UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LEANDRO AUGUSTO FRATA FERNANDES

On the Generalization of Subspace Detection in Unordered Multidimensional Data

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Prof. Dr. Manuel Menezes de Oliveira Neto
Advisor

Porto Alegre, august 2010

CIP – CATALOGING-IN-PUBLICATION

Fernandes, Leandro Augusto Frata

On the Generalization of Subspace Detection in Unordered Multidimensional Data / Leandro Augusto Frata Fernandes. – Porto Alegre: PPGC da UFRGS, 2010.

154 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2010. Advisor: Manuel Menezes de Oliveira Neto.

1. Subspace detection. 2. Pattern recognition. 3. Subspace parameterization. 4. Parameter space. 5. Generalization. 6. Error propagation. 7. Grassmannian. 8. Hough transform. 9. Geometric algebra. 10. Clifford algebra. I. Oliveira Neto, Manuel Menezes de. II. Título.

To my fiancée Raquel Franco
and to my parents João Fernandes and Sônia Regina Frata Fernandes.

ACKNOWLEDGMENTS

I offer my sincerest gratitude to my advisor, Manuel Menezes de Oliveira Neto, who has supported me throughout my thesis with his friendship, patience and knowledge whilst allowing me the room to work in my own way.

I wish to express my appreciation to my dissertation committee: Professors Jayme Vaz Jr., Roberto M. Cesar Jr., Cláudio R. Jung and João L. D. Comba. All these gentlemen have contributed valuable suggestions to this work.

I would also like to warmly acknowledge Professors Jorge Stolfi and Luis G. D. Mender, who provided valuable discussions on Plücker coordinates and the Grassmannian. I thank John A. Small and Joseph R. Michael, and Mikhail Matrosovich for kindly providing the electron backscatter diffraction image and the clonogenic assay picture used as examples in this project.

In my daily work I have been blessed with a friendly and cheerful group of fellow students. Most especially, I would like to thank André S. Spritzer, Barbara Bellaver, Carlos A. Dietrich, Christian A. Pagot, Daniel K. Osmari, Fernando Trebien, Francisco M. Pinto, Giovane R. Kuhn, Jonatas Medeiros, Juan M. Ibiapina, Juliano M. Franz, Kao Felix, Leonardo Chatain, Leonardo Fischer, Luis F. Mendes, Luiz F. Scheidegger, Márcio R. Zacarias Marilena Maule, Rafael P. Torchelsen, Renato Silveira, Vitor A. M. Jorge, and Vitor F. Pamplona for many helpful discussions and some good laughs. Also, I would like to thank Professors Carla M. D. S. Freitas, Edson Prestes e Silva Jr., João L. D. Comba, Luciana P. Nedel and Roberto da Silva for encouraging to me during the planning stages of this thesis.

The PPGC-UFRGS has provided the support and equipment I have needed to produce and complete my thesis, and the CNPq has funded my studies (process 142627/2007-0).

My parents deserve special mention for their inseparable support and prays. My father, João Fernandes, and my mother, Sônia Regina Frata Fernandes, are the ones who sincerely raised me with their caring and gently love. I thank Thais Regina Frata Fernandes, my sister, for being a supportive and caring sibling.

Words fail me to express my appreciation to my fiancée Raquel Franco whose dedication, love, patience and persistent confidence on me, has taken the load off my shoulders.

Finally, I would like to thank everybody who was important to the successful realization of this thesis, as well as expressing my apology that I could not mention personally one by one.

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

GA	Geometric Algebra
GATL	Geometric Algebra Template Library
GHT	Ballard's Generalized Hough Transform (BALLARD, 1981)
HT	Hough Transform
MOG	Model of Geometry
PDF	Probability Distribution Function
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
TV	Tensor Voting

LIST OF SYMBOLS

SPACES

$\mathbb{N}_0$	Set of all natural numbers including zero.
$\mathbb{Z}$	Set of all integer numbers.
$\mathbb{R}$	Set of all real numbers.
$\mathbb{R}^n$	n-Dimensional vector space over $\mathbb{R}$.
$\bigwedge \mathbb{R}^n$	Multivector space built from $\mathbb{R}^n$.
$\bigwedge^k \mathbb{R}^n$	The part of $\bigwedge \mathbb{R}^n$ in which k-blades reside.
$\mathbb{A}^m$	m-Dimensional affine space.
$\mathbb{P}^m$	m-Dimensional parameter space.
$\mathbb{D}^m$	m-Dimensional discrete representation of the parameter space.
$G(k,n)$	Grassmannian, the set of all k-dimensional linear subspaces of $\mathbb{R}^n$.

GEOMETRIC ALGEBRA

ϕ, θ	Angle in radians.
α, β, etc.	General scalar value (0-blade).
x, y, etc.	General scalar value (0-blade), typically point coordinates.
$\mathbf{a}$, $\mathbf{b}$, etc.	General vector (1-blade).
$\mathbf{A}_{\langle k \rangle}$, $\mathbf{B}_{\langle k \rangle}$, etc.	General k-blade, for $0 \leq k \leq n$, defined as the outer product of k vectors.
$\boldsymbol{A}$, $\boldsymbol{B}$, etc.	General rotor, defined as the geometric product of an even number of unit invertible vectors.
A, B, etc.	General multivector, not necessarily a blade or a rotor. The i-th coefficient is denoted as α^i, β^i, etc.
$\mathbf{e}_i$	i-th Unit basis vector, typically in an orthonormal basis.
$\mathbf{e}_0$	Unit basis vector interpreted as origin point in homogeneous model.

