= \Delta(\|x\|^2 \psi_{i+1}) \qquad i = 0, 1, \dots, \lfloor d/2 \rfloor, \end{cases}$$
(4.21)

where

$$\psi_j = \sum_{i=j}^{\lfloor d/2 \rfloor} \|x\|^{2(i-j)} \phi_i \qquad j = 0, 1, \dots, \lfloor d/2 \rfloor,$$

and then observe that, based on the uniqueness of the decomposition

$$\psi_i = \phi_i + \|x\|^2 \psi_{i+1} \qquad \Delta \phi_{d-2i} = 0 ,$$

and for every value of ψ_i , the second equation of (4.21) has a unique solution ψ_{i+1} . In fact, **Lemma 15** *The linear operator*

$$\begin{cases} \mathcal{L}_d: \mathcal{H}_d \to \mathcal{H}_d \\ \psi \mapsto \Delta(\|x\|^2 \psi) \end{cases}$$

is 1-1 and onto, and so, for every form ϕ , the linear equation

$$\Delta(\|x\|^2\psi) = \phi$$

has a unique solution.

The following algorithm recursively computes the harmonic decomposition of the form ϕ of degree d

If this decomposition has to be computed for many forms of the same degree d it is more appropriate to compute it for the basis vector $X_{[d]}$, or equivalently for the monomials, and then use the linearity of the decomposition. Now, since the decomposition is invariant under orthogonal transformations of the space variables, and permutations of variables are orthogonal transformations, it is suficient to compute the decomposition for those monomials x^{α} , with $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$, and then obtain the others by the corresponding permutations of coefficients. After finishing this process we will obtain the matrices

$$P_{[d],i}$$
 $i = 0, 1, \dots, \lfloor d/2 \rfloor$

of the orthogonal projectors $\mathcal{H}_d \to \|x\|^{2i} \mathcal{H}_{d-2i}^0$, that is, if $\phi = \Phi_{[d]}^t X_{[d]}$ then

$$||x||^2 \phi_{d-2i} = [P_{[d],i} \Phi_{[d]}]^t X_d$$

The rank of $P_{[d],i}$ is equal to $h_{d-2i} - h_{d-2i-2}$, the dimension of $||x||^{2i} \mathcal{H}^0_{d-2i}$. We can obtain an orthonormal basis of $||x||^{2i} \mathcal{H}^0_{d-2i}$ by applying the QR algorithm [66] to the matrix $P_{[d],i}$, that is, we orthonormalize the columns of $P_{[d],i}$ by performing Householder rotations, and obtain a decomposition of the form

$$P_{[d],i} = Q_{[d],i}R_{[d],i}$$

where $Q_{[d],i}$ is orthogonal, and $R_{[d],i}$ is upper triangular. Since the matrix $P_{[d],i}$ is not full rank, only the upper $h_{d-2i} - h_{d-2i-2}$ rows of $R_{[d],i}$ will be nonzero, and it will not be necessary to do a full decomposition. It will be sufficient to do as many rotations as the rank of $P_{[d],i}$. If $Q_{[d],i}^0$ is the submatrix of the first $h_{d-2i} - h_{d-2i-2}$ columns of $Q_{[d],i}$. These columns form an orthonormal basis of $||x||^{2i}\mathcal{H}_{d-2i}^0$. Finally we pack these matrices into a single orthogonal matrix

$$Q_{[d]} \;=\; \left(Q^0_{[d],0} | Q^0_{[d],2} | \cdots | Q^0_{[d],\lfloor d/2
floor}
ight)$$

which we store. The elements of the vector of forms

$$Y_{[d]}(x) = Q_{[d]}^{t} X_{[d]}(x)$$

form a new orthonormal basis of \mathcal{H}_d , the first $h_d - h_{d-2}$ elements define a basis of \mathcal{H}_d^0 , the following $h_{d-2} - h_{d-4}$ elements define a basis of $||x||^2 \mathcal{H}_{d-2}^0$, and so on. The decomposition of a form $\phi = \Phi_{[d]}^t X_{[d]}$ has been reduced to the multiplication of the vector of coefficients by the orthogonal matrix described above

$$\Phi_{[d]} \quad \mapsto \quad Q_{[d]}^t \Phi_{[d]} \; .$$

The first $h_d - h_{d-2}$ elements of this vector are the coefficients of ϕ_0 in the basis defined by $Y_{[d]}$, the following $h_{d-2} - h_{d-4}$ elements are the coefficients of $||x||^2 \phi_1$, and so on.

4.4 **Proofs**

Proof of Lemma 1:

Since we have the 1-1 correspondence

$$\{\alpha : |\alpha| = d, \alpha_1 \ge \eta_1, \dots, \alpha_n \ge \eta_n\} = \{\beta + \eta : |\beta| = d - k\}, = \{\beta : |\beta| = d - k\} + \eta,$$

from (4.12) we obtain

$$D^{\eta}\phi(x) = \sum_{|\alpha|=d} \frac{1}{\alpha!} \Phi_{\alpha} D^{\eta}(x^{\alpha}) = \sum_{|\beta|=d-k} \frac{1}{\beta!} \Phi_{\beta+\eta} x^{\beta}$$

or equivalently

$$\sqrt{\frac{1}{\eta!}} D^{\eta} \phi(x) = \sum_{|\beta|=d-k} \left(\sqrt{\frac{1}{\beta! \eta!}} \Phi_{\beta+\eta} \right) \left(\sqrt{\frac{1}{\beta!}} x^{\beta} \right)$$

which, if written in matrix form, is the desired result.

Proof of Lemma 2:

Since the polynomial ϕ is homogeneous of degree d, we have the identity

$$\theta^d \phi(x) \equiv \phi(\theta x)$$

in n + 1 variables θ, x_1, \ldots, x_n . Differentiating k times with respect to t, and using the chain rule, we obtain

$$\binom{d}{k} \theta^{d-k} \phi(x) = \sum_{|\eta|=k} \frac{1}{\eta!} D^{\eta} \phi(tx) x^{\eta} = X^{t}_{[k]}(x) D_{[k]} \phi(\theta x) .$$

Finally, we evaluate the previous expression at $\theta = 1$ and substitute the vector of partial derivatives according to Lemma 1

$$\binom{d}{k}\phi(x) = X_{[k]}^t(x)D_{[k]}\phi(x) = X_{[k]}^t(x)\Phi_{[k,d-k]}X_{[d-k]}(x).$$

Proof of Lemma 3 : The *multinomial formula* is

$$\frac{1}{d!}(x_1 + \dots + x_n)^d = \sum_{|\alpha|=d} \frac{1}{\alpha!} x^{\alpha}.$$

Let *x* and *y* be two *n*-dimensional vectors, and let us consider the multinomial expansion of the *d*-th power of the inner product $y^t x$, the polynomial of 2n variables

$$\frac{1}{d!} (y^t x)^d = \frac{1}{d!} (y_1 x_1 + \dots + y_n x_n)^d = \sum_{|\alpha|=d} \frac{1}{\alpha!} (y_1 x_1)^{\alpha_1} \cdots (y_n x_n)^{\alpha_n} = \sum_{|\alpha|=d} \frac{1}{\alpha!} y^{\alpha} x^{\alpha}.$$

This polynomial is homogeneous of degree d in both x and y, and it is obviously invariant under simultaneous orthogonal transformations of the variables x-y. In vector form,

$$\frac{1}{d!} (y^t x)^d = X_{[d]}(y)^t X_{[d]}(x) .$$

1a.) Let A and B be $n \times n$ nonsingular matrices. Then, the following expression

$$(AB)_{[d]}X_{[d]}(x) = X_{[d]}((AB)x) = X_{[d]}(A(Bx)) = = A_{[d]}X_{[d]}(Bx) = A_{[d]}(B_{[d]}X_{[d]}(x)) = (A_{[d]}B_{[d]})X_{[d]}(x)$$

is a polynomial identity, and all the coefficients of the polynomials on the left side are identically to the corresponding coefficients of the polynomials on the right side, that is

$$(AB)_{[d]} = (A_{[d]}B_{[d]}).$$

1b.) Follows from the uniqueness of representation of a homogeneous polynomial as a linear combination of monomials (4.9).

1c.) From 1b.), the identity matrix is map to the identity matrix. Let *A* be a $n \times n$ nonsingular matrix. Apply 1a.) with $B = A^{-1}$ to obtain

$$I = (AA^{-1})_{[d]} = A_{[d]}(A^{-1})_{[d]} \implies (A_{[d]})^{-1} = (A^{-1})_{[d]}.$$

2.) Let A be a $n \times n$ nonsingular matrix Then, the following expression

$$0 = \frac{1}{d!} [((Ay)^{t}x)^{d} - (y^{t}(A^{t}x))^{d}] = X_{[d]}(Ay)^{t}X_{[d]}(x) - X_{[d]}(y)X_{[d]}(A^{t}x) = X_{[d]}(y)^{t} ((A_{[d]})^{t} - (A^{t})_{[d]})X_{[d]}(x)$$

is a polynomial identity, and all the coefficients of the polynomial on the right side are identically zero, that is

$$(A^t)_{[d]} = (A_{[d]})^t$$
.

2a.) If *A* is symmetric, we have

$$(A_{[d]})^t = (A^t)_{[d]} = A_{[d]}.$$

2b.) The matrix *A* is symmetric positive definite, if and only if we can write $A = BB^t$, for certain nonsingular $n \times n$ matrix *B*. Then

$$A_{[d]} = (BB^t)_{[d]} = B_{[d]}B^t_{[d]}$$

and so $A_{[d]}$ is positive definite as well. 2c.) If A is orthogonal, we have

$$(A_{[d]})^{-1} = (A^{-1})_{[d]} = (A^t)_{[d]} = (A_{[d]})^t$$

Proof of Lemma 4 :

1.) If α and β are two multiindices of size d, the (α, β) -th element of the matrix $A_{[d]}$ is

$$\sqrt{\frac{1}{\alpha!\,\beta!}}\,D^{\beta}((Ax)^{\alpha})$$
.

If β follows α in the lexicographical order, then, for certain 1 < k < n we have

$$\alpha_1 = \beta_1, \ldots, \alpha_{k-1} = \beta_{k-1}, \alpha_k > \beta_k ,$$

and so

$$\alpha_{k+1} + \dots + \alpha_n < \beta_{k+1} + \dots + \beta_n.$$

Since the matrix *A* is lower triangular, the degree of

$$(Ax)^{\alpha} = \prod_{i=1}^{n} \left(\sum_{j=1}^{i} a_{ij} x_j \right)^{\alpha_i}$$

as a polynomial in x_{k+1}, \ldots, x_n with coefficients polynomials in x_1, \ldots, x_k is clearly not greater than $\alpha_{k+1} + \cdots + \alpha_n$, and so

$$\left(\frac{\partial}{\partial x_{k+1}}\right)^{\beta_{k+1}} \cdots \left(\frac{\partial}{\partial x_n}\right)^{\beta_n} \left((Ax)^{\alpha}\right) = 0.$$

It follows that $D^{\beta}((Ax)^{\alpha}) = 0$, and the matrix $A_{[d]}$ is lower triangular.

2.) First note that for every $1 \le k \le n$ the variable x_k appears only in the last term of the product

$$\prod_{i=1}^k \left(\sum_{j=1}^i a_{ij} x_j\right)^{\alpha_i} ,$$

and so

$$\left(\frac{\partial}{\partial x_k}\right)^{\alpha_k} \left(\prod_{i=1}^k \left(\sum_{j=1}^i a_{ij} x_j\right)^{\alpha_i}\right) = \left(\prod_{i=1}^{k-1} \left(\sum_{j=1}^i a_{ij} x_j\right)^{\alpha_i}\right) \alpha_k! a_{kk}^{\alpha_k}$$

By induction in k = n, n-1, ..., 1, it follows that the α -th element of the diagonal of $A_{[d]}$ is

$$\frac{1}{\alpha!}D^{\alpha}((Ax)^{\alpha}) = a_{11}^{\alpha_1}\cdots a_{nn}^{\alpha_n} = a^{\alpha_n}$$

3.) Since *A* is triangular, $|A| = a_{11} \cdots a_{nn}$. From 2.) we have

$$|A_{[d]}| = \prod_{|\alpha|=d} a^{\alpha} = a^{\gamma} ,$$

where $\gamma = \sum_{|\alpha|=d} \alpha$. By symmetry, all the components of the multiindex γ are equal, and so, for every $1 \le i \le n$

$$\gamma_i = \sum_{|\alpha|=d} \alpha_i = \frac{1}{n} \sum_{i=1}^n \sum_{|\alpha|=d} \alpha_i = \sum_{|\alpha|=d} = \frac{d}{n} \binom{n+d-1}{n-1} = \binom{n+d-1}{n} = m.$$

Finally

$$|A_{[d]}| = \left(\prod_{i=1}^{n} a_{ii}\right)^{m} = |A|^{m}$$

Proof of Corollary 1 :

For every matrix A, there exist an orthogonal matrix Q, and a lower triangular matrix L such that A = LQ. Since the map $A \mapsto A_{[d]}$ is a homomorphism, we have $A_{[d]} = L_{[d]}Q_{[d]}$, where $L_{[d]}$ is lower triangular and $Q_{[d]}$ is orthogonal, i.e., the decomposition is preserved. It follows that

$$|A_{[d]}| = |L_{[d]}| = |L|^m = |A|^m$$
.

Proof of Corollary 2 :

Apply Lemma 4 to both A and $A^t = A$.

Proof of Lemma 5:

1.) Since x' = Ax, the partial derivatives are related by the chain rule

$$\frac{\partial}{\partial x_i} = \sum_{j=1}^n \frac{\partial x'_j}{\partial x_i} \frac{\partial}{\partial x'_j} = \sum_{j=1}^n A_{ji} \frac{\partial}{\partial x'_j} ,$$

or in matrix form, $\,D'=(\partial/\partial x_1',\ldots,\partial/\partial x_n')^t=A^{-t}D$. Then

$$D'_{[k]} = X_{[k]}(D') = X_{[k]}(A^{-t}D) = (A^{-t})_{[k]}X_{[k]}(D) = A^{-t}_{[k]}D_{[k]}$$

2.)

$$D'_{[k,j]} = X_{[k]}(D')X^{t}_{[j]}(D') =$$

= $X_{[k]}(A^{-t}D)X^{t}_{[j]}(A^{-t}D) =$
= $A^{-t}_{[k]}X_{[k]}(D)X^{t}_{[j]}(D)A^{-1}_{[j]} = A^{-t}_{[k]}D_{[k,j]}A^{-1}_{[j]}$

Proof of Corollary 3:

1.)
$$\Phi'_{[d]} = D'_{[d]}\phi'(x') = A^{-t}_{[d]}D_{[d]}\phi(x) = A^{-t}_{[d]}\Phi_{[d]}$$

2.) $\Phi'_{[k,d-k]} = D'_{[k,d-k]}\phi'(x') = A^{-t}_{[k]}\left(D_{[k,d-k]}\phi(x)\right)A^{-1}_{[d-k]} = A^{-t}_{[k]}\Phi_{[k,d-k]}A^{-1}_{[d-k]}$.

Proof of Lemma 6:

If $C_{[1,1]}$ is a covariant matrix, and x' = Ax is a coordinate transformation, we have

$$\mathcal{C}'_{[1,1][k]} = \left(A \, \mathcal{C}_{[1,1]} \, A^t \right)_{[k]} = A_{[k]} \, \mathcal{C}_{[1,1][k]} \, A^t_{[k]} \, .$$

Similar derivations apply to the other cases.

Proof of Lemma 9 :

If $C_{[k,k]}$ is a covariant matrix, then

$$C'_{[k,k]} = A^{-t}_{[k]} C_{[k,k]} A^{-1}_{[k]}$$

Computing determinants we obtain

$$\left| \mathcal{C}'_{[k,k]} \right| = \left| A^{-t}_{[k]} \right| \left| \mathcal{C}_{[k,k]} \right| \left| A^{-1}_{[k]} \right| ,$$

and according to Corollary 1, we obtain

$$\left| \mathcal{C}_{[k,k]}' \right| \; = \; |A|^{-2m} \left| \mathcal{C}_{[k,k]} \right|$$

where $m = \binom{n+k-1}{m}$. The proof for contravariant matrices is similar.

 Proof of Lemma 10 :

 Particular case of Lemma 11.