$\mathbf{o}$	Null basis vector interpreted as origin point in conformal model.
$\boldsymbol{\infty}$	Null basis vector interpreted as point at infinity in conformal model.
$\mathbf{I}_{\langle n \rangle}$	Unit pseudoscalar of a n-dimensional space, defined as $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_n$.
$\langle A \rangle_k$	Retrieves the k-dimensional part of A.
$A\,B$	Geometric product of A and B.
$A \,/\, B$	Inverse geometric product, defined as $A\,B^{-1}$.
Q	Scalar-valued function defining a metric on $\mathbb{R}^n$.
$\mathbf{a} \cdot \mathbf{b}$	Vector inner product, defined as $\mathsf{Q}\,(\mathbf{a},\mathbf{b})$.
$\mathbf{A}_{\langle r \rangle} * \mathbf{B}_{\langle s \rangle}$	Scalar product, defined as $\left\langle \mathbf{A}_{\langle r \rangle}\,\mathbf{B}_{\langle s \rangle} \right\rangle_0$.
$\mathbf{A}_{\langle r \rangle} \rfloor \mathbf{B}_{\langle s \rangle}$	Left contraction, defined as $\left\langle \mathbf{A}_{\langle r \rangle}\,\mathbf{B}_{\langle s \rangle} \right\rangle_{s-r}$.
$\mathbf{A}_{\langle r \rangle} \wedge \mathbf{B}_{\langle s \rangle}$	Outer product, defined as $\left\langle \mathbf{A}_{\langle r \rangle}\,\mathbf{B}_{\langle s \rangle} \right\rangle_{r+s}$.
$\mathbf{A}_{\langle r \rangle} \vee \mathbf{B}_{\langle s \rangle}$	Regressive product, defined as $(\mathbf{B}^*_{\langle s \rangle} \wedge \mathbf{A}^*_{\langle r \rangle})^{-*}$.
$\mathbf{A}_{\langle r \rangle} \cap \mathbf{B}_{\langle s \rangle}$	Meet, defined as the subspace shared by $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$.
$\mathbf{A}_{\langle r \rangle} \cup \mathbf{B}_{\langle s \rangle}$	Join, defined as the subspace spanned by the disjoint and by the common parts of $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$.
$\widetilde{\mathbf{A}}_{\langle k \rangle}$	Reverse, defined as $(-1)^{k(k-1)/2}\,\mathbf{A}_{\langle k \rangle}$.
$\left\|\mathbf{A}_{\langle k \rangle}\right\|^2$	Squared norm, defined as $\mathbf{A}_{\langle k \rangle} * \widetilde{\mathbf{A}}_{\langle k \rangle}$.
$\mathbf{A}^{-1}_{\langle k \rangle}$	Inverse of $\mathbf{A}_{\langle k \rangle}$, defined as $\widetilde{\mathbf{A}}_{\langle k \rangle} \,/\, \left\|\mathbf{A}_{\langle k \rangle}\right\|^2$.
$\mathbf{A}^*_{\langle k \rangle}$	Dual of $\mathbf{A}_{\langle k \rangle}$, defined as $\mathbf{A}_{\langle k \rangle} \rfloor \mathbf{I}^{-1}_{\langle n \rangle}$.
$\mathbf{D}^{-*}_{\langle n-k \rangle}$	Undual of $\mathbf{D}_{\langle n-k \rangle}$, defined as $\mathbf{D}_{\langle n-k \rangle} \rfloor \mathbf{I}_{\langle n \rangle}$.
Γ	3rd-Rank tensor encoding the geometric product. The (i,j,k)-th entry is denoted as $\Gamma^{i,j,k}$.
Λ	3rd-Rank tensor encoding the left contraction. The (i,j,k)-th entry is denoted as $\Lambda^{i,j,k}$.
Υ	2nd-Rank tensor encoding the reverse operation. The (i,j)-th entry is denoted as $\Upsilon^{i,j}$.

MATRIX ALGEBRA

a, b, etc.	General column vector. The i-th coefficient is denoted as α^i, β^i, etc.
A, B, etc.	General matrix. The (i,j)-th entry is denoted as $\alpha^{i,j}$, $\beta^{i,j}$, etc.
$\mathrm{A}\,\mathrm{B}$	Matrix multiplication of A and B.
A^{T}	Transpose of A.
A^{-1}	Inverse of A.

STATISTICS

$\underline{x}$, $\underline{y}$, etc.	Random variable (Gaussian distribution is assumed).
$\overline{x}$, $\overline{y}$, etc.	Expectation (mean value) of $\underline{x}$, $\underline{y}$, etc.
σ_x, σ_y, etc.	Standard deviation of $\underline{x}$, $\underline{y}$, etc.
σ_x^2, σ_y^2, etc.	Variance of $\underline{x}$, $\underline{y}$, etc.
$\underline{\mathrm{x}}$, $\underline{\mathrm{y}}$, etc.	Vector-valued random variable (Gaussian distribution is assumed).
$\overline{\mathrm{x}}$, $\overline{\mathrm{y}}$, etc.	Expectation (mean vector) of $\underline{\mathrm{x}}$, $\underline{\mathrm{y}}$, etc.
Σ_{x}, Σ_{y}, etc.	Covariance matrix of $\underline{\mathrm{x}}$, $\underline{\mathrm{y}}$, etc.
α	Significance level.
e	Error, defined as $(x - x_{\text{true}})$, where x is some experimental value with error and x_{true} is the true value.
Φ	Cumulative distribution function of the standard normal distribution.

SET THEORY

$\mathcal{A}$, $\mathcal{B}$, etc.	Set of elements.
$\varnothing$	Empty set.
$\lvert\mathcal{A}\rvert$	Cardinality of $\mathcal{A}$.
$\mathcal{A} \cup \mathcal{B}$	Union of $\mathcal{A}$ and $\mathcal{B}$.
$\mathcal{A} \backslash \mathcal{B}$	Relative complement of $\mathcal{A}$ in $\mathcal{B}$.

MISCELLANEOUS

$\binom{n}{k}$	Binomial coefficient, defined as $n!/\left(k!\,\left(n-k\right)!\right)$, for $0 \leq k \leq n$.
$n!$	The factorial of a positive integer n, defined as $\prod_{i=1}^{n} i$.
$\max(\alpha, \beta)$	The larger of the two values.

CONVENTIONS OF THE PROPOSED APPROACH

n	The number of dimensions of the representational space.
p	The number of dimensions of the intended subspace, for $0 \leq r \leq n$.
r	The number of dimensions of an input subspace from the dataset, for $0 \leq r \leq n$.
m	The number of dimensions of the parameter space, defined as $p\,(n-p)$.
$\theta^{v,j}$, θ^{t}	One of the m parameters characterizing a given p-blade. It is defined in the $[-\pi/2, \pi/2)$ range. The use of double or single-index notation depends of the context.
Δ_θ	The linear step assumed for the discretization of the parameter space $\mathbb{P}^m$.
$\mathbf{X}_{\langle r \rangle}$	An input subspace from the dataset.
$\mathbf{C}_{\langle p \rangle}$	A p-dimensional subspace related to a given input entry $\mathbf{X}_{\langle r \rangle}$.
$\mathbf{E}_{\langle p \rangle}$	The reference blade for p-dimensional subspaces (Equation 4.5).
$\mathbf{P}^{(t)}_{\langle 2 \rangle}$	The plane where the t-th rotation applied on $\mathbf{E}_{\langle p \rangle}$ happens.
$\mathbf{F}^{(t)}_{l}$	A space of possibilities (Equation 5.5).
$\boldsymbol{R}_{v,j}$, $\boldsymbol{R}_{t}$	A rotor encoding a rotation of $\theta^{v,j}$ (or θ^{t}) radians on plane $\mathbf{e}_{j+1} \wedge \mathbf{e}_{j}$ (Equation 4.3). The use of double or single-index notation depends of the context.
$\boldsymbol{S}_{v}$	A rotor computed as the geometric product of a sequence of rotors $\boldsymbol{R}_{v,j}$ (Equation 4.2).
$\boldsymbol{T}$	A rotor computed as the geometric product of a sequence of rotors $\boldsymbol{S}_{v}$ (Equation 4.8).

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This dissertation presents a generalized closed-form framework for detecting data alignments in large unordered noisy multidimensional datasets. In this approach, the intended type of data alignment may be a geometric shape (*e.g.,* straight line, plane, circle, sphere, conic section, among others) or any other structure, with arbitrary dimensionality that can be characterized by a linear subspace. The detection is performed using a three-step process. In the initialization, a $p\,(n-p)$-dimensional parameter space is defined in such a way that each point in this space represents an instance of the intended alignment described by a p-dimensional subspace in some n-dimensional domain. In turn, an accumulator array is created as the discrete representation of the parameter space. In the second step each input entry (also a subspace in the n-dimensional domain) is mapped to the parameter space as the set of points representing the intended p-dimensional subspaces that contain or are contained by the entry. As the input entries are mapped, the bins of the accumulator related to such a mapping are incremented by the importance value of the entry. The subsequent and final step retrieves the p-dimensional subspaces that best fit input data as the local maxima in the accumulator array.

The proposed parameterization is independent of the geometric properties of the alignments to be detected. Also, the mapping procedure is independent of the type of input data and automatically adapts to entries of arbitrary dimensionality. This allows application of the proposed approach (without changes) in a broad range of applications as a pattern detection tool. Given its general nature, optimizations developed for the proposed framework immediately benefit all the detection cases.

I demonstrate a software implementation of the proposed technique and show that it can be applied in simple detection cases as well as in concurrent detection of multiple kinds of alignments with different geometric interpretations, in datasets containing multiple types of data. This dissertation also presents an extension of the general detection scheme to data with Gaussian-distributed uncertainty. The proposed extension produces smoother distributions of values in the accumulator array and makes the framework more robust to the detection of spurious subspaces.

Keywords: Subspace detection, pattern recognition, subspace parameterization, parameter space, generalization, error propagation, Grassmannian, Hough transform, geometric algebra, Clifford algebra.

Sobre a Generalização da Detecção de Subespaços em Dados Multidimensionais não Ordenados

RESUMO

Este trabalho apresenta uma solução geral para a detecção de alinhamentos de dados em conjuntos multidimensionais não ordenados e ruidosos. Nesta abordagem, o tipo requerido de alinhamento de dados pode ser uma forma geométrica (*e.g.,* linha reta, plano, círculo, esfera, seção cônica, entre outras) ou qualquer estrutura, com dimensionalidade arbitrária, que possa ser caracterizada por um subespaço linear. A detecção é realizada por meio de um procedimento composto por três etapas. Na etapa de inicialização, um espaço de parâmetros com $p\,(n-p)$ dimensões é definido de modo que cada ponto neste espaço represente uma instância do alinhamento requerido, descrito por um subespaço p-dimensional em um domínio n-dimensional. Em seguida, uma grade de acumuladores é criada como sendo a representação discreta do espaço de parâmetros. Na segunda etapa do procedimento, cada elemento no conjunto de dados de entrada (também um subespaço no domínio n-dimensional) é mapeado para o espaço de parâmetros como os pontos (no espaço de parâmetros) representando os subespaços requeridos que contém ou que estão contidos no elemento de entrada. À medida que os elementos de entrada são mapeados, as células do acumulador relacionadas com o mapeamento são incrementadas pelo valor de importância do elemento mapeado. A etapa final do procedimento recupera os subespaços p-dimensionais que melhor se ajustam aos dados de entrada como sendo os máximos locais na grade de acumuladores.

A parametrização proposta é independente das propriedades geométricas dos alinhamentos a serem detectados. Além disso, o procedimento de mapeamento é independente do tipo de dado de entrada e é capaz de se adaptar a elementos com dimensionalidades arbitrárias. Essas características permitem a utilização da técnica (sem a necessidade de modificações) como uma ferramenta para a detecção de padrões em uma grande quantidade de aplicações. Por conta de sua natureza geral, otimizações desenvolvidas para a abordagem proposta beneficiam, de forma imediata, todos os casos de detecção.

Neste trabalho eu demonstro uma implementação em software da técnica proposta e mostro que ela pode ser aplicada tanto em casos simples de detecção, quanto na detecção concorrente de tipos diferentes de alinhamentos, com diferentes interpretações geométricas e em conjuntos de dados compostos por vários tipos de elementos. Esta dissertação também apresenta uma extensão do esquema de detecção para dados de entrada com distribuição Gaussiana de incerteza. A extensão proposta produz distribuições de valores mais suaves na grade de acumuladores e faz com que a técnica fique mais robusta à detecção de subespaços espúrios.