Proof of Lemma 11 :

The transformation rules are satisfied, because

$$\mathcal{C}(\phi', x') = |A|^{-w} \mathcal{C}(\phi, x)$$

for every form ϕ of degree d, in particular for the form $\phi = \theta_1 \phi_{(1)} + \cdots + \theta_r \phi_{(r)}$. Since in this case $\phi' = \theta_1 \phi'_{(1)} + \cdots + \theta_r \phi'_{(r)}$, we have the polynomial identity in $\theta = (\theta_1, \ldots, \theta_r)^t$

$$\sum_{|\alpha|=k} \frac{1}{\alpha!} \mathcal{C}_{\alpha}(\phi'_{(1)}, \dots, \phi'_{(r)}, x') \theta^{\alpha} =$$

$$\mathcal{C}(\theta_{1}\phi'_{(1)} + \dots + \theta_{r}\phi'_{(r)}, x') = |A|^{-w} \mathcal{C}(\theta_{1}\phi_{(1)} + \dots + \theta_{r}\phi_{(r)}, x)$$

$$= \sum_{|\alpha|=k} \frac{1}{\alpha!} \left(|A|^{-w} \mathcal{C}_{\alpha}(\phi_{(1)}, \dots, \phi_{(r)}, x') \right) \theta^{\alpha} ,$$

which is equivalent to the identity of the corresponding coefficients

$$\forall \alpha \qquad : \qquad \mathcal{C}_{\alpha}(\phi'_{(1)},\ldots,\phi'_{(r)},x') = |A|^{-w} \mathcal{C}_{\alpha}(\phi_{(1)},\ldots,\phi_{(r)},x)$$

By symmetry, we only need to prove the second part for i = 1. It follows by a similar argument applied to the identity

$$\sum_{\alpha|=k} \frac{1}{\alpha!} \mathcal{C}_{\alpha}(\lambda \phi_{(1)}, \dots, \phi_{(r)}, x') \theta^{\alpha} = \mathcal{C}(\theta_{1}(\lambda \phi_{(1)}) + \dots + \theta_{r} \phi_{(r)}, x) = \mathcal{C}((\theta_{1}\lambda)\phi_{(1)} + \dots + \theta_{r} \phi_{(r)}, x)$$
$$= \sum_{|\alpha|=k} \frac{1}{\alpha!} \mathcal{C}_{\alpha}(\phi_{(1)}, \dots, \phi_{(r)}, x') \lambda^{\alpha_{1}} \theta^{\alpha}$$

where λ is a new variable, which is equivalent to

$$\forall \alpha \qquad : \qquad \mathcal{C}_{\alpha}(\lambda \phi_{(1)}, \dots, \phi_{(r)}, x') = \lambda^{\alpha_1} \mathcal{C}_{\alpha}(\phi_{(1)}, \dots, \phi_{(r)}, x')$$

for every multiindex α .

Proof of Lemma 12 :

Let ψ be a form of degree $d \ge 2$, and let ξ be a form of degree d - 2. Let $\zeta(x) = ||x||^2 \xi(x)$, then, $\zeta(D) = \Delta \xi(D) = \xi(D)\Delta$, and so

$$\langle \|x\|^2 \xi, \psi \rangle = \langle \zeta, \psi \rangle = \zeta(D)\psi = \xi(D) (\Delta \psi) = \langle \xi, \Delta \psi \rangle.$$

Proof of Lemma 13 :

The Laplacian of a form ϕ of degree d is

$$\Delta\phi(x) = \sum_{i=1}^{n} D^{2e_i} \left(\sum_{|\alpha|=d} \frac{1}{\alpha!} \Phi_{\alpha} x^{\alpha} \right) = \sum_{|\beta|=d-2} \frac{1}{\beta!} \left(\sum_{i=1}^{n} \Phi_{\beta+2e_i} \right) x^{\beta}.$$

So, it is harmonic if and only if it satisfies the implicit equations

$$0 = \sum_{i=1}^{n} \Phi_{\beta+2e_i} \qquad |\beta| = d-2 .$$

But for each multiindex β of size d - 2, we have

$$\sum_{i=1}^{n} \Phi_{\beta+2e_i} = \langle \phi, \|x\|^2 x^\beta \rangle$$

and since $\{||x||^2 x^\beta : |\beta| = d - 2\}$ is a basis of the subspace of forms of degree d divisible be $||x||^2$, we have shown that this subspace is complementary of the subspace of harmonic forms f degree d.

Proof of Lemma 14 :

By induction in k. For k = 0 it is just Lemma 12. For k > 0, by the same lemma we have

$$\langle \|x\|^{2k}\psi, \|x\|^{2k+2}\xi \rangle = \langle \Delta(\|x\|^{2k}\psi), \|x\|^{2k}\xi \rangle$$

but, since ψ is an harmonic form of degree d, by the chain rule and Euler's theorem

$$\begin{array}{lll} \Delta(\|x\|^{2k}\psi) &=& \Delta(\|x\|^{2k})\psi + 2\nabla(\|x\|^{2k})^t\nabla\phi + \|x\|^{2k}\Delta\psi \\ &=& (2k(2k-2+n)+2d) \ \|x\|^{2k-2}\psi \end{array}$$

because

$$\begin{cases} \nabla(\|x\|^{2k}) &= 2k\|x\|^{2k-2}x\\ \Delta(\|x\|^{2k}) &= 2k(2k-2+n)\|x\|^{2k-2}\\ x^t\nabla\psi &= d\psi \ . \end{cases}$$

It follows that

$$\langle \|x\|^{2k}\psi, \|x\|^{2k+2}\xi \rangle = (2k(2k-2+n)+2d) \langle \|x\|^{2k-2}\psi, \|x\|^{2k}\xi \rangle,$$

which, by the inductive hypothesis, finishes the proof. \Box

Proof of Lemma 15 :

It is sufficient to prove that the matrix of \mathcal{L}_d in the basis of monomials is positive definite. Now, by Lemma 12, the element (α, β) of this matrix is

$$\langle \Delta(\|x\|^2 x^{\alpha}), x^{\beta} \rangle = \langle \|x\|^2 x^{\alpha}, \|x\|^2 x^{\beta} \rangle$$

This matrix is a constant times the matrix of inner products of the elements of the basis $\{||x||^2 x^{\alpha} : |\alpha| = d\}$ of the subspace $||x||^2 \mathcal{H}_d$ of \mathcal{H}_{d+2} , and so, positive definite.

Chapter 5

Geometric Matching of Algebraic Curves and Surfaces

In this chapter we are concerned with the positioning of algebraic curves and surfaces. Given to curves or surfaces of the same degree, which are almost the same, but in different positions and orientations, we want to recover the Euclidean transformation which transforms the first curve or surface into the second one. At the same time, since finite collections of curves or surfaces can be represented by a single curve or surface by multiplying the corresponding polynomials, we will solve the apparently more complex problem of recovering the transformation which aligns two finite sets of curves or surfaces, without knowing a priori which element of the first set corresponds to which element of the second set.

Our solution to these problems is to define for every algebraic curve or surface an intrinsic, or *covariant* according to the general definition given in the previous chapter, frame of reference. By this we mean, a center and an orthonormal basis which are functions of the coefficients of the polynomials that define the curve or surface, and are rigidly attached to it. This *intrinsic frame of reference* is commonly referred to as the *object coordinate system* in the Computer Vision literature. In our object coordinate system, the object center is at the origin, and the orthonormal basis for the object coincides with the coordinate unit vectors. The coefficients of the polynomials recomputed with respect to their intrinsic frames of reference, i.e., their object coordinate systems, are new Euclidean invariants, and we can also use these new coefficients to decide whether there is a match or not. If the two curves or surfaces are the same, their coefficient vectors should be the same, except for a nonzero multiplicative constant. Then, the rotation and translation which transforms the first curve or surface into the second one can be easily computed from the corresponding intrinsic frames of reference.

Although the algorithms for computing the center and canonical orientation will produce results for every curve and surface, from the numerical point of view, in certain singular cases small perturbations in the coefficients of the polynomials will introduce large perturbations in the coordinates of the center and the canonical orientation, i.e., certain curves and surfaces will not be *well conditioned* with respect to the computation of the center and canonical orientation. The classification based on comparing the coefficients of the polynomials with respect to the intrinsic frame of reference, and the recovered position, will not be accurate in these cases. However, we can reliably detect this problem by computing certain invariants of the homogeneous term of highest degree of the polynomials, which is invariant under translation, and, if the determination of the intrinsic frame of reference is not well conditioned, execute an alternative procedure. The alternative procedure will consist on imposing certain linear constraints on the coefficients of the polynomials, a regularization process, and then compute the intrinsic frame of reference reliably for the constrained curve or surface.

If x' = Ax + b is a nonsingular Euclidean coordinate transformation, and f(x) is a polynomial, we will denote by f'(x') the polynomial which defines the same curve or surface in the new coordinate system i.e., $f'(x') = f(A^{-1}(x' - b))$. Both Z(f') and Z(f) describe the same set of points which have different coordinates in the two different coordinate systems. The intrinsic frame of reference for the polynomial f will be defined by a matrix $A = A_f$ and a vector $b = b_f$, functions of the coefficients of f, such that Z(f') will be located in a *canonical position and orientation*. Then, if Z(f') and Z(g') are in the canonical positions for the polynomials f and g, respectively, we can say that Z(f) and Z(g) match exactly if $f'(x') \equiv \lambda g'(x')$, for certain nonzero multiplicative constant λ , where $f'(x') = f(A_f^{-1}(x' - b_f))$ and $g'(x') = g(A_g^{-1}(x' - b_g))$. With this approach, we will also be able to make approximate matches by comparing the vectors of coefficients of the two polynomials. Since two polynomials which differ by a multiplicative constant define the same curve or surface, we can chose an inner product $\langle \cdot, \cdot \rangle$ in the vector space of coefficients of polynomials of degree $\leq d$, and use the following number as a *matching measure*

$$\mathrm{match}\,(f,g) \;=\; \frac{\langle f',g'\rangle^2}{\|f'\|^2\|g'\|^2} \;,$$

where as usual $||f||^2 = \langle f, f \rangle$ is the norm of f with respect to the given inner product. This inner product should include appropriate weighting for the different components of the polynomial coefficient vectors. Note that $0 \leq \operatorname{match}(f, g) \leq 1$ and for an exact match

$$match(f,g) = 1 \qquad \Leftrightarrow \qquad \exists \lambda \neq 0 : f'(x') \equiv \lambda g'(x').$$

Alternatively, we can define a probability distribution in the projective space of coefficients of polynomials as a function of the data base of objects, and classify curves or surfaces using statistical decision methods. This treatment applies to a 2D curve or surface defined by a single polynomial, but we will extend it to 3D curves as well.

As examples of the methods to be described in this chapter, figures (5.1-a) and (5.1-c) show two cubic 2D curves given by the union of three straight lines extracted from the edge images, and figures (5.1-b) and (5.1-d) show their corresponding frames of reference. Figures (5.2-a) and (5.2-c) show two irreducible fourth degree 2D curves, and figures (5.2-b) and (5.2-d) show their corresponding frames of reference.

We will decompose the computation of the intrinsic frame of reference in two parts, the translation vector or *center*, and then the rotation matrix or *canonical orientation*.



Figure 5.1: (a): Cubic 2D curve, union of the three straight lines fitted to the data points in the dark region. (b): Intrinsic frame of reference for the curve in (a). (c): Cubic 2D curve, union of the three straight lines fitted to the data points in the dark region, matching the curve in (a). (d): Intrinsic frame of reference for the curve in (c).



Figure 5.2: (a): Interest region, quartic 2D curve fitted to the data points inside the circle. (b): Intrinsic frame of reference for the curve in (a). (c): Interest region, quartic 2D curve fitted to the data points inside the circle, matching the curve in (a). (d): Intrinsic frame of reference for the curve in (c).
5.1 The center of a planar curve or surface

Our definition of *center* of a 2D curve or 3D surface, defined as the set of zeros of a single polynomial of degree $d \ge 2$, is a generalization to d > 2 of the well known case of a non-singular quadratic curve or surface. For example, a quadratic polynomial of two variables can be written as a sum of forms

$$f(x) = f_2(x) + f_1(x) + f_0 , \qquad (5.1)$$

where

$$\begin{cases} f_2(x_1, x_2) &= \frac{1}{2} F_{(2,0)} x_1^2 + F_{(1,1)} x_1 x_2 + \frac{1}{2} F_{(0,2)} x_2^2 &= F_{[2]}^t X_{[2]}(x_1, x_2) \\ f_1(x_1, x_2) &= F_{(1,0)} x_1 + F_{(0,1)} x_2 &= F_{[1]}^t X_{[1]}(x_1, x_2) \\ f_0 &= F_{(0,0)} &= F_{[0]} \end{cases}$$

The second degree polynomial g(x) = f(x+y) can also be written as a sum of homogeneous polynomials $g(x) = g_2(x) + g_1(x) + g_0$, where $g_2(x) = f_2(x)$, $g_0 = f(y)$, and

$$g_1(x_1, x_2) = [F_{(1,0)} + F_{(2,0)} y_1 + F_{(1,1)} y_2] x_1 + [F_{(0,1)} + F_{(1,1)} y_1 + F_{(0,2)} y_2] x_2.$$

The *center* of Z(f) is the vector y which makes the linear term $g_1(x)$ identically zero. It is defined only if the matrix

$$F_{[1,1]} = \begin{pmatrix} F_{(2,0)} & F_{(1,1)} \\ F_{(1,1)} & F_{(0,2)} \end{pmatrix}$$

is nonsingular, in which case it is given by the following expresion

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = -\begin{pmatrix} F_{(2,0)} & F_{(1,1)} \\ F_{(1,1)} & F_{(0,2)} \end{pmatrix}^{-1} \begin{pmatrix} F_{(1,0)} \\ F_{(0,1)} \end{pmatrix} = -F_{[1,1]}^{-1}F_{[1,0]} , \qquad (5.2)$$

where the matrix $F_{[1,1]}$ is computed from the coefficients of the form f_2 and $F_{[1,0]}$ from the coefficients of f_1 as we have described in the previous chapter.

5.1.1 General case

Every polynomial f of degree d can be written in a unique way as

$$f(x) = \sum_{k=0}^{d} f_k(x) ,$$

where f_k is a form of degree k, and f_d is not identically zero. For every fixed space vector y, the translated polynomial g(x) = f(x + y), as a polynomial in x, has exactly the same degree d, and so, it can also be written in a unique way as a sum of homogeneous polynomials

$$f(x+y) = g(x) = \sum_{k=0}^{d} g_k(x) ,$$

where the coefficients of g_k are polynomials of degree d - k in y. Expanding the homogeneous terms of f in Taylor series at x, and regrouping terms corresponding to the same degrees, we obtain that the term of degree d is invariant under translation

$$g_d \equiv f_d$$
,

and the term of degree d - 1 is given by

$$g_{d-1} \equiv f_{d-1} + y^t \nabla f_d = f_{d-1} + \sum_{i=1}^n y_i \frac{\partial f_d}{\partial x_i}$$

We define the *center* of f as the vector y which minimizes the invariant norm of the homogeneous polynomial g_{d-1}

$$||f_{d-1} + y^t \nabla f_d||^2$$
,

a least squares problem, which has a unique solution if the vectors of coefficients of the partial derivatives of the term of degree *d*, the homogeneous polynomials

$$\frac{\partial f_d}{\partial x_1}, \dots, \frac{\partial f_d}{\partial x_n}$$

are linearly independent.