Palavras-chave: Detecção de subespaços, reconhecimento de padrões, parametrização de subespaços, espaço de parâmetros, generalização, propagação de erro, Grassmaniana, transformada de Hough, álgebra geométrica, álgebra de Clifford .

1 INTRODUCTION

Data analysis is a fundamental element in scientific discovery and data mining. In many scientific fields, visual inspection of experimental datasets is often performed in order to identify strong local coherence in the data. Such coherence results from data alignments (in some multidimensional space), and usually emerges as geometric shapes and patterns. For instance, straight lines and circles appear as well-defined structures in the analysis of electron backscatter diffraction images (Figure 1.1a) and clonogenic essays (Figure 1.1b), respectively. However, when large volumes of data need to be analyzed, visual inspection becomes impractical. For this reason, automatic detectors for specific types of data alignments have been broadly applied by scientists in many different areas, such as particle physics (MANKEL, 2004; FECHNER et al., 2009), astronomy (KRISHNAN et al., 2004; ABBOTT et al., 2008), microbiology (BEWES; SUCHOWERSKA; MCKENZIE, 2008; KüRNER; FRANGAKIS; BAUMEISTER, 2005), crystallography (NAUMOVIć et al., 2001; LIU; RAABE; ZAEFFERER, 2008), and medicine (KOBATAKE; YOSHINAGA, 1996; DING; FENSTER, 2003). In data mining for e-commerce, large datasets have been analyzed in search for patterns of customer behaviors and preferences (BöHM et al., 2004). Automatic detectors are also a central component of many computer vision (HARTLEY; ZISSERMAN, 2004) and image processing (GONZALEZ; WOODS, 2008) applications. The goal of automatic detectors is to identify certain kinds of alignments that best fit a given unordered dataset, even in presence of noise and discontinuities.

This thesis introduces an approach for detecting data alignments in unordered noisy multidimensional data. The proposed approach is based on the observation that a wide class of alignments (*e.g.,* straight lines, planes, circles, spheres, conic sections, among others) can be represented as linear subspaces. Thus, instead of defining a different detector for each specific case and input data type, it is possible to design a unifying framework to detect the occurrences of emerging subspaces in multidimensional datasets. The proposed approach presents the following properties:

1. It has a closed-form solution for the detection of subspaces of a given dimensionality on datasets that may be heterogeneous and contain elements (*i.e.,* subspaces) with arbitrary dimensionalities;
2. It can be applied, without changes, to the detection of any class of data alignments that can be characterized by a linear subspace;
3. It is independent of the dimensionality and the metric of the space where data resides; and
4. It guarantees the use of the smallest set of parameters in the representation of the intended subspaces. Thus, when applied as a geometric shape detector, the proposed