In matrix form, the term of degree d - 1 of the translated polynomial can be written as

$$g_{d-1}(x) = [F_{[d-1]} + F_{[d-1,1]} y]^{t} X_{[d-1]}(x) ,$$

in such a way that the center of f is the minimizer of the quadratic

$$\|F_{[d-1]} + F_{[d-1,1]} y\|^2 . (5.3)$$

The condition for unique solution is now a condition on the rank of the matrix $F_{[d-1,1]}$, which has to be equal to n. Equivalently, the covariant matrix

$$F_{[1,d-1]}F_{[d-1,1]} = F_{[d-1,1]}^t F_{[d-1,1]}$$
(5.4)

has to be nonsingular. But even when such a condition is not satisfied,

$$y = -F_{[d-1,1]}^{\dagger}F_{[d-1]}$$
(5.5)

is a solution, where $F_{[d-1,1]}^{\dagger}$ is the pseudoinverse of $F_{[d-1,1]}$. When the condition for unique solution is satisfied, the pseudoinverse is given by

$$F_{[d-1,1]}^{\dagger} = [F_{[1,d-1]}F_{[d-1,1]}]^{-1}F_{[d-1,1]}^{t}$$

The first step in the classification and positioning process, is to determine whether the matrix (5.4) is close to singular or not, by computing its n eigenvalues. These eigenvalues are Cartesian invariants of the form f_d of higher degree, and so, Euclidean invariants of the polynomial f. They are used as the first step in the classification process, and to limit the

search to a small subset of the database. Since the matrix (5.4) might not be well conditioned, the proper way to compute this eigenvalues is by appling the singular value decomposition algorithm [66] to the matrix $F_{[1,d-1]}$. The square of these singular values are the eigenvalues of (5.4).

Finally, the center of a planar curve or surface, as has been defined in this section, is independent of the coordinate system

Lemma 16 Let f(x) be a polynomial of degree d, x' = Ax + b an Euclidean coordinate transformation, $f'(x') = f(A^t(x'-b))$, $y = -F^{\dagger}_{[d-1,1]}F_{[d-1]}$ and $y' = -F'^{\dagger}_{[d-1,1]}F'_{[d-1]}$. Then y' = Ay + b.

5.1.2 Singular case

If $F_{[d-1,1]}$ has rank k < n, there exists an orthogonal transformation x' = Ax such that, if $f'(x') = f(A^tx)$ is the unique polynomial which satisfies the then the last n - k columns of $F'_{[d-1,1]}$ are identically zero. Since the columns of $F'_{[d-1,1]}$ are the coefficients of the partial derivatives of f', this means that the form $\phi'_d(x')$ is only function of the first k variables x'_1, \ldots, x'_k . The matrix A can be computed diagonalizing $[F_{[1,d-1]}F_{[d-1,1]}]$ because

$$F'_{[1,d-1]}F'_{[d-1,1]} = A [F_{[1,d-1]}F_{[d-1,1]}] A^t.$$

Only the first k variables of y' are present in (5.3), and are uniquely determined by (5.5). After fixing these k variables, we look at the form of highest degree j < d which is linear in the remaining variables y'_{k+1}, \ldots, y'_n , which do not appear in the highest degree form, and we can minimize its invariant norm with respect to these variables in the same way as we did before. If the problem is not totally solved yet, we can repeat this procedure with the forms of lower degree.

In general, the center will be uniquely determined if the partial derivatives

$$\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}$$

are linearly independent, or equivalently, if the block matrix of coefficients of the partial derivatives

$$(F_{[1,d-1]} | F_{[1,d-2]} | \cdots | F_{[1,1]} | F_{[1,0]})$$
(5.6)

has maximal rank n. If the rank of this matrix is k < n, the previous procedure will uniquely determine k components of the center. The others can take any value, because in such a case, the curve or surface defined by f' would be independent of them. We will determine the numerical rank of the matrix (5.6) as the first step, using the QR algorithm, or the singular value decomposition algorithm, and reduce the number of variables of the problem if so is required.

In the singular case, some orientation information is obtained. Furthermore, the ranks of the matrices of partial derivatives are Euclidean invariants, and can be used for recognition as well. The center of a planar curve or surface, as defined by this procedure, is covariant with the curve or surface, as in the nonsingular case.

5.2 The center of a 3D curve

An algebraic curve has been defined as the set of zeros of a vector $\mathbf{f}(x) = (f(x), g(x))^t$ of polynomials of degree $\leq d$, with at least one of the two components of degree d. We can decompose the polynomials as sums of forms

$$f = \sum_{i=0}^{d} f_i$$
 and $g = \sum_{i=0}^{d} g_i$

and without loss of generality, we will assume that the the degree of f is d, and the two forms of higher degree, f_d and g_d , are orthogonal with respect to the invariant inner product of forms of degree d. Otherwise, by independence of representation, we can replace the two polynomials by two linear combinations of them which satisfy this condition. These transformations do not change the curve or surface, and so, these constraints do not restrict the number of representable curves. If the degree of g is less than d, we define the center of the curve as the center of the surface associated with f.

If both f and g are polynomials of the same degree d, we can also assume that the invariant norms of f_d and g_d are both equal to 1. In this case we define the center as the point y which minimizes the sum of the square norms of the two forms of degree d - 1 of the translated polynomials, the quadratic

$$||F_{[d-1]} + F_{[d-1,1]}y||^2 + ||G_{[d-1]} + G_{[d-1,1]}y||^2$$

which can also be written as

$$||| H_{[d-1]} + H_{[d-1,1]} y||^2$$

where, for each pair of nonnegative integers j and k, the matrix $H_{[j,k]}$ is constructed by concatenating the corresponding matrices of coefficients of f and g

$$H_{[j,k]} = \begin{pmatrix} F_{[j,k]} \\ G_{[j,k]} \end{pmatrix} ,$$

and we also write $H_{[k]}$ instead of $H_{[k,0]}$. Finally, the solution is given by

$$y = -H_{[d-1,1]}^{\dagger}H_{[d-1]}$$
.

when $[H_{[1,d-1]}H_{[d-1,1]}]$ is nonsingular, and can be extended to the singular case as we have done for the planar curves and surfaces.

It can be proved that the center of a curve defined in this way is independent of the coordinate system, and also independent of the representation.

5.3 Orientation of a curve or surface

The *canonical orientation* of an algebraic planar curve or surface, defined by only one polynomial, can be defined in several ways, all of them based on the fact that a symmetric $n \times n$

matrix with nonrepeated eigenvalues $\lambda_1 > \cdots > \lambda_n$, has an associated set of unit length eigenvectors v_1, \ldots, v_n , thus generating 2^n different orthogonal coordinate systems having unit vectors in the directions of these eigenvectors, i.e., the coordinate systems defined by the orthogonal matrices

$$(\pm v_1 \mid \cdots \mid \pm v_n)$$
 .

In the example of the quadratic polynomial of two variables, the eigenvectors of the matrix

$$F_{[1,1]} = \begin{pmatrix} F_{(2,0)} & F_{(1,1)} \\ F_{(1,1)} & F_{(0,2)} \end{pmatrix}$$
(5.7)

define the orientation of the polynomial, if its eigenvalues are not repeated.

For a polynomial $f = \sum_{k=0}^{d} f_k$ of degree d, we consider the symmetric $n \times n$ matrix whose (i, j)-th element is the invariant inner product of the *i*-th and *j*-th partial derivatives of f_d with respect to x_i and x_j

$$\left\langle \frac{\partial f_d}{\partial x_i}, \frac{\partial f_d}{\partial x_j} \right\rangle = F_{[d-1,1]}^t F_{[d-1,1]}$$
(5.8)

In the case of the quadratic polynomial of two variables, it is the square of (5.7), which has the same eigenvectors, if its eigenvalues are not repeated.

If the matrix (5.8) has all different eigenvalues, we define the *canonical orientation* of f as the orientation induced by the eigenvectors of (5.8). Then, in order to disambiguate among the 2^n different frames of reference, we can find the location of certain covariant points, other than the center. Every nonzero component of a covariant point can be used to chose the orientation of the corresponding axis. For example, if the polynomial $f = \sum_{k=0}^{d} f_k$ is already centered, the following are covariant points

$$\begin{cases} F_{[1,k]}F_{[k]} & 1 \le k \le d-1\\ F_{[1,j]}F_{[j,k]}F_{[k]} & 1 \le j, k \le d-1, j+k \le d \end{cases}.$$

If the matrix has repeated eigenvalues, we will have to use information provided by the other homogeneous terms of f to construct another $n \times n$ covariant matrix. In general, we can consider the eigenvectors of the $n \times n$ matrix whose (i, j)-th component is given by

$$\sum_{k=1}^{d} w_k \left\langle \frac{\partial f_k}{\partial x_i}, \frac{\partial f_k}{\partial x_j} \right\rangle = \sum_{k=1}^{d} w_k \left(F_{[1,k-1]} F_{[k-1,1]} \right)$$
(5.9)

where w_1, \ldots, w_d are fixed constants, chosen to minimize the likelihood of repeated eigenvalues among the family of expected curves or surfaces, and the polynomial $f = \sum_{k=0}^{d} f_k$ has been previously centered. For every value of w_1, \ldots, w_d , the *n* eigenvalues of (5.9) are Euclidean invariants of the polynomial *f*, and although they are not sufficient to differentiate between any two polynomials of the same degree, they can also be used as the first step towards the classification of *f*.

The *canonical orientation* of an algebraic 3D curve can be defined in the same way, but using the matrices $H_{[j,k]}$ defined in section 5.2, instead of the matrices $F_{[j,k]}$.

5.4 Geometric matching of groups of curves or surfaces

Given two groups of polynomials, $\{f_1, \ldots, f_r\}$ and $\{g_1, \ldots, g_r\}$, representing planar curves or surfaces, an apparently more difficult problem is to find the Euclidean transformation which best aligns the first group with respect to the second one. We will assume that for each degree d, there are exactly as many polynomials of degree d in the first group as in the second group, as for example when all the polynomials have a common degree, but the correspondences are a priori unknown. The simplest case, when all the polynomials are linear, is related to the recognition and matching of objects which can be well approximated by polyhedrals. The methods described in this chapter already provide the solution of this problem. We consider the two polynomials

$$\begin{cases} f = f_1 \cdots f_r \\ g = g_1 \cdots g_r \end{cases},$$

the products of all the polynomials in the first and second group, respectivelly, we compute the corresponding intrinsic frames of reference, and obtain the Euclidean transformation which produces the best global alignment from them. In fact, not all the homogeneous terms of the products have to be computed. In general it will be suficient to compute the two terms of higher degree, because only them are used to compute the intrinsic frame of reference in the nonsingular case.

Note that, since the multiplication of polynomials is commutative and asociative, the coefficients of f and g are symmetric functions of the coefficients of the factors. These symmetric functions are independent of the order of the factors within the group, but still contain enough information to recover the transformation which best aligns the two groups. With this formulation the solution of the correspondence problem is avoided, but the correspondences can be recovered *a posteriori*, it they are needed.

Related work on matching of groups of curves or surfaces

The idea of using a vector of symmetric functions with enough information to determine the best matching transformation has been proposed by Brockett [26] in the context of point set matching problems. He shows how to minimize, among other related partially continuous and partially combinatorial optimization problems, the sum of distances

$$\sum_{i=1}^{r} \|Qx_{\pi(i)} - y_i\|^2 , \qquad (5.10)$$

where Q is an orthogonal matrix, π is a permutation of r elements, and $x_1, \ldots, x_r, y_1, \ldots, y_r$ are *n*-dimensional points. Although for every given permutation π the problem has a closed form solution

$$Q_{\pi} = \left(\Sigma_{\pi}^{t}\Sigma_{\pi}\right)^{-1/2}\Sigma_{\pi}^{t} \quad \text{where} \quad \Sigma_{\pi} = \sum_{i=1}^{r} x_{\pi(i)}y_{i}^{t}, \quad (5.11)$$

and $(\Sigma_{\pi}^{t}\Sigma_{\pi})^{-1/2}$ is the inverse of the unique symmetric positive definite square root of $\Sigma_{\pi}^{t}\Sigma_{\pi}$, it would be necessary to check the r! candidates Q_{π} to solve the original problem. Brockett proposes the use of a higher dimensional vector of symmetric functions of the points, and the correspondent linear representation of the orthogonal matrix Q in that vector space to turn the problem into a purely continuous one, which can be solved using standard smooth optimization techniques.

Related to both, the problem of simultaneously matching groups of implicit curves, and these point set matching problems, is the methods introduced by Faugeras and Hebert [55, 56] for matching groups of planar or quadratic patches. In the first case all the polynomials are linear, and they write

$$\begin{cases} f_i(x) &= v_i^t x - \delta_i \\ g_i(x) &= v_i^{\prime t} x - \delta_i^{\prime} \end{cases} \quad i = 1, 2, \dots, r ,$$

where $v_1, \ldots, v_r, v'_1, \ldots, v'_r$ are unit length vectors, normal to the planar patches that they represent, and $\delta_1, \ldots, \delta_r, \delta'_1, \ldots, \delta'_r$ are the signed distances from the origin of the coordinate system to the corresponding planes. In this case they assume that the correspondences, determined by the permutation π , are a priori known, and the function of the Euclidean transformation T(x) = Qx + b that they minimize is

$$\sum_{i=1}^{r} \|Qv_{\pi(i)} - v'_i\|^2 + W \sum_{i=1}^{r} \left(v_{\pi(i)}^t(Q^t b) + \delta_{\pi(i)} - \delta'_i\right)^2 , \qquad (5.12)$$

where W is a positive constant. Note that, if we denote $b' = Q^t b$, the left sum is only function of Q, and the right sum is only function of b'. Minimizing (5.12) is equivalent to minimizing the two sums independently of each other. The first sum is equivalent to (5.10), and it has a closed form solution given by (5.11), with $x_{\pi(i)}$ replaced by $v_{\pi(i)}$, and y_i by v'_i . Faugeras and Hebert use quaternions to turn this minimization problem into an eigenvalue problem instead of the direct solution described above. With respect to the second sum, since there is no constraint on b' in the quadratic function

$$\sum_{i=1}^{r} \left(v_i^t b' + \delta_i - \delta_i' \right)^2 \, ,$$

minimizing this sum is a standard least squares problem, which also has a direct solution

$$b_{\pi} = \Delta_{\pi} V_{\pi}^t \left(V_{\pi} V_{\pi}^t \right)^{-1}$$

where V_{π} is the $n \times r$ matrix

$$V_{\pi} = [v_{\pi(1)} | \cdots | v_{\pi(r)}],$$

and Δ_{π} is the row vector

$$\Delta_{\pi} = \left[\delta_1' - \delta_{\pi(1)}, \dots, \delta_r' - \delta_{\pi(r)} \right].$$

Faugeras and Hebert also propose a similar method for matching groups of quadratic patches. The new performance function turns out to be a particular case of the family of problems considered by Brockett, whose solution involves nonlinar optimization techniques.

There are clear advantages in the methods introduced in this chapter, with respect to Faugeras and Hebert's. In the first place, our method is more general and applies to any family of planar algebraic curves or algebraic surfaces. In the second place it produces a direct solution with polynomial complexity in the dimension of the space and the degrees of the polynomials, while the other method have complexity exponential in the number of curves, because the correspondence is unknown, it produces a direct solution only if all the curves are stright lines or all the surfaces are planes, and requires nonlinear optimization in the quadric case.

The solution (5.11) of the point matching problem corresponding to the minimization of (5.10) with a priori knowledge of the correspondence associated with the permutation π , has been derived and used by several authors in Computer Vision. It seems to be that Nadás [102] was the first to give the closed form solution. Arun, Huang and Blostein [7] propose an alternative derivation based on the singular value decomposition. Schwartz and Sharir [119] obtain the same result for continuously parameterized curves, and Wolfson [145] uses it for curve matching.

5.5 A Remark on algebraic curve and surface fitting

Since the homogeneous term f_d of highest degree of a polynomial $f = \sum_{k=0}^{d} f_k$ of degree d is invariant under translations, the invariant norm of f_d is an Euclidean invariant of the polynomial f. This invariant can be used as a constraint for fitting an algebraic surface or 2D to a data set $\mathcal{D} = \{p_1, \ldots, p_q\}$, by minimizing the mean square error

$$\frac{1}{q} \sum_{i=1}^{q} |f(p_i)|^2 ,$$

constrained by

$$||f_d||^2 = 1$$
.

Since this constraint is invariant under Euclidean transformations, the curve or surface defined by the minimizer of the problem is independent of the coordinate system. Bookstein [20] introduced the constraint

$$||f_2||^2 = \frac{1}{2}F_{(2,0)}^2 + F_{(1,1)}^2 + \frac{1}{2}F_{(0,2)}^2$$

for fitting conics to planar data sets following this method, and Cernuschi [31] derived the constraint

$$||f_2||^2 = \frac{1}{2}F_{(2,0,0)}^2 + F_{(1,1,0)}^2 + F_{(1,0,1)}^2 + \frac{1}{2}F_{(0,2,0)}^2 + F_{(0,1,1)}^2 + \frac{1}{2}F_{(0,0,2)}^2.$$

for fitting quadric surfaces to three dimensional data sets. The problem with this approach to algebraic curve and surface fitting is that, in general, the mean square error is a very biased approximation of the mean square distance from the data points to the set of zeros of f. The curve or surface defined by the solution of this minimization problem, although invariant, fails to represent the data near singular points. The methods described in Chapter 2 produce better results.

5.6 Proofs

Proof of Lemma 16 :

It is suficient to consider the cases of pure translation and pure rotation separately. Let us first consider the case of a pure translation, i.e., x' = x + b. Since the term of degree d of a polynomial is independent of the transformation parameters, we have

$$f'_d(x') = f_d(x) ,$$

or equivalently,

$$F'_{[d-1,1]} = F_{[d-1,1]}$$
.

The term of degree d - 1 is given by

$$f'_{d-1}(x') = f_{d-1}(x) - D_{[1]}f_{[d-1]}(x)b$$

or, in terms of the coefficients

$$F'_{[d-1]} = F_{[d-1]} - F_{[d-1,1]}b$$
.

The center is in this case

$$y' = -F'_{[d-1,1]}{}^{\dagger}F'_{[d-1]} = -F^{\dagger}_{[d-1,1]}F_{[d-1]} + F^{\dagger}_{[d-1,1]}F_{[d-1,1]}b = y+b.$$

Now, the case of pure rotation x' = Ax. In this case, since the terms of different degrees transform independently of each other, we can apply the transformation rules studied in the previous chapter

$$F'_{[d-1,1]} = A_{[d-1]}F_{[d-1,1]}A^t$$
 and $F'_{[d-1]} = A_{[d-1]}F_{[d-1]}$,

and obtain

$$F'_{[d-1,1]}^{\dagger} = AF^{\dagger}_{[d-1,1]}A^t_{[d-1]}.$$

Finally,

$$y' = -F'_{[d-1,1]}{}^{\dagger}F'_{[d-1]} = -AF^{\dagger}_{[d-1,1]}A^{t}_{[d-1]}A_{[d-1]}F_{[d-1]} = -AF^{\dagger}_{[d-1,1]}F_{[d-1]} = Ay.$$

Chapter 6

Moment invariants

The boundaries of many three dimensional objects cannot be well approximated by piecewise algebraic surfaces. In this chapter we will develop an alternative approch for the recognition and positioning of irregular objects, partially based on some of the techniques described in Chapter 4. Due to the unknown ammount of occlusion present in the data, these method will be based, again, on comparing, matching, and orienting small regions of the date set with respect to regions of known models stored in a data base.

Recognition will be based on computing invariants of vectors and matrices of moments of regions of the data sets, not invariants of the fitting curves or surfaces, but of the sets of points themselves, and comparing these invariants against the corresponding invariants previously computed for similar regions of the models in the database.

The methods to recover the unknown transformation which relates two matching sets of points, one in the data, and the other in a model, will be based, as in Chapter 5, on an intrinsic, or covariant, frame of reference of a finite sets of points. The center of a set of points will not be defined as for curves and surfaces, but the canonical orientation will be defined in a similar way, by diagonalizing square covariant matrices.

We are primarily interested in Euclidean transformations, because they are related to the recognition and positioning of three-dimensional objects from range data, but we will also study affine invaraints, and will define a canonical frame of reference with respect to affine transformations.

Besides, we can not consider general projective transformations because they transform bounded sets in unbounded sets, and so, the moments are not well defined with respect to all the projective coordinate systems. Also, the transformation rules of moments with respect to projective transformations are no longer linear. The main application of this affine normalization of shapes is in the recognition and positioning of objects from the projections of some of their contours onto the two-dimensional image plane. This is so because when the camera is far away from the scene, the projective transformation which corresponds to the imaging operation can be approximated by an affine transformation. Since there exists an infinite number of nonsingular linear transformation which transform a given square covariant matrix into the identity matrix, the affine normalization is based on transforming one covariant matrix into the identity matrix, and simultaneouly diagonalizing a second square covariant matrix.

In section 6.1 we show how to properly define moments for continuous and discrete data sets. In section 6.2 we define the center of a shape and the centered moments. In section 6.3 we apply the techniques of Chapter 4 for the computation of affine and Euclidean moment invariants. In section 6.4 we show how to define both Euclidean and affine intrinsic coordinate systems of shapes. In section 6.6 we introduce the *shape polynomials* and the Δ -distances to measure how well a shape fits a subset of another shape, obtaining new tools for testing hypotheses supported by several regions of the data set. Finally, in section 6.7 we review the literature related to these subjects.

6.1 Moments of discrete and continuous data sets

In the first place, we will consider dense data sets, such as those provided by laser range sensors. In this case it is better to assume that the data is a sampled version of a *n*-dimensional nonnegative integrable density function $\mu(x)$, and base the analysis on the continuous case. We will only consider density functions which are bounded, nonnegative, and have compact support. In this way the integral

$$\int \phi \, d\mu = \int \phi(x)\mu(x) \, dx \tag{6.1}$$

will be finite for every polynomial $\phi(x)$. Furthermore, we will also require the *total mass* of μ

$$|\mu| = \int d\mu = \int \mu(x) \, dx$$

to be positive, otherwise the integral (6.1) will be zero for every polynomial ϕ .

If x' = Ax + b is an affine transformation, $\phi'(x') = \phi(A^{-1}(x' - b)) = \phi(x)$, and $\mu'(x') = \mu(A^{-1}(x' - b)) = \mu(x)$, by the well known change of variables formula, we have

$$\int \phi'(x')\mu'(x')\,dx' = \int \phi(x)\mu(x)\,|A|\,dx\,,$$

because the determinant of the matrix *A* is the Jacobian of the affine transformation. The prescence of this multiplicative factor is a problem which would affect the transformation rules of moments, and the way to solve it is to consider the *normalized* integral

$$\frac{1}{|\mu'|} \int \phi'(x')\mu'(x')\,dx' = \frac{1}{|A||\mu|} \int \phi(x)\mu(x)\,|A|\,dx = \frac{1}{|\mu|} \int \phi(x)\mu(x)\,dx\,. \tag{6.2}$$

With this normalization, and if we define, consistently with the notation of Chapter 4, the vectors and matrices of moments as

$$\mu_{[d]} = \frac{1}{|\mu|} \int X_{[d]}(x) \, d\mu(x) \quad \text{and} \quad \mu_{[k,j]} = \frac{1}{|\mu|} \int X_{[k,j]}(x) \, d\mu(x) ,$$

with d, k and j nonnegative integers, these vectors and matrices of moments follow the well known transformation rules of the corresponding vectors and matrices of monomials, i.e., if x' = Ax is a nonsingular homogeneous coordinate transformation, then

$$\begin{aligned} \mu'_{[k,j]} &= \frac{1}{|\mu'|} \int X_{[k,j]}(x') \, d\mu'(x') \\ &= \frac{1}{|\mu|} \int \left(A_{[k]} X_{[k]} X_{[j]}^t(x) A_{[j]}^t \right) d\mu(x) &= A_{[k]} \, \mu_{[k,j]} \, A_{[j]}^t \end{aligned}$$

Note that, for every multiindex α of degree d, the α -th element of the vector of moments $\mu_{[d]}$ is

$$\mu^{\alpha} = \frac{1}{|\mu|} \int x^{\alpha} d\mu(x) ,$$

multiplied by the constant $\sqrt{\frac{1}{\alpha!}}$.

We will also consider sparse data sets, composed of easily distinguishable feature points, such as sharp corners, and points of high curvature, which can also be recovered using stero techniques, or even data provided by tactile sensors. In this case, the data will be represented by a finite set $\mu = \{p_1, \ldots, p_q\}$ of *n*-dimensional points, and the normalized integral (6.2) will be replaced by the normalized sum

$$\frac{1}{q}\sum_{i=1}^{q}\phi(p_i) \ .$$

The vectors and matrices of moments are defined accordingly

$$\mu_{[d]} = \frac{1}{q} \sum_{i=1}^{q} X_{[d]}(p_i) , \qquad \mu_{[k,j]} = \frac{1}{q} \sum_{i=1}^{q} X_{[k,j]}(p_i) .$$

The jacobian of the transformation does not appear in the discrete change of variables formula, but the vectors and matrices of moments still transform as the corresponding vectors and matrices of moments, under homogeneous coordinate transformations x' = Ax.

$$\mu'_{[k,j]} = \frac{1}{q} \sum_{i=1}^{q} X_{[k,j]}(Ap_i)$$

= $\frac{1}{q} \sum_{i=1}^{q} \left(A_{[k]} X_{[k,j]}(p_i) A_{[j]}^t \right)$
= $A_{[k]} \left(\frac{1}{q} \sum_{i=1}^{q} X_{[k,j]}(p_i) \right) A_{[j]}^t = A_{[k]} \mu_{[k,j]} A_{[j]}^t.$

Both cases, the positive density and the finite set of points, can be handled with the same formalism, because both are particular cases of finite positive measures of compact support, which we have called *shapes* [131]. The discrete case corresponds to a singular measure equal to a finite sum of Dirac deltas, and the continuous case to an absolutely continuous measure with respect to the Lebesgue measure. Thus, from now on we will continue with the analysis in the continuous case. The discrete case can be obtained by replacing the normalized integrals by normalized sums.

6.2 Centered moments

As in the case of the coefficients of polynomials analyzed in the previous chapter, the vectors and matrices of moments of different degrees transform independently of each other when the translation part b of the affine transformation x' = Ax + b is zero. Moments of different degrees are combined when a translation part is present. However, the transformation rules for moments under translation differ from the corresponding rules for coefficients of polynomials. We have seen that the coefficients degree d - 1 of a polynomial of degree dtransform linearly under translations, while in this case, the moments of first degree transform linearly. This property makes the definition of the center of a shape much easier. It is natural to define the *center* of a shape μ as the the vector of first degree moments

$$\mu_{[1]} = \frac{1}{|\mu|} \int x \, d\mu(x) \; .$$

The center of a shape is covariant under affine transformations, because if x' = Ax + b is an affine transformation

$$\mu'_{[1]} = \frac{1}{|\mu'|} \int x' \, d\mu'(x') = \frac{1}{|\mu|} \int (Ax+b) \, d\mu(x) = A \, \mu_{[1]} + b \, ,$$

and then, only consider the *centered* moments, moments computed with respect to the center

$$M^{\alpha} = \frac{1}{|\mu|} \int (x - \mu_{[1]})^{\alpha} d\mu(x) ,$$

and the corresponding vectors and matrices of centered moments

$$M_{[k,j]} = \frac{1}{|\mu|} \int X_{[k,j]}(x-\mu_{[1]}) \, d\mu(x) \; .$$

Note that all of these vector and matrices are clearly invariant under translation of the data μ .

If we assume that the origin of the coordinate system is located at the center of the data, then we only have to consider the transformation rules of the vectors and matrices of centered moments under homogeneous linear transformations x' = Ax in the affine case, or under orthogonal transformations x' = Qx, in the Euclidean case.

The matrices of moments are covariant matrices, both in the discrete and in the continuous cases, and we can apply the same methods for the computation of invariants and covariants developed in Chapter 4. The invariants of these matrices, and the joint invariants of several of these matrices can be used for the classification of the data set μ .

It is important to note that the square matrices of moments are nonnegative definite. If $\Phi_{[d]}$ is an arbitrary vector of h_d coefficients, or equivalently, if $\phi(x) = \Phi_{[d]}^t X_{[d]}(x)$ is aform of degree d, then

$$\Phi_{[d]}^{t} M_{[d,d]} \Phi_{[d]} = \frac{1}{|\mu|} \int \Phi_{[d]}^{t} X_{[d,d]}(x) \Phi_{[d]} d\mu(x)
= \frac{1}{|\mu|} \int (\Phi_{[d]}^{t} X_{[d]}(x))^{2} d\mu(x) = \frac{1}{|\mu|} \int \phi(x)^{2} d\mu(x) \ge 0,$$
(6.3)

because the integrand is nonnegative for every value of x. In general, these matrices are positive definite, unless all the points are interpolated by the algebraic curve or surface defined by the zeros of a form $\phi(x) = \Phi_{[d]}^t X_{[d]}(x)$. In particular, if the density function $\mu(x)$ is positive on an open set, all the square matrices $M_{[d,d]}$ are positive definite. However, some of them might be very badly conditioned.

The matrix $M_{[1,1]}$ is also known as the *scatter matrix* of the data μ . We will assume that it is positive definite, otherwise all the data is contained in hyperplane of the *n*-dimensional space, and this is very unlikely.

6.3 Cartesian and affine moment invariants

Since the vectors and matrices of centered moments $M_{[k,j]}$ are covariant, the problem of computing moment invariants reduces to the computation of joint invariants of a family of covariant matrices, studied in detail in Chapter 4.

6.3.1 Cartesian moment invariants

In this case, the vectors and matrices of centered moments follow the same transformation rules, under orthogonal transformations, than the corresponding covariant matrices of coefficients of forms. Defining the *moment forms*

$$\phi(x, y) = X_{[k]}^t(x) M_{[k,j]} X_{[j]}(y) .$$

the problem of computing moment invariants reduces to the computation of joint invariants of a family of forms.

Among the techniques that we have described for the computation of Cartesian invariants of forms, the eigenvalues of square covariant matrices are the best suited for numerical computation, followed by the harmonic decomposition. The h_k eigenvalues of the matrix of moments $M_{[k,k]}$ are Cartesian invariants.

6.3.2 Affine moment invariants

In this case, in order to use the eigenvalues of square matrices as absolute invariants, we need to construct square matrices which are covariant on one side, and contravariant on the other side. The matrices $M_{[j,k]}$ are covariant on both sides, but since the square matrices $M_{[k,k]}$ are usually positive definite, we can define a new family of matrices with the desired properties. For every pair of nonnegative integers j and k we will write

$$H_{[j,k]} = M_{[j,k]} M_{[k,k]}^{-1} . (6.4)$$

Note that this matrices only make sense for $j \neq k$, because $H_{[k,k]}$ is the identity matrix. Also note that $H_{[j,k]} \neq H_{[k,j]}^t$. Now, if $M_{[j,j]}$ and $M_{[k,k]}$ are positive definite, the square $h_k \times h_k$ matrix

$$H_{[k,j]}H_{[j,k]} = M_{[k,j]}M_{[j,j]}^{-1}M_{[j,k]}M_{[k,k]}^{-1}$$

is left covariant and right contravariant, and so, its h_k principal values $\lambda_1 > ... > \lambda_{h_k} \ge 0$ are joint absolute invariants of the moments of degrees 2j, j + k, 2k under affine transformations. This only makes sense for $k \le j$, because the other combination yields the same principal values, followed by zeros. For example, if k = 1 and j = 2 we obtain the simplest absolute affine moment invariants of a shape, the $h_1 = n$ principal values of the $n \times n$ matrix

$$H_{[1,2]}H_{[2,1]} = M_{[1,2]}M_{[2,2]}^{-1}M_{[2,1]}M_{[1,1]}^{-1}$$

which is a rational function of the centered moments of degree 2,3 and 4.

Another family of left covariant and right contravariant matrices can be constructed replacing $M_{[k,k]}$ by $M_{[1,1][k]}$ in (6.4), where $M_{[1,1][k]}$ *k*-th. degree representation of $M_{[1,1]}$. It is symmetric and positive definite when $M_{[1,1]}$ is positive definite. In this case we write

$$U_{[j,k]} = M_{[j,k]} M_{[1,1][k]}^{-1}$$
.