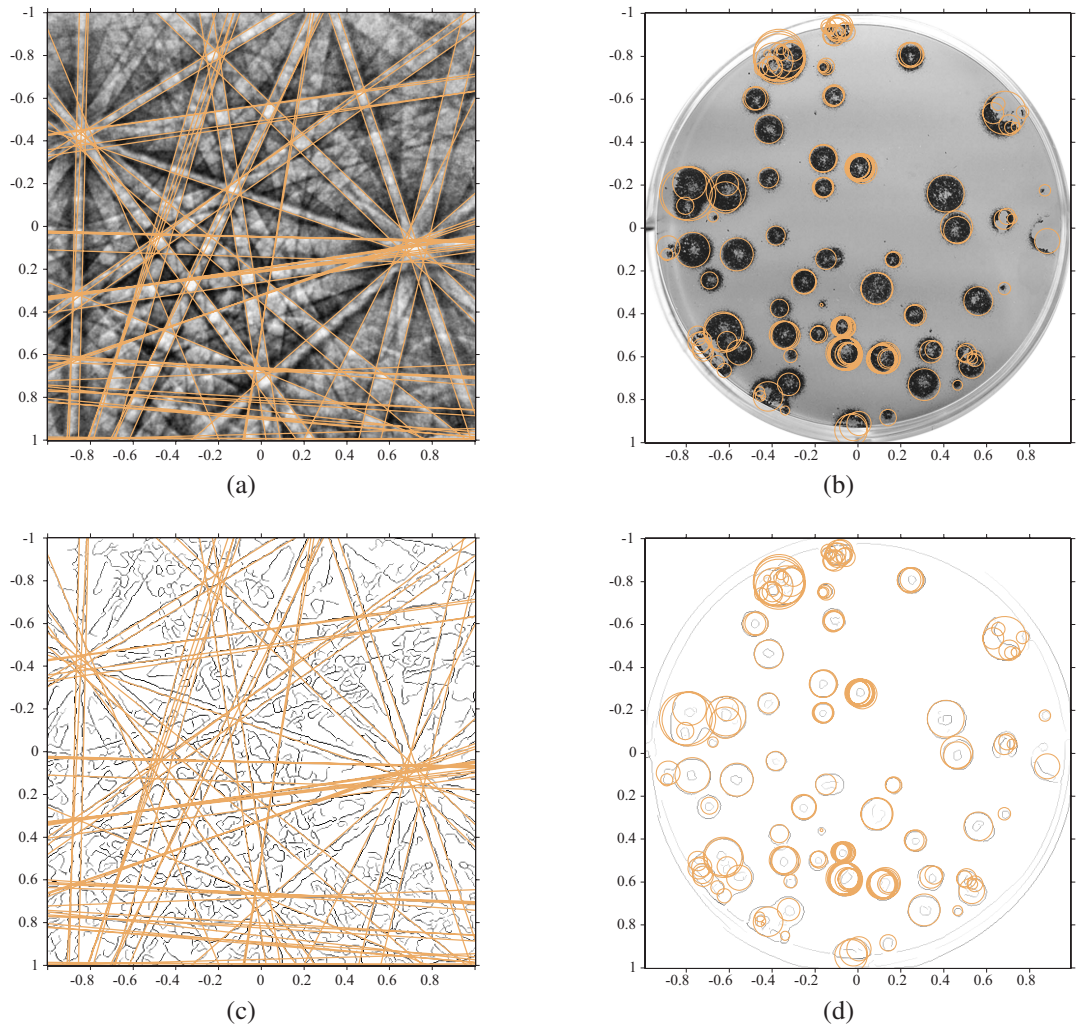


Figure 1.1: Automatic detection of data alignments on real datasets. (a) Electron backscatter diffraction image (445×445 pixels) taken from a particle of wulfenite ($PbMoO_4$). The detection of straight lines is key for the identification of the particle's crystalline phase. (b) Gray image (529×534 pixels) of infection with A/Memphis/14/96-M (H1N1) in MDCK-SIAT1 cells. The detection of circles is important for automated counting process in clonogenic assays. The proposed approach was used, without any changes, to automatically detect the straight lines and circles shown in (a) and (b) from the edge information shown in (c) and (d), respectively.

approach always represents the intended shapes in the most compact way.

I have formulated the subspace detector using Geometric Algebra (GA). GA is a powerful mathematical system encompassing many mathematical concepts (*e.g.,* complex numbers, quaternions algebra, Grassmann-Cayley algebra, and Plücker coordinates) under the same framework (DORST; FONTIJINE; MANN, 2007; PERWASS, 2009). GA is mainly based on the algebraic system called Clifford Algebra (LOUNESTO, 2001), but with a strong emphasis on geometric interpretation. In GA, subspaces are treated as primitives for computation. As such, it is an appropriate mathematical tool for modeling the subspace-detection problem. Also, GA has been proven to be capable of representing many types of geometry. Examples of models of geometry (MOGs) successfully

encoded by GA include Euclidean, Projective (HESTENES; ZIEGLER, 1991), Spherical (HESTENES, 1987, 2001), Hyperbolic (LI, 1997), and Conic (PERWASS; FÖRSTNER, 2006) spaces. These MOGs provide practical applications for the proposed technique as a detector of emerging geometric shapes on datasets like, but not limited to, images, volumetric datasets, and point clouds.

1.1 Conventional Approaches

Most of the techniques for detecting data alignments are derived from the Hough Transform (HT) (HOUGH, 1959, 1962), Random Sample Consensus (RANSAC) (FISCHLER; BOLLES, 1981), or Tensor Voting (TV) (MEDIONI; LEE; TANG, 2000) paradigms. A discussion on these paradigms and some of their variations is presented in Chapter 2.

The HT performs the detection of an intended type of shape by mapping each input entry from the dataset to a set of points (in a parameter space) representing the shapes potentially passing through that entry. Based on this simple mechanism, the shape detection problem can be converted into the simpler problem of identifying peaks in an accumulator array representing the discretized parameter space.

The RANSAC is a non-deterministic technique where the most significant instance of a given type of data alignment is identified by iteratively setting candidate instances from randomly sampled input data entries. After a given number of trials, the candidate instance which approximates the most entries in the dataset is reported as detected.

The TV paradigm retrieves the most salient surfaces, curves and junctions in a dataset by encoding the geometric information associated with each input entry into tensor form. Then, the encoded information is propagated throughout a neighborhood via tensor voting fields. After two rounds of voting and refinement, the salient features (with any dimensionality) are identified as peaks of votes in dense tensor voting maps computed (at the same time) for each type of feature.

In order to use the HT or RANSAC approaches, one needs to assume a mathematical model for the intended type of data alignment with respect to the expected type of input data. Thus, traditionally, the variations of the HT and RANSAC have been designed for detecting specific types of structures in a given type of data. Such a specialization prevents the development of generally applicable techniques and optimizations due to specificities in their formulations. Although there are partial generalizations of the HT and RANSAC designed to some classes of analytic shapes (BALLARD, 1981; SCHNABEL; WAHL; KLEIN, 2007; ACHTERT et al., 2008) and HTs for non-analytic shapes (BALLARD, 1981; WANG; REEVES, 1990), such approaches are still restrictive regarding the assumed input or intended output data.

The TV, on the other hand, follows a generalized definition. It is designed to detect alignments with arbitrary dimensionality in heterogeneous datasets. The TV, however, returns all possible features at the same time. Such a behavior prevents the efficient detection of predefined types of alignments, because it requires a subsequent filtering step.

1.2 Thesis Statement

I propose a generalization of the voting-based approaches for detecting data alignments that can be characterized by subspaces in any MOG. The central thesis statement of this research is presented as follows:

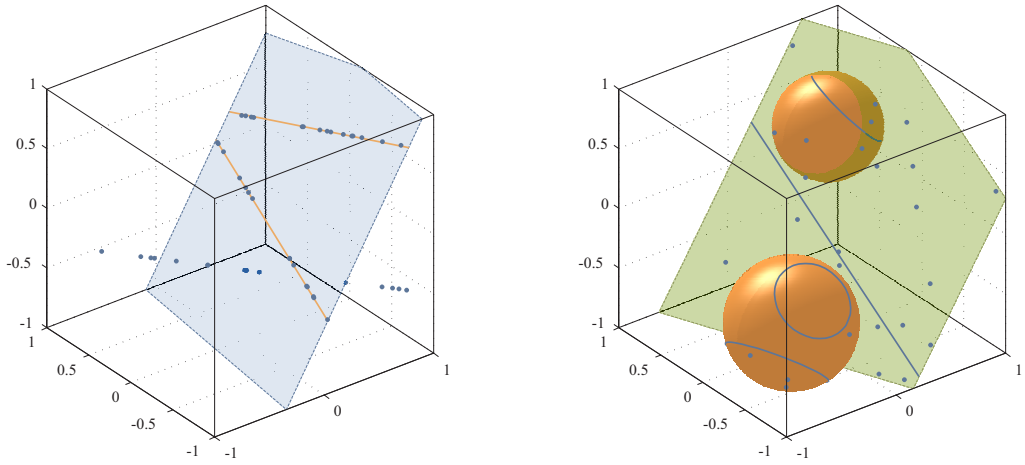


Figure 1.2: Automatic detection of data alignments on heterogeneous synthetic datasets. (a) Detection of lines that best fit an input set comprised by 45 points and 1 plane. In this example, we are concerned with the detection of lines on the input plane that best fit subsets of input points. (b) Concurrent detection of plane and spheres by a single application of the proposed approach. The input dataset is comprised by 43 points, 1 straight line, and 3 circles.

> *It is possible to define a parameterization for linear subspaces which is independent of the dimensionality of the intended subspace, the input data type, and the metric of the whole space. Moreover, since linear subspaces of a given dimensionality can be interpreted as some type of data alignments by assuming a model of geometry, it is possible to use such a parameterization in the development of a generalized voting-based approach for the automatic detection of data alignments in heterogeneous datasets.*

Three central issues must be addressed in order to explore the domain associated with the proposed detection framework. The first one is to define a parameterization for linear subspaces of a given dimensionality embedded in some n-dimensional space. Chapter 4 presents a derivation where p-dimensional subspaces, for $0 \leq p \leq n$, are parameterized by $p\,(n-p)$ rotation operations applied to a canonical subspace with the same dimensionality. The parameterization is defined in such a way that it is independent of the interpretation of the subspace in the actual context (*i.e.,* the geometric interpretation in some MOG).

The second problem is to define a duality relationship between input data and the parameters characterizing the intended subspaces. This relationship is used to derive a procedure for mapping input entries with arbitrary dimensionality to the parameter space. Such a mapping is the central component of a voting scheme, where a discrete representation of the parameter space is used in order to identify which are the p-dimensional subspaces that best fit input data. This subject is explored in Chapter 5, where general mapping and voting procedures for input r-dimensional subspaces ($0 \leq r \leq n$) are defined in such a way that no prior knowledge on the geometric interpretation of input data is taken into account. These procedures are further extended in Chapter 6 by allowing input datasets to contain data with Gaussian distributed uncertainty. In such a case, the uncertainty intrinsic to experimental (real) data is taken into account while spreading

votes in the discrete map.

The third issue is how to retrieve the most relevant subspaces from the discrete representation of the parameter space after the voting procedure has been performed. This topic is discussed in Chapter 7, where a sweep-hyperplane approach for performing peak-detection in a multidimensional histogram is proposed.

1.3 Overview of the General Subspace Detection Framework

A p-dimensional subspace $\mathbf{B}_{\langle p \rangle}$ in a n-dimensional space can be characterized by a set of $m = p\,(n - p)$ rotations applied to a canonical subspace ($\mathbf{E}_{\langle p \rangle}$) used as reference. More precisely, in GA notation:

$$\mathbf{B}_{\langle p \rangle} = \boldsymbol{T}\,\mathbf{E}_{\langle p \rangle} \,/\, \boldsymbol{T}, \tag{1.1}$$

where $\boldsymbol{T}$ is the rotor (rotors are defined in Chapter 3) encoding the sequence of m rotation operations. The computation of $\boldsymbol{T}$ is presented in Chapter 4. In Equation 1.1, the transformation encoded by $\boldsymbol{T}$ is applied on $\mathbf{E}_{\langle p \rangle}$ in order to obtain $\mathbf{B}_{\langle p \rangle}$. The m rotation angles are the parameters characterizing $\mathbf{B}_{\langle p \rangle}$, and the values of p and n depend on the intended data alignment. For instance, by assuming the homogeneous (or projective) MOG for line detection in images (Figure 1.1a), $n = 2 + 1 = 3$ and $p = 2$, leading to $m = 2\,(3 - 2) = 2$.

Table 1.1: The steps of the proposed subspace detection framework. It takes as input a set $\mathcal{X}$ of subspaces (*i.e.,* the input dataset encoded into a MOG), the dimensionality p of subspaces interpreted as the intended data alignment (see Table 3.1 for a few examples) in the same MOG, and the dimensionality n of the whole vector space. The algorithm outputs the p-dimensional subspaces that best fit the input set $\mathcal{X}$.

Step	**Description**
1. Initialization	Setup the model function for p-dimensional subspaces (Equation 1.1), and define the parameter space $\mathbb{P}^m$ (Equation 1.2) from it. Then, create (and initialize with zeros) an m-dimensional accumulator array as the discrete representation of $\mathbb{P}^m$.
2. Voting Procedure	Map each input subspace $\mathbf{X}_{\langle r \rangle} \in \mathcal{X}$ to $\mathbb{P}^m$ using the procedure described in Chapter 5 or in Chapter 6. During the mapping, some parameters receive a single value, while others assume all values in $[-\pi/2, \pi/2)$. However, given the discrete nature of the accumulator array, these parameters only need to assume a set of discrete values in the $[-\pi/2, \pi/2)$ range. As a result, the mapping procedure retrieves a finite set of points in $\mathbb{P}^m$. These points are used to address bins in the accumulator array. Voting is performed by incrementing such bins by the importance ω of $\mathbf{X}_{\langle r \rangle}$.
3. Peak Detection	Identify the bins that correspond to local maxima in the accumulator array. Use the coordinates of such bins (*i.e.,* parameter vectors) to retrieve the most significant p-dimensional subspaces.

The m rotation angles related to the sequence of rotation operations in Equation 1.1 define a parameter space for p-dimensional subspaces. The proposed subspace detector uses such parameter space. The application of the approach consists of three steps:

1. Create an accumulator array as a discrete representation of the parameter space;
2. Perform a voting procedure where the input dataset is mapped to the accumulator array; and
3. Search for the peaks of votes in the accumulator, as they correspond to the intended p-dimensional subspaces that best fit the input dataset.