When j = k, the square matrix $U_{[k,k]}$ is not the identity matrix, and its h_k principal values are absolute invariants of the data μ . Since $M_{[1,1]}$ is positive definite, it has a nonsingular square root, a square matrix L such that $LM_{[1,1]}L^t = I$. We can take L as the inverse of the lower triangular Cholesky decomposition of $M_{[1,1]}$. From the properties of the representation map described in Lemma 3, we have

$$M_{[1,1][k]} = L_{[k]}^{-1} L_{[k]}^{-t}.$$

The matrices $U_{[k,k]}$ and $L_{[k]}U_{[k,k]}L_{[k]}^{-1}$ are conjugates, and so, they have the same characteristic polynomials. However, the last matrix is symmetric

$$L_{[k]}U_{[k,k]}L_{[k]}^{-1} = L_{[k]}M_{[k,k]}L_{[k]}^{t},$$

and has all real eigenvalues, which are absolute invariants of the shape μ . The simplest of these matrices corresponds to the case k = 2, which produces $h_2 = \binom{n+2-1}{n-1} = n(n+1)/2$ absolute invariants, functions of the centered moments of degree 2 and 4. The simplest affine absolute invariants of a shape are the *n* eigenvalues of the $n \times n$ symmetric matrix

$$LU_{[1,2]}U_{[2,1]}L^{-1} = LM_{[1,2]}M_{[1,1][2]}^{-1}M_{[2,1]}L^{t}$$

which are functions of the centered moments of degree 2 and 3. Note that, if we consider the coordinate transformation x' = Lx, this last matrix is nothing but

$$LM_{[1,2]}M_{[1,1][2]}^{-1}M_{[2,1]}L^t , = \left(LM_{[1,2]}L_{[k]}^t\right)\left(L_{[k]}M_{[2,1]}L^t\right) = M_{[1,2]}'M_{[2,1]}',$$

This property is the basis for the definition of the intrinsic affine frame of reference of a shape, described in detail in the next section.

6.4 Canonical frame of reference

We have defined the center of a shape as the mean of the data, the vector of moments of degree one, which is an affine covariant vector of the shape. In order to define an intrinsic frame of reference, we still have to determine a canonical orthogonal matrix, in the cartesian case, and a canonical nonsingular matrix, in the affine case.

6.4.1 Cartesian case

In this case we can define the orientation of a set of points as we did for algebraic curves and surfaces, as one of the 2^n orthonormal sets which diagonalizes a symmetric $n \times n$ covariant matrix. The simplest matrix of this kind is the *scatter matrix* $M_{[1,1]}$, but we can also use any one of the following

$$\begin{cases}
M_{[1,k]}M_{[k,1]} \\
M_{[1,k]}M_{[k,k]}M_{[k,1]} \\
M_{[1,k]}M_{[k,k]}^{-1}M_{[k,1]} \\
M_{[1,k]}M_{[1,1][k]}M_{[k,1]} \\
M_{[1,k]}M_{[1,1][k]}^{-1}M_{[k,1]}
\end{cases}$$
(6.5)

or any linear combination of them

$$\theta_1 M_{[1,1]} + \sum_{k \ge 2} \theta_k \left(M_{[1,k]} M_{[k,1]} \right) + \cdots$$

In order to disambiguate among the 2^n candidate orthogonal frames of reference, as we did in the case of polynomials, we use covariant *n*-dimensional vectors. Every nonzero element of one of these covariant vectors can be used to choose the orientation of the corresponding coordinate axis. The simplest covariant vector is $M_{[1]}$, but since the moments are centered, it is identically zero. Other covariant vectors can be computed as in (6.5), as follows

$$\begin{cases} M_{[1,k]}M_{[k]} \\ M_{[1,k]}M_{[k,k]}M_{[k]} \\ M_{[1,k]}M_{[k,k]}^{-1}M_{[k]} \\ M_{[1,k]}M_{[1,1][k]}M_{[k]} \\ M_{[1,k]}M_{[1,1][k]}^{-1}M_{[k]} \end{cases}$$
(6.6)

The simplest Cartesian covariant vector of this family is a function of centered moments of degree 2 and 3

$$v_1 = M_{[1,2]} M_{[2]}$$
.

If this vector is not identically zero, the following vector is another covariant vector

$$v_2 = M_{[1,1]}v_1 = M_{[1,1]}M_{[1,2]}M_{[2]}$$
.

If v_1 is not an eigenvector of $M_{[1,1]}$, then v_1 and v_2 are linearly independent, and in threedimensional space, the vector product of them $v_3 = v_1 \times v_2$ defines a third nonzero covariant vector. With these three linearly independent vectors, the orthogonal transformation can be uniquely determined. In the two-dimensional case, only one nonzero covariant vector is necessary to determine the orientation of the shape.

6.4.2 Affine case

The determination of an intrinsic affine coordinate system differs from the cartesian case. In the first place, although there are only 2^n orthogonal matrices which diagonalize a symmetric $n \times n$ matrix, the number of nonsingular matrices which diagonalize the same symmetric

matrix is infinite. However, if two nonsingular matrices transform a positive definite matrix into the identity matrix, they are related by an orthogonal transformation.

Lemma 17 Let M be a symmetric positive definite $n \times n$ matrix. Then,

- 1. The inverse of the lower triangular Cholesky decomposition of M is the unique lower triangular matrix L, with positive diagonal elements, such that $LML^t = I$.
- 2. If A and B are two $n \times n$ matrices such that $AMA^t = BMB^t = I$, then AB^{-1} is an orthogonal matrix. In particular, for every $n \times n$ matrix A such that $AMA^t = I$, there exists a unique orthogonal matrix Q such that A = QL.

For the proof of 1 see Golub [66], and for 2 just note that

$$I = AMA^{t} = A (B^{-1}B^{-t}) A^{t} = (AB^{-1}) (AB^{-1})^{t} ,$$

or equivalently, $Q = AB^{-1}$ is orthogonal.

Now, let M be any $n \times n$ covariant matrix of moments, such as $M_{[1,1]}$, or one of the matrices in (6.5). Let L be the triangular matrix of the Lemma, and let us consider the coordinate transformation x' = Lx defined by this matrix. Then, the corresponding covariant matrix M' in the new coordinate system is the identity matrix, because $M' = LML^t = I$. In order to determine a canonical affine transformation, we still need to uniquely specify a canonical orthogonal matrix, because, for every orthogonal matrix Q, if A = QL, and x'' = Qx' = Ax then, we also have $M'' = AMA^t = QQ^t = I$. After the coordinate transformation defined by L, we are in the Cartesian case, but since M' = I has all the eigenvalues repeated, we cannot use this matrix to determine an orientation, and we have to consider *a second covariant matrix* N, with nonrepeated eigenvalues, for the determination of the rotation part of A. This orthogonal matrix is, as in the cartesian case, one of the 2^n orthogonal matrices which diagonalize $N' = LNL^t$ leaving the eigenvalues in decreasing order.

In the applications we will take $M = M_{[1,1]}$, and

$$N = M'_{[1,k]}M'_{[k,1]} = LM_{[1,k]}M_{[k]}^{-1}M_{[k,1]}L^t ,$$

for the smallest value of k = 2, 3, ... for which N has nonrepeated eigenvalues. In general, it will be sufficient to consider the case k = 2. Finally, in the two and three-dimensional cases, we will use the covariant vectors

$$v_1 = M'_{[1,2]}M'_{[2]}$$

and

$$v_2 = M'_{[1,1]}v_1$$

to disambiguate among the 2^n candidate orthogonal transformations.

Affine covariant implicit curve and surface fitting

The fitting techniques of Chapter 2 are covariant with respect to similarity transformations, basically because distance is a metric concept. With the techniques to normalize a shape with respect to affine transformations described above, we can extend the fitting techniques to make them covariant with respect to affine transformations. In the first place the shape is normalized making its scatter matrix equal to the identity matrix, i.e., the data set $\mathcal{D} = \{p_1, \ldots, p_q\}$ is transformed to a new set $\mathcal{D}' = \{p'_1, \ldots, p'_q\}$ according to the transformation $x' = L(x - \mu_{[1]})$, where $\mu_{[1]}$ is the mean of the set \mathcal{D} , and L is the inverse of the lower triangular Cholesky decomposition of the scatter matrix $M_{[1,1]}$ of \mathcal{D} . The transformed data set \mathcal{D}' has zero mean and scatter matrix equal to the identity matrix. We know that any other affine transformation with these properties differ from this one only by an orthogonal transformation. Since the fitting methods of Chapter 2 are covariant with respect to similarity transformations, it is not necesary to finish the normalization process. If g(x') is a polynomial which describes the fitting curve or surface according to one of the fitting methods of Chapter 2, then $f(x) = g(L(x - \mu_{[1]}))$ describes a curve or surface which is an affine covariant function of the data set.

Note that these modifications are based on changing the distance metric. The inner product $\langle x, y \rangle = x^t y$ in *n*-dimensional space, is replaced by

$$\langle x, y \rangle = x^t M_{[1,1]}^{-1} y = (Lx)^t (Ly)^t = x'^t y'.$$

In this way, the approximate *affine invariant* square distance, from a point x to the set of zeros of **f**, is now

$$\mathbf{f}(x)^t \left(L^{-t} \mathbf{D} \mathbf{f}(x)^t \mathbf{D} \mathbf{f}(x) L^{-1} \right)^{-1} \mathbf{f}(x) ,$$

and in the case of planar curves or surfaces, it reduces to

$$\frac{f(x)^2}{\mathrm{D}f(x)M_{[1,1]}\mathrm{D}f(x)^t} \ .$$

6.5 Examples and implementation

A few simple examples of affine invariants are shown in figures 6.1, 6.2, and 6.3. The five numbers at the botom of the figures are five affine moment invariants. The first two are the eigenvalues of the symmetric 2×2 matrix $M'_{[1,2]}M'_{[2,1]}$, multiplied by 1000, and the last three are the eigenvalues of the 3×3 matrix $M'_{[2,2]}$, multiplied by 1000 as well. These centered moments are computed not with respect to the original coordinate system, but with respect to the coordinate system defined by x' = Lx, where L is a 2×2 lower triangular matrix such that $LM_{[1,1]}L^t = I$, and $M_{[1,1]}$ is the 2×2 matrix of moments with respect to the original coordinate system.

In order to show how simple this computations are, we now describe how these five affine moment invariants are evaluated. Given a set of edge points $\{p_1, \ldots, p_q\}$, we first



Figure 6.1: Some affine moment invariants. Only the data inside the circle is used in the computation.



Figure 6.2: Some affine moment invariants. Only the data inside the circle is used in the computation.



Figure 6.3: Some affine moment invariants. Only the data inside the circle is used in the computation.

compute its center

$$\bar{x}_1 = \frac{1}{q} \sum_{i=1}^{q} p_{i1} \bar{x}_2 = \frac{1}{q} \sum_{i=1}^{q} p_{i2}$$

where p_{i1} and p_{i1} are the two coordinates of the point p_i . Then we compute the centered second degree moments

which we rearrange into a 2×2 matrix

$$M_{[1,1]} = \begin{pmatrix} M_{(2,0)} & M_{(1,1)} \\ M_{(1,1)} & M_{(0,2)} \end{pmatrix} .$$

The third step is to find the lower triangular matrix L such that $LM_{[1,1]}L^t = I$. We compute it in two steps; we first find the lower triangular matrix L such that $LL^t = M_{[1,1]}$, the Cholesky decomposition of $M_{[1,1]}$

$$L = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} \text{ whith } \begin{cases} L_{11} = \begin{pmatrix} M_{(2,0)} \end{pmatrix}^{1/2} \\ L_{12} = & 0 \\ L_{21} = & M_{(1,1)}/L_{11} \\ L_{22} = \begin{pmatrix} M_{(0,2)} - L_{21}^2 \end{pmatrix}^{1/2} \end{cases}$$

and then we invert it in place

$$\begin{cases} L_{11} = 1/L_{11} \\ L_{21} = -L_{21}L_{11}/L_{22} \\ L_{22} = 1/L_{22} \end{cases}$$

At this point we compute moments of degree three, four, and eventually higher degree

$$M'_{(i,j)} = \frac{1}{q} \sum_{i=1}^{q} [L_{11}(p_{i1} - \bar{x}_1)]^i [L_{21}(p_{i1} - \bar{x}_1) + L_{22}(p_{i2} - \bar{x}_2)]^j$$

for i + j > 2. The first two affine invariants are the two eigenvalues of the 2×2 symmetric positive definite matrix $M'_{[1,2]}M'_{[2,1]} = M'_{[1,2]}M'^t_{[1,2]}$, where

$$M'_{[1,2]} = \begin{pmatrix} \frac{1}{\sqrt{2}}M'_{(3,0)} & M'_{(2,1)} & \frac{1}{\sqrt{2}}M'_{(1,2)} \\ \frac{1}{\sqrt{2}}M'_{(2,1)} & M'_{(1,2)} & \frac{1}{\sqrt{2}}M'_{(0,3)} \end{pmatrix} .$$

The last three affine moment invariants of the figures are the eigenvalues of the 3×3 , symmetric nonnegative definite matrix

$$M'_{[2,2]} = \begin{pmatrix} \frac{1}{2}M'_{(4,0)} & \frac{1}{\sqrt{2}}M'_{(3,1)} & \frac{1}{2}M'_{(2,2)} \\ \frac{1}{\sqrt{2}}M'_{(3,1)} & M'_{(2,2)} & \frac{1}{\sqrt{2}}M'_{(1,3)} \\ \frac{1}{2}M'_{(2,2)} & \frac{1}{\sqrt{2}}M'_{(1,3)} & \frac{1}{2}M'_{(0,4)} \end{pmatrix} .$$

The computation of Euclidean moment invariants is even simpler, because it does not require the computation of the matrix L. In the Euclidean case, the eigenvalues of $M_{[1,1]}$ are also moment invariants.

If we also want to compute an intrinsic affine coordinate system, we have to chose a value of k such that the eigenvalues of the 2×2 symmetric nonnegative definite matrix $M'_{[1,k]}M'_{[k,1]}$ are well separated. How to chose this value of k is currently under study, because early experiments using low values of k do not seem to provide sufficiently reliable orientations. This problem is related to the question of which affine moment invariants are more robust and stable than others. For example, when the shape of the data set is relatively simple, such as the example of figure 6.1, the five moment invariants described in this section seem to be stable. However, in more complex shapes, like those of figures 6.2 and 6.3, the first two third degree moment invariants do not seem to be very stable, but the three fourth degree moment invariants do seem stable. The matrices of moments introduced in this paper are closely related to the problem of fitting algebraic curves and surfaces to data [127, 128, 130], and the questions addressed in this paragraph are related to statistical tests of *goodness of fit* of the curves and surfaces. A more extensive experimentation and analysis of these and related problems is required, it is under way, and will be described in a future report.

Other problems to be addressed are related to how to implement the computation of the moments themselves. The strightforward evaluation of the sums is plagued with roundoff errors, in particular when the number of points in the region is large. The moments have to be evaluated hierarchically. The data set has to be partitioned in a number of of subsets, the moments evaluated for these subsets, and then added together. In the 2D case, it is also important to consider the individual pixels not as discrete points, but as small rectangles with area, and evaluate the moments on a rectangular area. Similarly, in the 3D case, surface-like data sets, have to be treated as sampled surfaces. The points have to be used to define a triangulation of the surfaces, and the moments evaluated as the sum of the surfaces integrals on the triangles. Explicit formulas for the moments on a 3D triangular surface can also be derived in this case.

6.6 Global testing and distances between shapes

With the methods described above in this chapter, the hypothesis generation part of a recognition and positioning system can be implemented. Small regions of the data set will be chosen. For each of them a vector of moment invariants will be computed. With these invariants the database of regions of objects, indexed by these invariants, will be searched. For every match found, a hypothesis of a certain object in certain position will be generated aligning the intrinsic coordinate systems of the two matching regions. Then, these hypotheses will have to be globally tested. Groups of these hypotheses will correspond to the same object in almost the same position and orientation. One way to solve these problems is to define an asymetric distance measure between two shapes. This distance measure has to measure how well one shape fits as a *subset* of a second shape. It has to be independent of the global coordinate system used, i.e., if the two shapes are transformed according to the same affine transformation, the distance measure has to stay invariant. With respect to improving a hypothesis supported by several regions of the data set, if one shape is transformed according to an affine transformations, while the other is kept unchanged, the distance measure also has to be a smooth function of the transformation parameters, in order to let us use numerical optimization techniques to iteratively improve a hypothesized position of the object.

Our solution to this problem is to define, for every shape, a family of smooth *potential functions*. These potential functions will be positive polynomials of increasing degree. We have called them *shape polynomials* [129, 131]. The coefficients of the shape polynomial of degree 2d are functions of the moments of degree $\leq 2d$, they yield low values close to the shape, and increase to infinity far away from the shapes. The approximation improves when the degree increases. Another shape will be a subset of the first shape, if the potential is low at every point of it, or equivalently, if the mean value of the shape polynomials on the new shape yield low values for a suficiently high degree.