Table 1.1 summarizes these steps. Step (1) defines a parameter space ($\mathbb{P}^m$) for the m degrees of freedom:

$$\mathbb{P}^m = \{(\theta^1, \theta^2, \cdots, \theta^m) \mid \theta^t \in [-\pi/2,\pi/2)\}, \tag{1.2}$$

where each parameter vector $(\theta^1, \theta^2, \cdots, \theta^m) \in \mathbb{P}^m$ characterizes an instance of a p-dimensional subspace in some n-dimensional vector space. The values of p and n are related to the MOG where data has been encoded and the type of data alignment (represented in such a MOG) one wants to detect. In practice, one needs to discretize $\mathbb{P}^m$, for which an accumulator array is built to receive "votes" and initialized with zeros.

Step (2) maps the input dataset to parameter space. Essentially, the mapping procedure takes each r-dimensional subspace $\mathbf{X}_{\langle r \rangle}$ in the input dataset and identifies the parameters (coordinates in $\mathbb{P}^m$) of all p-dimensional subspaces related to it. When $r \leq p$, the mapping procedure identifies in $\mathbb{P}^m$ all p-dimensional subspaces containing $\mathbf{X}_{\langle r \rangle}$ (*e.g.,* the lines containing input points in Figure 1.2a). If $r \geq p$, the procedure identifies in $\mathbb{P}^m$ all p-dimensional subspaces contained in $\mathbf{X}_{\langle r \rangle}$ (*e.g.,* the lines on the input plane in Figure 1.2a).

After the voting procedure has been performed for all $\mathbf{X}_{\langle r \rangle}$, the number of votes deposited in each accumulator bin defines the importance of that bin to the input subspaces. Thus, the most voted bins represent the detected p-dimensional subspaces. The final step of the approach searches for local maxima in the accumulator array. The parameter vectors associated with such bins are used in Equation 1.1 to retrieve the detected subspaces. This is achieved by applying the sequence of rotations specified by these bins $(\theta^1, \theta^2, \cdots, \theta^m)$ to the canonical subspace $\mathbf{E}_{\langle p \rangle}$.

1.4 Results

This dissertation presents some original results that include:

- A general framework for subspace detection in unordered multidimensional datasets;
- A parameterization scheme for subspaces based on the rotation of a canonical subspace with the same dimensionality;
- An algorithm that enumerates all instances of subspaces with a given dimensionality p that either contain or are contained by an input subspace of arbitrary dimensionality;
- A procedure that maps subspaces with Gaussian distributed uncertainty to the parameter space characterizing p-dimensional subspaces;
- A number of experimental evidences supporting that the open affine covering of the Grassmannian (HARRIS, 1992) can be used as an auxiliary space where the

uncertainty of geometric structures like straight line, plane, circle, sphere and conic section, among others, can be handled in an unified fashion; and

- An algorithm that identifies local maxima in a multidimensional histogram.

In addition to these original results, the following assertions about the proposed framework will be demonstrated:

- It is the generalization of the HTs for analytical shapes representable by linear subspaces;
- It allows the detection of subspaces that best fit an input set of subspaces with different dimensionalities and different geometric interpretations (*e.g.,* the detection of straight lines that best fit points and planes – Figure 1.2a);
- It allows the concurrent detection of subspaces with different geometric interpretations (*e.g.,* planes and spheres – Figure 1.2b) but with the same dimensionality in a given MOG;
- The intended p-dimensional subspaces are represented with the smallest possible number of parameters, leading to the most compact parameterization of analytical shapes (*e.g.,* straight lines, circles, and general conic sections in the plane are parameterized with two, three, and four parameters, respectively);
- The detection can be driven to a specific type of data alignment just by changing the assumed MOG where data have been encoded, while the formulation of the detection scheme remains unchanged; and
- An approximation of the dth-order Voronoi diagram (VORONOI, 1908) of a set of points in $\mathbb{R}^d$ can be retrieved as byproduct of the detection of subspaces geometrically interpreted as circles, spheres, and their higher-dimensional counterparts.

It is important to emphasize that the proposed approach is not restricted to the detection of geometric shapes. It can be applied to any domain in which a problem can be cast as a subspace detection one. For example, the subspace clustering problem in data mining applications, where the goal is to find among all possible subspaces those that accommodate as many database objects as possible. Also, as a closed-form solution, the same implementation of the proposed framework and its optimizations naturally generalize to all detection cases.

1.5 Demonstration and Validation of the Techniques

The technique proposed in this thesis is demonstrated by proof of concept implementations of the described algorithms. The implementations are validated by applying the subspace detection framework to real (Figure 1.1) and synthetic (Figure 1.2) datasets. The proposed framework has a broad range of applications as a pattern detection tool. For the purpose of illustration, however, the examples shown in this thesis are restricted to the important problem of detecting analytic geometric shapes in spaces of arbitrary dimensionalities.

2 STATE OF THE ART

This chapter discusses techniques for detecting data alignments, geometric shapes and patterns that best fit a given unordered datasets. The discussion is focused on the generality of the approaches. Despite many detecting techniques have been proposed in the last decades, most of them are derived from one of three main paradigms:

i. Hough Transform (HT).
ii. Random Sample Consensus (RANSAC);
iii. Tensor Voting (TV); and

In Section 2.1 an in-depth treatment to HT is given because the generalized subspace detection scheme proposed in this thesis is closely related to the HT. Sections 2.2 and 2.3 present a brief description of RANSAC and TV, respectively.