6.6.1 Shape polynomials

For the definition of the shape polynomials, we need to use noncentered moments and homogeneous coordinates. To every *n*-dimensional point $x = (x_1, \ldots, x_n)^t$, we associate its homogeneous (n + 1)-dimensional version $\hat{x} = (1, x_1, \ldots, x_n)$. To an affine transformation x' = Ax + b it corresponds the homogeneous (projective) transformation

$$\begin{pmatrix} 1\\x' \end{pmatrix} = \begin{pmatrix} 1&0\\b&A \end{pmatrix} \begin{pmatrix} 1\\x \end{pmatrix} .$$
(6.7)

If the variable x_0 is kept equal to 1 in the vector of monomials of degree d in n+1 variables we obtain a vector of monomials of all the degrees 0, 1, ..., d in n variables. Let us denote

$$X_{(d)}(x) = X_{[d]}(\hat{x}) = \begin{pmatrix} \frac{1}{d!}X_{[0]}(x) \\ \vdots \\ \frac{1}{(d-j)!}X_{[j]}(x) \\ \vdots \\ \frac{1}{0!}X_{[d]}(x) , \end{pmatrix}$$

where $X_{[d]}$ on the left side is the vector of monomials of degree d in n + 1 variables, and the $X_{[j]}$ on the right side is the vector of monomials of degree j in n variables. The vector $X_{(d)}(x)$, except for the different scaling of its elements, is the vector X(x) used as a basis of the polynomials of degree $\leq d$ in n variables in Chapter 2. If μ is a shape, let us denote

$$\Sigma_{\mu,d} = \frac{1}{|\mu|} \int X_{(d)}(x) X_{(d)}^t(x) \, d\mu(x)$$

when the shape is an absolutely continuous positive measure with compact support, and

$$\Sigma_{\mu,d} = \frac{1}{q} \sum_{i=1}^{q} X_{(d)}(p_i) X_{(d)}^t(p_i)$$

when it is a finite set of data points. The matrix $\Sigma_{\mu,d}$ is symmetric, nonnegative definite and its elements are moments of degrees $0, 1, \ldots, 2d$. It is singular only if all the data is interpolated by an algebraic surface of degree $\leq d$, and we will assume that this is not the case. Except for the scaling of the elements of the vector of monomials, this is one of the two matrices involved in the generalized eigenvector fit method of Chapter 2.

The matrix $\Sigma_{\mu,d}$ is not covariant with respect to general projective transformations of the n + 1 homogeneous coordinates. However, it is covariant with respect to the subgroup of affine transformations. If x' = Ax + b is an affine transformation, and we write

$$B = \begin{pmatrix} 1 & 0 \\ b & A \end{pmatrix} ,$$

we have

$$X_{(d)}(x') = X_{[d]}(\hat{x}') = X_{[d]}(B\hat{x}) = B_{[d]}X_{[d]}(\hat{x}) , = B_{[d]}X_{(d)}(x)$$
(6.8)

where $B_{[d]}$ is the representation of degree d of the $(n + 1) \times (n + 1)$ matrix B. It easily follows that

$$\left(\Sigma_{\mu[d]}\right)' = \Sigma_{\mu'[d]} = B \Sigma_{\mu,d} B^t , \qquad (6.9)$$

i.e., that the matrix $\Sigma_{\mu,d}$ is covariant with respect to affine transformations.

We define the *shape polynomial* of degree 2d of the shape μ , as

$$\Upsilon_{\mu,d}(x) = \frac{1}{\binom{n+d}{n}} X_{(d)}(x)^t \Sigma_{\mu,d}^{-1} X_{(d)}(x) .$$

Shape polynomials of odd degrees are not defined, and we already know that the value of these polynomials are invariant under affine transformations, because, from the well known transformation rules of covariant and contravariant matrices, and from (6.8) and (6.9), we obtain

$$\Upsilon_{\mu',d}(x') = \Upsilon_{\mu,d}(x) \, .$$

Since the matrix $\Sigma_{\mu,d}^{-1}$ is positive definite, the shape polynomial $\Upsilon_{\mu,d}(x)$ yields a positive value vor every choice of x. Also, $\Upsilon_{\mu,d}(x) \to \infty$ when $||x|| \to \infty$, because, if $\lambda > 0$ is the minimum eigenvalue of $\Sigma_{\mu,d}^{-1}$, then

$$\Upsilon_{\mu,d}(x) \geq \frac{\lambda}{\binom{n+d}{n}} \|X_{(d)}(x)\|^2 = \frac{\lambda}{\binom{n+d}{n}} \frac{\|\hat{x}\|^2}{d!} = \frac{\lambda}{\binom{n+d}{n}} \frac{1+\|x\|^2}{d!} \to \infty$$

Figures 6.4, 6.5, and 6.6 show the graphs of shape polynomials of two dimensional shapes for different degrees. Darker regions correspond to low values, and light regions to higher values.

In the same way that we have done here, we can define *trigonometric* shape polynomials, replacing the vector $X_{(d)}(x)$ by a vector of complex exponentials. For a large family of shapes it can be proved that the sequence of trigonometric shape polynomials converges to the original shape [129]. The corresponding analysis for the shape polynomials defined in this section remains to be done, but the experimental evidence suggests a similar result.



Figure 6.4: (a) : Data points scattered around an ellipse. (b) : Graph of shape polynomial of degree 4. (c) : Graph of shape polynomial of degree 8. (d) : Graph of shape polynomial of degree 12.



Figure 6.5: (a) : Wrench. (b) : Graph of shape polynomial of degree 8. (c) : Graph of shape polynomial of degree 12. (d) : Graph of shape polynomial of degree 32.



Figure 6.6: (a) : Plier. (b) : Graph of shape polynomial of degree 4. (c) : Graph of shape polynomial of degree 12. (d) : Graph of shape polynomial of degree 24.

6.6.2 Comparing shapes with the △-distances

Let μ and ν be two shapes. We define the Δ -distance of degree d from ν to μ as

$$\Delta_{\mu,d}(\nu) = \frac{1}{|\nu|} \int \Upsilon_{\mu,d}(x) \, d\nu(x) \, d\nu(x)$$

The computation of these distances is inexpensive, and just involve matrix operations, because $A_{1}(x) = \frac{1}{2} \int V_{1}(x) dx = \frac{1}{2} \int V_{2}(x) dx = \frac{1}{2} \int V_{2}(x)$

$$\begin{aligned} \Delta_{\mu,d}(\nu) &= \frac{1}{\binom{n+d}{n}} \frac{1}{|\nu|} \int X_{(d)}(x)^{t} \Sigma_{\mu,d}^{-1} X_{(d)}(x) \, d\nu(x) \\ &= \frac{1}{\binom{n+d}{n}} \frac{1}{|\nu|} \int \operatorname{trace} \left(\Sigma_{\mu,d}^{-1} X_{(d)}(x) X_{(d)}(x)^{t} \right) \, d\nu(x) \\ &= \frac{1}{\binom{n+d}{n}} \operatorname{trace} \left(\Sigma_{\mu,d}^{-1} \frac{1}{|\nu|} \int X_{(d)}(x) X_{(d)}(x)^{t} \, d\nu(x) \right) \\ &= \frac{1}{\binom{n+d}{n}} \operatorname{trace} \left(\Sigma_{\mu,d}^{-1} \Sigma_{\nu,d} \right) \end{aligned}$$

The Δ -distances are joint affine covariants of the pair of shapes, i.e., they are independent of the affine coordinate system, because, under affine transformations, both matrices $\Sigma_{\mu,d}$ and $\Sigma_{\nu,d}$ transform covariantly, and so $\Sigma_{\mu,d}^{-1}\Sigma_{\nu,d}$ is contravariant on the left side, and covariant on the right side. Since the trace of this matrix is one of the coefficients of its characteristic polynomial, it is invariant under affine coordinate transformations.

Furthermore,

Lemma 18 If the shape ν represents a subset of the shape μ , then

$$\Delta_{\mu,d}(\nu) \leq \frac{|\mu|}{|\nu|}$$

for every degree d.

Note that the conclusion of this lemma is independent of the affine coordinate system, because both the Δ -distance, and the ratio $\frac{|\mu|}{|\nu|}$ are absolute affine invariants of the pair of shapes. We will use this lemma in the following way, if the inequality is satisfied for a sufficiently high value of d, we can hypothesize that ν is a subset of μ . Or equivalently, if the inequality is not satisfied, and $\Delta_{\mu,d}(\nu) \gg \frac{|\mu|}{|\nu|}$ for one value of d, we have to reject the hypothesis of ν being a subset of μ .

The Δ -distances can be used for positioning as well. If x' = Ax + b is either an affine transformation, a similarity transformation, or an Euclidean transformation, and B is the $(n+1) \times (n+1)$ matrix which represents the same transformation in homogeneous coordinates, we will consider the Δ -distance $\Delta_{\mu,d}(\nu')$ as a function of the transformation parameters

$$\xi(B) = \Delta_{\mu,d}(
u') = rac{1}{\binom{n+d}{n}} \operatorname{trace} \left(\Sigma_{\mu,d}^{-1} B_{[d]} \Sigma_{
u,d} B_{[d]}^t
ight) \, .$$

The global minimizer, B, of this functional is an estimate of the transformation which makes ν best fit as a subset of μ . Based on the last lemma, we can develop test to decide whether to accept the result of this minimization process or not as a valid hypothesis, and to remove conflicting hypotheses.

6.7 Related work on moment invariants

Several authors have considered moment based methods for object recognition and positioning, first for two-dimensional, and more recently for three-dimensional objects, spanning a period of almost thirty years [83, 5, 138, 123, 49, 46, 146, 98, 115, 116, 133, 112, 28, 55, 24, 86, 87, 1, 2, 108, 39, 93, 29, 132, 147, 94, 79, 80, 134, 135, 52, 131, 95].

The first to introduce moment invariants in the Pattern Recognition literature, was Hu [83]. He presented a theory of two-dimensional moment invariants for planar geometric figures based on the classic theory of algebraic invariants of binary forms. He derived complete systems of two-dimensional moment invariants under Euclidean transformations, expresing the rotations as multiplications by exponentials in the complex plane. The harmonic decomposition is equivalent to his method in the two-dimensional case. Some moment invariants under affine transformations are also included. For the second and third order moments he derives the following seven orthogonal invariants

$$\begin{split} \mathcal{I}_{1} &= M^{(2,0)} + M^{(0,2)} \\ \mathcal{I}_{2} &= (M^{(2,0)} - M^{(0,2)})^{2} + 4(M^{(1,1)})^{2} \\ \mathcal{I}_{3} &= (M^{(3,0)} - 3M^{(1,2)})^{2} + (3M^{(2,1)} - M^{(0,3)})^{2} \\ \mathcal{I}_{4} &= (M^{(3,0)} + M^{(1,2)})^{2} + (M^{(2,1)} + M^{(0,3)})^{2} \\ \mathcal{I}_{5} &= (M^{(3,0)} - 3M^{(1,2)})(M^{(3,0)} + M^{(1,2)})[(M^{(3,0)} + M^{(1,2)})^{2} - 3(M^{(2,1)} + M^{(0,3)})^{2}] + \\ & (3M^{(2,1)} - M^{(0,3)})(M^{(2,1)} + M^{(0,3)})[3(M^{(3,0)} + M^{(1,2)})^{2} - (M^{(2,1)} + M^{(0,3)})^{2}] \\ \mathcal{I}_{6} &= (M^{(2,0)} - M^{(0,2)})[(M^{(3,0)} + M^{(1,2)})^{2} - (M^{(2,1)} + M^{(0,3)})^{2}] + \\ & 4M^{(1,1)}(M^{(3,0)} + M^{(1,2)})(M^{(2,1)} + M^{(0,3)}) \\ \mathcal{I}_{7} &= (3M^{(2,1)} - M^{(0,3)})(M^{(3,0)} + M^{(1,2)})[(M^{(3,0)} + M^{(1,2)})^{2} - 3(M^{(2,1)} + M^{(0,3)})] - \\ & (M^{(3,0)} - 3M^{(1,2)})(M^{(2,1)} + M^{(0,3)})[3(M^{(3,0)} + M^{(1,2)})^{2} - (M^{(2,1)} + M^{(0,3)})^{2}] \,. \end{split}$$

These invariants are functionally equivalent to some of the invariants described above. For example, the first two are functionally equivalent to the two orthogonal invariants of the matrix $M_{[1,1]}$, because

$$\begin{aligned} \mathcal{I}_1 &= \operatorname{trace}(M_{[1,1]}) \\ \mathcal{I}_2 &= \left(\operatorname{trace}(M_{[1,1]})\right)^2 - 4\operatorname{det}(M_{[1,1]}) \,. \end{aligned}$$

Several other researchers used Hu's invariants for different purposes. Dudani, Breeding, and McGhee [49] use the seven two-dimensional moment invariants of Hu for the identification of aircraft from their projected contours. Wong and Hall [146] use the seven two-dimensional moment invariants of Hu for the matching of radar to optical images using a hierachical search technique with the moment invariants as similarity measures. Maitra [98] modifies Hu's seven orthogonal invariants to make then also invariant under scale and illumination changes. Sadjadi and Hall [115] study numerical methods for the evaluation of the Hu's seven moment invariants. Later [116], they partially extend Hu's work to the three-dimensional. Based on the theory of algebraic forms, they develop certain orthogonal invariants of quadratic and cubic forms, which are particular cases of the methods described in this and previous chapters.

Several authors derived moment invariants under constrained families of affine transformations, and also considered moments with respect to other functions, not polynomials. Alt [5] use moments as invariant features for the recognition of printed symbols, under a limited family of affine transformations. Teague [133] introduce Zernike moments as features for the recognition of two-dimensional patterns, and establish their relation with the usual moments. Reddi [112] defines angular and radial moments, and establishes their relation with Hu's seven moment invariants. Casasent, Cheatham, and Fetterly [28] describe and optical system to compute intensity moments of two-dimensional images. Boyce and Hossack [24] use Zernike moments for image reconstruction. Abu-Mostafa and Psaltis [1] evaluate the two-dimensional moment invariants as features for pattern recognition in terms of discrimination power and noise tolerance. Later [2], they consider a new normalization process for two-dimensional images based on complex moments. Cash and Hatamian [29] use moments as invariant features for the recognition of characters in printed documents. Teh and Chin [134, 135] compare different types of moments, regular moments, Legendre moments, Zernike moments, pseudo-Zernike moments, rotational moments, and complex moments, with respect to the representation and recognition of two-dimensional patterns.

Affine normalization of two-dimensional shapes is a subject treated by differen authors as well. Udagawa, Toriwaki, and Sugino [138] define a procedure for the normalization twodimensional patterns, capital letters in their examples, under affine transformations based on moments, and use the normalized moments as invariant features for recognition. Dirilten and Newman [46] are concerned with the problems of recognition and positioning of patterns under affine transformations. They show that there are infinitely many affine transformations which make the moments up to degree two of two patterns match, and two of them differ by an orthogonal transformation, following the same approach that we have followed, but they do not show a direct method to recover the unknown orthogonal transformation. They also derive certain orthogonal moment invariants by contracting indices of the symmetric moment tensors. These invariants can also be obtained with the methods described here. For simplicity, and because the treatment presented in the text was suficient for our purposes, we have deliberately ommited to introduce tensors, and to mention the relation between symmetric tensors and forms. Faber and Stokely [52] determine the affine transformation which relates two three-dimensional shapes by computing four pairs of covariant points using tensor-based techniques, and then solve the linear system which results from the pairing. These covariant points usualy involve moments of degree up to five. They also use the method of the principal directions of the tensor of inertia, a covariant matrix of second degree moments, for recovering Euclidean transformations.

Hong and Tan [79, 80] introduced the concept of moment curve of a set of points, as a tool for the affine normalization of planar shapes. The moment curve of a shape is an algebraic curve of degree two or three, with its coefficients functions of the second degree moments of set of points. It is a circunference if and only if the matrix of second degree moments $M_{[1,1]}$ is a multiple of the identity matrix, and two shapes are equivalent with respect to affine transformations, if and only if their corresponding moment curves are equivalent with respect to orthogonal transformations. They propose as a disimilarity function between two shapes, the minimum, over all the rotations, of an orthogonal disimilarity function between the cor-

responding moment curves. The orthogonal disimilarity function is based on heuristics, and involves rotating one curve to a finite number of angles, and comparing it with the other. Using the implicit equation of the moment curve, we could use the methods for curve positioning to improve their method, but it is less expensive to recover the affine transformation directly from the moments, as we have explained above.

The Euclidean and affine matching problems are also related to the motion estimation problem. Lin, Lee, and Huang [93] estimate the Euclidean transformation which transforms one set of points into a second one. The translation part is computing as the difference between the centers of both sets, and the rotation part by diagonalizing their scattering matrices $M_{[1,1]}$ and $M'_{[1,1]}$, obtaining, as we did, 2^n candidate solutions, or 2^{n-1} if only proper orthogonal matrices are allowed. The method that the propose for discriminating among these 2^{n-1} candidate transformations has a complexity function of the number of points, though.

A few authors have worked out extensions of Hu's invariants to the three-dimensional case. Pinjo, Cyganski and Orr [108, 39] describe moment based methods for the determination of the orientation of 3-D objects in 3-space either from 2-D projections or 3-D surface coordinate information. Their methods require the computation of moments up to degree five. Lo and Don [94, 95] develop three-dimensional orthogonal moment invariants using complex moments and the irreducible decomposition of the representation of the orthogonal group defined by these moments. This approach produces invariants which are functionally equivalent to those produced by what we have called the harmonic decomposition. They also determine 2^n candidate Euclidean transformations for matching two sets of points, by centering the moments and diagonalizing the matrix of second degree moments. They also discrimninate among these 2^n candidates by looking at third degree moments, as we do, obtaining a totaly equivalent method for position estimation.

6.8 **Proofs**

Proof of Lemma 18:

In the discreate case, the shape ν represents a subset of the shape μ if the data points of ν are part of the data points of μ . In the continuous case, ν represents a subset of μ if the weight functions satisfy $\nu(x) \le \mu(x)$ for almost every point x. But in both cases, for every continuous function f(x) they satisfy the inequality

$$\int f(x)^2 d\nu(x) \leq \int f(x)^2 d\mu(x) ,$$

which has to be interpreted as a sum in the discreate case. In particular, the inequality is true for every polynomial of degree $\leq d$. A polynomial of degree $\leq d$ can be written as $f(x) = FX_{(d)}(x)$, where F is a row vector of coefficients. Since

$$\int f(x)^2 \, d\nu(x) = |\nu| F\left(\frac{1}{|\nu|} \int X_{(d)}(x) X_{(d)}^t(x) \, d\nu(x)\right) F^t = |\nu| F \Sigma_{\nu,d} F^t \, ,$$

the previous inequality becomes for this particular case

$$|\nu| F \Sigma_{\nu,d} F^t \leq |\mu| F \Sigma_{\nu,d} F^t .$$

Since this inequality is true for every vector F we have an inequality between the symmetric matrices

$$|\nu| \Sigma_{\nu,d} \leq |\mu| \Sigma_{\nu,d}$$
.

Let *L* be any square matrix such that $L\Sigma_{\mu,d}L^t = I$. The matrix $L\Sigma_{\nu,d}L^t$ is symmetric and satisfies the inequality

$$L\Sigma_{\nu,d}L^t \leq \frac{|\mu|}{|\nu|}I$$

The same inequality is mantained after computing the trace on both sides

trace
$$\left(L\Sigma_{\nu,d}L^{t}\right) \leq \frac{|\mu|}{|\nu|}$$
 trace $(I) = \frac{|\mu|}{|\nu|} {n+d \choose n}$. (6.10)

But

trace
$$(L\Sigma_{\nu,d}L^t)$$
 = trace $(L^t L\Sigma_{\nu,d})$ = trace $(\Sigma_{\mu,d}^{-1}\Sigma_{\nu,d})$ = $\Delta_{\mu,d}(\nu)$. (6.11)

Finally, from (6.10) and (6.11), we obtain

$$\Delta_{\mu,d}(\nu) \leq \frac{|\mu|}{|\nu|}$$
.

Chapter 7 Recognition Algorithms

In the introductory chapter we described the basic structure of the recognition and positioning systems which can be built using the tools developed in previous chapters. However, such a general description admits several different implementations, all of them using the same building blocks. We can divide the best known implementations which could make use of the representation and matching techniques described in this thesis, into two large groups: tree-search schemes, and voting schemes.

Tree-search algorithms are generally sequential algorithms which generate a hypothesis based on some local information, and then try to search for more evidence sustaining the current hypothesis, before generating a new hypothesis. The combinatorial complexity of the search is limited based on geometrical considerations. As typical examples of tree-search schemes, we can cite the works of Bolles and Horaud [19, 18], Faugeras and Hebert [55, 56], Grimson and Lozano-Perez [72, 71], Grimson [70], and Chen and Kak [32]. All of these systems are based on modeling objects as piecewise algebraic surfaces, but using very simple surface patches such as planes and certain quadrics. For matching and positioning they consider different special cases, an this is where the unified treatment described in previous chapters could be use to improve the performance of these methods.

Voting schemes are better suited for parallel architectures [136], and are generally based on extensions of Ballard's generalized Hough transform [9]. A typical example of voting scheme procedures is the work of Bolle, Califano, Kjeldsen, and Taylor [15, 14]. A different approach within the voting scheme procedures is a family of algorithms known as *geometric hashing*. These algorithms have been developed by Schwartz and Sharir [119], Lamdan, Schwartz and Wolfson [89] Lamdan and Wolfson [90], Kishon and Wolfson [88], Hong and Wolfson [81], Lamdan [91], and Wolfson [145]. In the geometric hashing algrithms, the database is implemented as a hash table indexed by a vector of coordinate invariant features. Since the models, and their visible regions, are known in advance, the hash function can be optimized according to the typical operating conditions, such as the amount of occlusion. Every succesfull local match generates a hypothesis, consisting of a pair object-coordinate system. With each hypothesis a vote is generated a discretized parameter space corresponding to the parameters of the coordinate transformations. The weight of the vote with is proportional to certain more global matching measure. Finally, the hypotheses with larger votes are globally tested.

Chapter 8 Conclusions

In this thesis we have described several algebro-geometric techniques necessary for the implementation of model based object recognition systems in clutered environments. The emphasis of the thesis is on viewpoint independent, or *invariant* methods to describe, compare, and orient small regions of a data set. Models are fitted to regions of the data set, and vectors of geometric invariants of the model parameters are used as the basis for recognition. These geometric invariants are functions of the parameters which do not change when tha data undergo a viewpoint transformation, while the model parameters do. Closely related to the computation of these invariants are the methods introduced for recovering the unknown transformations which make the matching regions coincide. Two types of models have been studied, algebraic curves and surfaces, and moments. A good part of the thesis has been devoted to developing methods for fitting algebraic curves and surfaces, or more generaly implicit curves and surfaces, to data sets. Computing moments of a data set is relatively strightforward. These two families of models have a lot in common in the area of invariant theory, and a several methods developed for the algebraic curve and surface models are used without modification to compute moment invariants. The same is true with respect to defining a canonical orientation for algebraic curves and surfaces, or moments. Our emphasis throughout this thesis has been on developing numerical methods based on well known, numerically stable and computationally efficient techniques. Besides the extensive algebro-geometric analysis of the problems under consideration, it has been central to both the algebraic curve and surface fitting techniques, and the computation of invariants, the large body of well established and well anlyzed matrix algebra algorithms, in particular the algorithms for computing egenvalues and eigenvectors of symmetric matrices.

We believe that this thesis represents an important contribution to the field of Computer Vision. Several of the techniques introduced in this thesis represent new concepts in the field, and most of them yield both conceptual and computational improvements with respect to other methods used by other researchers for similar purposes.
8.1 Future directions

Although we have implemented and tested in typical situations all the algorithms described in this thesis, recognition systems based on these methods still have to be implemented and tested. We are planning to implement some of these recognition systems in the near future. Several implementation specific problems will arise and we will address them at the proper time.

Other problems which we are planning to address in the near future are related to the robustness of the methods introduced in this thesis. We are interested in both the robustness of the curve and surface fitting techniques, the discrimination power of different invariants computed from the same model parameters, and the accuracy of the position estimation. That is, since all the mesurements contain noise, we have to look at the computational methods introduced in this thesis as statistical parameter estimation problems, and establish how good they are based on analizing their estimation power.

In this thesis, the intrinsic coordinate system of algebraic curves and surfaces is defined with respect to Euclidean transformations only, while the intrinsic coordinate system of moments is defined also with respect to affine transformations. Although it is not posible to find an intrinsic coordinate system for moments with respect to projective transformations, it is not clear whether it is posible or not to find a projective intrinsic coordinate system for algebraic curves and surfaces. Since the solution of this problem is related to the recognition and positioning of 3D objects from the projections of their occluding boundaries, it is important to find an answer to this problem. This problem is closely related to the problem of extending the techniques for computing invariants to the computation of invariants of 3D surfaces from invariants of the curves which are the projections of their occluding boundaries. We are planning to study these problems as well.

Appendix A The Levenberg-Marquardt algorithm

Let $\mathcal{D} = \{p_1, \ldots, p_q\}$ be a set of *n*-dimensional data points, and let $\phi(\alpha, x)$ a parameterization of the family of admissible maps \mathcal{F} . The nonlinear least squares problem is to minimize the length of the *residual vector* $R = (R_1, \ldots, R_q)^t$

$$||R(\alpha)||^2 = \sum_{i=1}^q R_i(\alpha)^2 = q \Delta_{\mathcal{D}}^2(\alpha)$$

where in our case

$$R_i(\alpha) = \delta(\alpha, p_i) \qquad i = 1, \dots, q$$
,

and the number of points is not less than the number of parameters. The Levenberg-Marquardt algorithm, one of several methods to solve the nonlinear least squares problem, is based on the following iteration step

$$\alpha^{n+1} = \alpha^n - \left(J(\alpha^n)J(\alpha^n)^t + \mu_n \mathbf{I}_q\right)^{-1} J(\alpha^n)^t R(\alpha^n) ,$$

where $J(\alpha)$ is the Jacobian of *R* with respect to α

$$J_{ij}(\alpha) = \frac{\partial R_i}{\partial \alpha_j}(\alpha) \qquad i = 1, \dots, q \quad j = 1, \dots, r ,$$

and the constant μ_n is chosen as a small nonnegative number, equal to zero whenever possible, which makes the matrix

$$J(\alpha^n)J(\alpha^n)^t + \mu_n \mathbf{I}_q$$

safely positive definite. This strategy assures that the algorithm reduces the value of $||R(\alpha)||^2$ at each iteration, converging to a local minimum, with fast quadratic local convergence. See Dennis and Shnabel [42] for details.

We only need to provide procedures to compute the values of the residual vector $R(\alpha)$ and the Jacobian $J(\alpha)$. Since all the components of the residual vector have the same form, we only need a procedure to compute $\delta(\alpha, x)$ and its partial derivatives with respect to $\alpha_1, \ldots, \alpha_r$.

For example, let us consider the linear parameterization of planar curves and surfaces, k = 1. In this case we have $f(x) = \phi(\alpha, x) = FX$, where $F = \alpha^t$ is an *r*-dimensional row vector,

$$\delta(F,x) = \left(\frac{F[XX^t]F^t}{F[DXDX^t]F^t}\right)^{1/2} = \left(\frac{f(x)^2}{nablaf(x)\|^2}\right)^{1/2}$$

and

$$\frac{\partial \delta}{\partial F_j}(F, x) = \frac{1}{\delta(F, x)[FDX][FDX]^t} \left\{ [FX]X_j - \delta(F, x)^2 [FDX]DX_j^t \right\}$$

In the case of cylinders, and in general in all the cases of parameterized families of polynomials, where we can write f(x) = FX, with the coefficient vector as a function of the parameters $F = F(\alpha)$, we just apply the chain rule to the previous expression

$$\frac{\partial \delta}{\partial \alpha_i}(\alpha, x) = \sum_{j=1}^h \frac{\partial \delta}{\partial F_j}(F(\alpha), x) \frac{\partial F_j}{\alpha_i}(\alpha) .$$

Bibliography

- [1] Y.S. Abu-Mostafa and D. Psaltis. Recognitive aspects of moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:698–706, 1984.
- [2] Y.S. Abu-Mostafa and D. Psaltis. Image normalization by complex moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:46–55, 1985.
- [3] M. Ackerman. *Hilbert's Invariant Theory Papers,* volume VIII of *Lie Groups: History, Frontiers and Applications.* Math Sci Press, Brookline, Massachusets, 1978.
- [4] A. Albano. Representation of digitized contours in terms of conic arcs and stright-line segments. *Computer Graphics and Image Processing*, 3:23–33, 1974.
- [5] F.L. Alt. Digital pattern recognition by moments. *The Journal of the Association for Computer Machinery*, 9:240, 1962.
- [6] T.M. Apostol. *Mathematical Analysis*. Addison-Wesley, 1974.
- [7] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.
- [8] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In Proceedings, First International Conference on Computer Vision, pages 231–240, June 1987.
- [9] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [10] A.H. Barr. Superquadrics and angle-preserving transformations. IEEE Computer Graphics and Applications, 1:11–23, 1981.
- [11] P.J. Besl. Surfaces in Range Image Understanding. Springer-Verlag, 1988.
- [12] P.J. Besl and R.C. Jain. Segmentation trough variable-order surface fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, March 1988.
- [13] R.H. Biggerstaff. Three variations in dental arch form estimated by a quadratic equation. *Journal of Dental Research*, 51:1509, 1972.

- [14] R.M. Bolle, A. Califano, R. Kjeldsen, and R.W. Taylor. Computer Vision Research at the IBM T.J. Watson Research Center. In *Proceedings*, *DARPA Image Understanding Workshop*, pages 471–478, 1989.
- [15] R.M. Bolle, A. Califano, R. Kjeldsen, and R.W. Taylor. Visual recognition using concurrent and layered parameter networks. In *Proceedings*, *IEEE Conference on Computer Vision and Pattern Recognition*, pages 625–631, June 1989.
- [16] R.M. Bolle and D.B. Cooper. Bayesian recognition of local 3D shape by approximating image intensity functions with quadric polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4), July 1984.
- [17] R.M. Bolle and D.B. Cooper. On optimally combining pieces of information, with applications to estimating 3D complex-object position from range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5), September 1986.
- [18] R.C. Bolles and P. Horaud. 3DPO: A three-dimensional part orientation system. *The International Journal of Robotics Research*, 5(3):3–26, 1986.
- [19] R.C. Bolles, P. Horaud, and M.J. Hannah. 3DPO: a three-dimensional part orientation system. In *Proceedings*, 8th. International Joint Conference on Artificial Intelligence, 1983.
- [20] F.L. Bookstein. Fitting conic sections to scattered data. Computer Vision, Graphics, and Image Processing, 9:56–71, 1979.
- [21] G. Boole. Exposition of a general theory of linear transformations. *Cambridge Mathematical Journal*, 3:1–20, November 1841.
- [22] T.E. Boult and A.D. Gross. Recovery of superquadrics from depth information. In Proceedings, AAAI Workshop on Spatial Reasoning and Multisensor Integration, October 1987.
- [23] T.E. Boult and A.D. Gross. On the recovery of superellipsoids. CUCS 331-88, Columbia University, Computer Science Department, 1988.
- [24] J.F. Boyce and W.J. Hossack. Moment invariants for pattern recognition. Pattern Recognition Letters, 1:451–456, 1983.
- [25] R.A. Bradley and S.S. Srivastava. Correlation in polynomial regression. *The American Statistician*, 33:11, 1979.
- [26] R.W. Brockett. Least squares matching problems. *Linear Algebra and its Applications*, pages 761–777, 1989.
- [27] J.F. Canny. *The complexity of robot motion planning*. The MIT Press, 1988.
- [28] D. Casasent, L. Cheatham, and D. Fetterly. Optical system to compute intensity moments: Design. *Applied Optics*, 21:3292–3298, 1982.

- [29] G.L. Cash and M. Hatamian. Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, 39:291–310, 1987.
- [30] A. Cayley. *The collected mathematical papers of Arthur Cayley*. Cambridge University Press, Cambridge, 1889–1897.
- [31] B. Cernuschi-Frias. Orientation and location parameter estimation of quadric surfaces in 3D from a sequence of images. PhD thesis, Brown University, May 1984.
- [32] C.H. Chen and A.C. Kak. A robot vision system for recognizing 3D objects in loworder polynomial time. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1535– 1563, November/December 1989.
- [33] D.S. Chen. A data-driven intermediate level feature extraction algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), July 1989.
- [34] A. Clebsh. *Theorie der Binären Algebraischen Formen*. B.G. Teubner, Leipzig, 1872.
- [35] D.B. Cooper, Y.P. Hung, and G. Taubin. A new model-based stereo approach for 3D surface reconstruction using contours on the surface pattern. In *Proceedings, Second International Conference on Computer Vision*, pages 74–83, December 1988.
- [36] D.B. Cooper and N. Yalabik. On the cost of approximating and recognizing noiseperturbed straight lines and quadratic curve segments in the plane. NASA-NSG-5036/I, Brown University, March 1975.
- [37] D.B. Cooper and N. Yalabik. On the computational cost of approximating and recognizing noise-perturbed straight lines and quadratic arcs in the plane. *IEEE Transactions* on Computers, 25(10):1020–1032, October 1976.
- [38] C.P. Cox. A Handbook of Introductory Statistical Methods. John Wiley & Sons, 1987.
- [39] D. Cygansky and J.A. Orr. Applications of tensor theory to object recognition and orientation determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:662–673, 1985.
- [40] P.J. Davis. *Interpolation and Approximation*. Dover Publications, Inc., 1975.
- [41] C. De Boor. A practical guide to splines. Springer-Verlag, 1978.
- [42] J.E. Dennis and R.B. Shnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.
- [43] L.E. Dickson. *Algebraic Invariants*, volume 14 of *Mathematical Monographs*. John Wiley & Sons, Inc., New York, 1st. edition, 1914.
- [44] J.A. Dieudonné. La théorie des invariants au XIX^e siècle. In *Lecture Notes in Mathematics* 244, Séminaire Bourbaki vol. 1970/71, chapter Exposé n° 395, pages 257–274. Springer-Verlag, 1971.

- [45] J.A. Dieudonné and J.B. Carrell. *Invariant Theory, Old and New*. Academic Press, New York and London, 1971.
- [46] H. Dirilten and T.G. Newman. Pattern matching under affine transformations. *IEEE Transactions on Computers*, 26:314–317, 1977.
- [47] A.L. Dixon. The elimination of three quantics in two independent variables. *Proceed*ings of the London Mathematical Society, 7, 1908. Series 2.
- [48] R.O Duda and P.E. Hart. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973.
- [49] S.A. Dudani, K.J. Breeding, and R.B. McGhee. Aircraft identification by moment invariants. *IEEE Transactions on Computers*, 26:39–45, 1977.
- [50] E.B. Elliot. *An Introduction to the Algebra of Quantics*. Oxford University Press, Oxford, 2nd. edition, 1913.
- [51] I. Elliot. Discussion and implementation description of an experimental interactive superquadric-based 3D drawing system. Internal Report LP 2/IME4, European Computer-Industry Research Center, 1986.
- [52] T.L. Faber and E.M. Stokely. Orientation of 3-D structures in medical images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:626–633, 1988.
- [53] R.T. Farouki and V.T. Rajan. On the numerical condition of algebraic curves and surfaces. RC-13263, IBM Research Division, November 1987.
- [54] R.T. Farouki and V.T. Rajan. On the numerical condition of berstein polynomials. RC-12626, IBM Research Division, March 1987.
- [55] O.D. Faugeras and M. Hebert. A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces. In *Proceedings*, 8th. International Joint Conference on Artificial Intelligence (IJCAI), 1983.
- [56] O.D. Faugeras and M. Hebert. The representation, recognition, and location of 3-D objects. *The International Journal of Robotics Research*, 5(3):27–52, 1986.
- [57] O.D. Faugeras, M. Hebert, and E Pauchon. Segmentation of range data into planar and quadric patches. In *Proceedings*, *IEEE Conference on Computer Vision and Pattern Recognition*, 1983.
- [58] D. Forsyth, J.L. Mundy, A. Zisserman, and C.M. Brown. Projectively invariant representation using implicit algebraic curves. In *Proceedings, First European Conference on Computer Vision*, 1990.
- [59] Y. Gaardan. *Numerical methods for CAD*. The MIT Press, 1986.

- [60] B.S. Garbow, J.M. Boyle, J.J. Dongarra, and C.B. Moller. Matrix Eigensystem Routines - EISPACK Guide Extension, volume 51 of Lecture Notes in Computer Science. Springer-Verlag, 1977.
- [61] M. Gardiner. The superellipse: a curve that lies between the ellipse and the rectangle. *Scientific American*, 213(3):222–234, September 1965.
- [62] D.B. Gennery. Object detection and measurement using stereo vision. In Proceedings, DARPA Image Understanding Workshop, pages 161–167, April 1980.
- [63] A.A. Giordano and F.M. Hsu. *Least squares estimation with applications to digital signal processing*. Wiley, 1985.
- [64] R Gnanadesikan. Methods for Statistical Data Analysis of Multivariate Observations. Wiley, 1977.
- [65] R.N. Goldman and T.W. Sederberg. Some applications of resultants to problems in computational geometry. *The Visual Computer*, 1:101–107, 1985.
- [66] G. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 1983.
- [67] G.H. Golub and R. Underwood. Stationary values of the ratio of quadratic forms subject to linear constraints. Z. Angew. Math. Phys., 21:318–326, 1970.
- [68] P. Gordan and G. Kerschensteiner. Vorlensungen über Invariantentheorie. ?, Liepzig, 1887.
- [69] J.H. Grace and A. Young. *The Algebra of Invariants*. Cambridge University Press, 1903.
- [70] W.E. Grimson. On the recognition of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):632–643, June 1989.
- [71] W.E. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, July 1987.
- [72] W.E.L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *The International Journal of Robotics Research*, 3(3), 1984.
- [73] A.D. Gross and T.E. Boult. Error of fit measures for recovering parametric solids. In *Proceedings, Second International Conference on Computer Vision,* December 1988.
- [74] G.B. Gurevich. *Foundations of the Theory of Algebraic Invariants*. P. Noordhoff Ltd., Groningen, The Netherlands, 1964.
- [75] E.L. Hall, J.B.K. Tio, C.A. McPherson, and F.A. Sadjadi. Measuring curved surfaces for robot vision. *IEEE Computer*, pages 42–54, December 1982 1982.

- [76] S. Helgason. *Groups and Geometric Analysis*. Academic Press, New York and London, 1984.
- [77] D. Hilbert. Über die Theorie der Algebraischen Formen. Math. Ann., 36:473, 1890.
- [78] D. Hilbert. Über die vollen Invariantensysteme. *Math. Ann.*, page 313, 1893.
- [79] J. Hong and X. Tan. A new approach to point pattern matching. Robotics Report 132, Courant Institute of Mathematical Sciences, New York University, New York, December 1987.
- [80] J. Hong and X. Tan. The similarity between shapes under affine transformation. Robotics Report 133, Courant Institute of Mathematical Sciences, New York University, New York, December 1987.
- [81] J. Hong and H.J. Wolfsn. An improved model-based matching method using footprints. In *Proceedings, International Conference on Pattern Recognition,* Rome, Italy, November 1988.
- [82] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educ. Psychol.*, 24:417–444,498–520, 1933.
- [83] Ming-Kuei Hu. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, 8:179–187, 1962.
- [84] E. Isaacson and H.B. Keller. *Analysis of Numerical Methods*. Wiley, New York, 1966.
- [85] I.T. Jollife. Principal Component Analysis. Springer Series in Statistics. Springer-Verlag, New York, 1986.
- [86] Ken-Ichi Kanatani. Distribution of directional data and fabric tensors. *International Journal of Engineering Science*, 22:149–164, 1984.
- [87] Ken-Ichi Kanatani. Stereological determination of structural anisotropy. *International Journal of Engineering Science*, 22:531–546, 1984.
- [88] E. Kishon and H. Wolfson. 3D curve matching. In *Proceedings, Workshop on Spacial Reasoning and Multi-Sensor Fusion*, pages 250–261, 1987.
- [89] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Object recognition by affine invariant matching. In *Proceedings*, *IEEE Converence on Computer Vision and Pattern Recognition*, pages 335–344, 1988.
- [90] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings, Second International Conference on Computer Vision*, pages 238–249, December 1988.

- [91] Yehezkel Lamdan. Object Recognition by Geometric Hashing. PhD thesis, New York University, Courant Institute of Mathematical Sciences, Computer Science Division, August 1989. Also, Robotics Report No. 206.
- [92] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.
- [93] Z.C. Lin, H. Lee, and T.S. Huang. Finding 3-d point correspondences in motion estimation. In *Proceedings, Eight International Conference on Pattern Recognition*, pages 303–305, Paris, October 1986.
- [94] C.H. Lo and H.S. Don. Object identification and positioning via 3-d moments: a grouptheoretic approach. Technical Report 87-06, Laboratory for Image Analysis and Graphics, SUNY, Stony Brook, October 1987.
- [95] C.H Lo and H.S. Don. 3-D moment forms: Their construction and application to object identification and positioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1053–1064, 1989.
- [96] E.A. Lord and C.B. Wilson. *The mathematical description of shape and form*. Ellis Horwood, Chichester, 1984.
- [97] F.S. Macaulay. *The algebraic theory of modular systems*. Cambridge University Press, 1916.
- [98] S. Maitra. Moment invariants. *Proceedings of the IEEE*, 67:697–699, 1979.
- [99] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11:431–441, 1963.
- [100] J.J. Moré, B.S. Garbow, and K.E. Hillstrom. User guide for MINPACK-1. ANL-80-74, Argone National Laboratories, 1980.
- [101] D. Mumford. *Geometric Invariant Theory*, volume 34 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer-Verlag, New York, 1965.
- [102] A. Nadás. Least squares and maximum likelihood estimation of rigid motion. Technical Report RC-6945, IBM Research, Yorktown Heights, 1978.
- [103] K.A. Paton. Conic sections in authomatic chromosome analysis. *Machine Intelligence*, 5:411, 1970.
- [104] K.A. Paton. Conic sections in chromosome analysis. *Pattern Recognition*, 2:39, 1970.
- [105] C. Pearson. *Numerical methods in engineering and science*. Van Nostrand Reinhold, 1986.
- [106] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philos. Mag.*, 2:559, 1901. Series 2.

- [107] A.P. Pentland. Recognition by parts. In *Proceedings, First International Conference on Computer Vision*, pages 612–620, June 1987.
- [108] Z. Pinjo, D. Cyganski, and J.A. Orr. Determination of 3-d object orientation from projections. *Pattern Recognition Letters*, 3:351–356, 1985.
- [109] J. Ponce and D.J. Kriegman. On recognizing and positioning curved 3D objects from image contours. In *Proceedings, IEEE Workshop on Interpretation of 3D Scenes,* November 1989.
- [110] V. Pratt. Direct least squares fitting of algebraic surfaces. *Computer Graphics,* pages 145–152, July 1987. (Proceedings SIGGRAPH'87).
- [111] J.O. Ramsay. A comparative study of several robust estimates of slope, intercept, and scale in linear regression. *Journal of the American Statistical Association*, 72:608–615, 1977.
- [112] S.S. Reddi. Radial and angular moment invariants for image identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:240–242, 1981.
- [113] M. Rioux and L. Cournoyer. The NRCC three-dimensional image data files. CNRC 29077, National Research Council of Canada, June 1988.
- [114] G.C. Rota and B. Sturmfels. Introduction to Invariant Theory in Superalgebras, volume 19, Invariant Theory and Tableaux of The IMA Volumes in Mathematics and Its Applications, pages 1–35. Springer-Verlag, New York, 1989.
- [115] F.A. Sadjadi and E.L. Hall. Numerical computations of moment invariants for scene analysis. In *Proceedings, IEEE Conference on Pattern Recognition and Image Processing,* pages 181–187, 1978.
- [116] F.A. Sadjadi and E.L. Hall. Three-dimensional moment invariants. *IEEE Transactions* on Pattern Analysis and Machine Intelligence, 2:127–136, 1980.
- [117] G. Salmon. *Modern higher algebra*. Hodges, Smith and Co., Dublin, 1866.
- [118] P.D. Sampson. Fitting conic sections to very scattered data: an iterative refinment of the bookstein algorithm. *Computer Vision, Graphics, and Image Processing*, 18:97–108, 1982.
- [119] J. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *The International Journal of Robotics Research*, 6(2), 1987.
- [120] T.W. Sederberg, D.C. Anderson, and R.N. Goldman. Implicit representation of parametric curves and surfaces. *Computer Vision, Graphics, and Image Processing*, 28(1):72– 84, 1984.

- [121] J.F. Silverman and D.B. Cooper. Bayesian clustering for unsupervised estimation of surface and texture models. *IEEE Transactions on Pattern Analysis and Machine Intelli*gence, 10, July 1988.
- [122] B.T. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ikebe, V.C. Klema, and C.B. Moller. *Matrix Eigensystem Routines - EISPACK Guide*, volume 6 of *Lecture Notes in Computer Science*. Springer-Verlag, 1976.
- [123] F.W. Smith and M.H. Wright. Automatic ship photo interpretation by the method of moments. *IEEE Transactions on Computers*, 20:1089–1095, 1971.
- [124] F. Solina. *Shape Recovery and Segmentation with Deformable Part Models*. PhD thesis, University of Pennsylvania, December 1987.
- [125] T.A. Springer. *Invariant Theory*, volume 585 of *Lecture Notes in Mathematics*. Springer-Verlag, New York, 1977.
- [126] J.J. Sylvester. *Collected Mathematical Papers*. Cambridge University Press, Cambridge, 1904–1912.
- [127] G. Taubin. Algebraic nonplanar curve and surface estimation in 3-space with applications to position estimation. Technical Report LEMS-43, Brown University, February 1988. Also, IBM Technical Report RC-13873.
- [128] G. Taubin. Nonplanar curve and surface estimation in 3-space. In *Proceedings, IEEE Conference on Robotics and Automation*, pages 644–645, May 1988.
- [129] G. Taubin. About shape descriptors and shape matching. Technical Report LEMS-57, Brown University, March 1989.
- [130] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. Technical Report LEMS-66, Brown University, January 1990.
- [131] G. Taubin, R.M. Bolle, and D.B. Cooper. Representing and comparing shapes using shape polynomials. In *Proceedings*, *IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–516, 1989.
- [132] R.W. Taylor and A.P. Reeves. Three-dimensional image transformations in moment space. In *Proceedings*, *IEEE* ?, pages 366–368, 1987.
- [133] M.R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70:920–930, 1980.
- [134] Cho-Huak Teh and R.T. Chin. On image analysis by the methods of moments. In Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, pages 556–561, 1988.

- [135] Cho-Huak Teh and R.T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:496–5113, 1988.
- [136] L.W. Tucker, C.R. Feynman, and Fritzsche D.M. Object recognition using the connection machine. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 871–878, 1988.
- [137] K. Turner. *Computer Perception of Curved Objects using a Television Camara*. PhD thesis, University of Edimburgh, November 1974.
- [138] K. Udagawa, J. Toriwaki, and K. Sugino. Normalization and recognition of twodimensional patterns with linear distortion by moments. *Electronics and Communications in Japan*, 47:34–46, 1964.
- [139] B.C. Vemuri. *Representation and recognition of objects from dense range maps*. PhD thesis, University of Texas at Austin, Department of Electrical Engineering, 1987.
- [140] B.C. Vemuri, A. Mitiche, and J.K. Aggarwal. Curvature-based representations of objects from range data. *Image and Vision Computing*, 4(2):107–114, May 1986.
- [141] R. Walker. *Algebraic Curves*. Princeton University Press, 1950.
- [142] I. Weiss. Projective invariants of shapes. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 291–297, 1988.
- [143] H. Weyl. The Classical Groups. Princeton University Press, Princeton, New Jersey, 1939.
- [144] N. White. *Implementation of the Straightening Algorithm*, volume 19, Invariant Theory and Tableaux of *The IMA Volumes in Mathematics and Its Applications*, pages 36–45. Springer-Verlag, New York, 1989.
- [145] H. Wolfson. On curve matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):483–489, May 1990.
- [146] R.Y. Wong. Hall, e.l. *Scene Matching with Invariant Moments*, 8:16–24, 1978.
- [147] M.F. Zakaria, L.J. Vroomen, P.J.A. Zsombor-Murray, and J.M.H.M. Van Kessel. Fast algorithm for the computation of moment invariants. *Pattern Recognition*, 20:639–643, 1987.