2.1 Hough Transforms

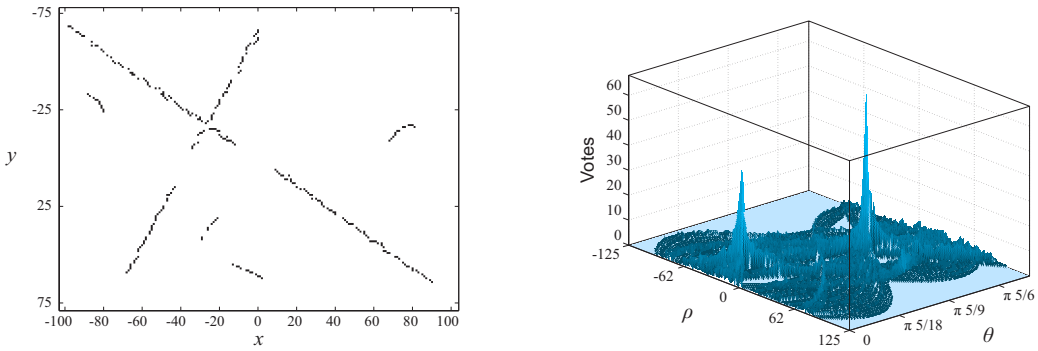
The HT is a classical paradigm for detecting instances of geometric shapes and patterns in an unordered dataset. Initially proposed to detect straight lines in binary images (HOUGH, 1959, 1962; DUDA; HART, 1972), the HT was specialized to identify other types of shapes, such as circles (DUDA; HART, 1972; KIMME; BALLARD; SKLANSKY, 1975), parabolas (SKLANSKY, 1978), and ellipses (BENNETT; BURRIDGE; SAITO, 1999); and generalized for the detection of non-analytical shapes in images (BALLARD, 1981).

A HT can be defined for any shape that can be represented by a **model function** with the form:

$$f\left(x_1, x_2, \cdots, x_d; p_1, p_2, \cdots, p_m\right) = 0, \tag{2.1}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$ is an entry from the input dataset (usually a point in $\mathbb{R}^d$), and $\boldsymbol{p} = (p_1, p_2, \cdots, p_m) \in \mathbb{P}^m$ is a vector in a **parameter space** $\mathbb{P}^m$. Each parameter vector $\boldsymbol{p}$ in $\mathbb{P}^m$ characterizes an instance of the intended shape. By using a **mapping procedure**, the HT takes each input $\boldsymbol{x}$ and determines all instances of the shape potentially passing through $\boldsymbol{x}$. For each such an instance, its associated parameter vector $\boldsymbol{p}$ is used to address a bin in an **accumulator array** (*i.e.,* a discrete representation of $\mathbb{P}^m$). The value stored in the bin is then incremented by some **importance value** ω (usually *one*). At the end of the process, the bins having the largest values correspond to the parameters of the most likely shapes in the input data.

The mapping function used in the voting procedure is obtained from the model function (Equation 2.1) and consists in arbitrating a subset of k parameter values from $\boldsymbol{p}$ (for $k < m$) and computing the $(m - k)$ non-arbitrated parameter values using a **mapping**



(a) Image space (b) Parameter space

Figure 2.1: The accumulator array produced for a simple example of straight line detection from points. (a) A simple image containing approximately straight line segments. (b) A 3-dimensional visualization of the accumulator array after the voting procedure. The two peaks represent the two main lines in (a).

function with the form:

$$(p_{k+1}, p_{k+2}, \cdots, p_m) = g\left(x_1, \cdots, x_d; p_1, \cdots, p_k\right). \tag{2.2}$$

Equation 2.2 is evaluated for all $(p_1, \cdots, p_k) \in \mathbb{P}^k$. Choosing which parameters are arbitrated and which ones are computed depends on how the model (Equation 2.1) and the mapping (Equation 2.2) functions are defined. For instance, when the normal equation of the line (Equation 2.3) is used as model function in straight line detection from points:

$$x\ \cos\phi + y\ \sin\phi - \rho = 0, \tag{2.3}$$

the mapping function can be written as:

$$\rho = x\ \cos\phi + y\ \sin\phi, \tag{2.4}$$

where $\phi \in [0, \pi)$ is the arbitrated parameter, and ρ is the parameter computed from ϕ and from the (x, y) coordinates of some input point. In Equations 2.3 and 2.4, (ρ, ϕ) is the parameter vector characterizing a straight line. In such a formulation, ρ defines the distance from the origin of the coordinate system of the image to the line, and ϕ is the angle between the x-axis and the normal to the line. Figure 2.1b shows a 3-dimensional visualization of the accumulator array associated with the set of pixels shown in Figure 2.1a. The two peaks represent the two main straight lines approximated by larger pixel segments in the image.

The strong aspects of the HT are its robustness to noise, clutter, and missing data. Its major deficiencies are the computational load associated to its voting scheme, and the large memory required for the accumulator array. Illingworth and Kittler (1988) and Leavers (1993) provide in-depth discussions of many attempts to minimize the drawbacks of the HT.

2.1.1 Specialized Detection Cases

Traditionally, a different variation of the HT has been designed to detect specific types of data alignments or shapes in a given type of input data. That is because one needs to assume a model (Equation 2.1) and define a mapping function (Equation 2.2) for each

combination of input data type and intended structure. For instance, Duda and Hart (1972) propose the use of the normal equation of the line (Equation 2.3) while performing the line detection from points. The circle detection of Duda and Hart (1972), instead, uses the center-radius parameterization as model:

$$(x - x_c)^2 + (y - y_c)^2 - r^2 = 0,$$

where (x_c, y_c, r) is the parameter vector for circles centered at (x_c, y_c) and with radius r; and (x, y) are the coordinates of some input point.

The specialization of the HT prevents the development of optimizations that are generally applicable. Thus, optimizations to the HT need to be done on a case-by-case basis. For instance, O'Gorman and Clowes (1973) pointed out that line detection from feature pixels in images may be optimized by the use of gradient information computed for the pixels. Kimme *et al.* (1975) follow such an approach providing the same level of optimization to the HT for circles in images. Note that the same optimization took almost two years to be extended to a single case of HT. The design of most general variations of the HT may lead to the development of optimizations that immediately benefit all the detection cases.

2.1.2 Previous Attempts to Generalize the Hough Transform

Ballard (1981) presented a conceptual HT algorithm for detecting **analytical curves on the plane**, like straight lines, circles, parabolas, and ellipses. This algorithm does not have a closed formulation for the transform, relying on the derivatives of the curves with respect to a point in order to define the mapping function. As pointed out by Ballard (1981), the algorithm often requires considerable algebraic manipulations to be defined. Also, it restricts the expected input data type.

Recently, Achtert *et al.* (2008) proposed a HT to find **oriented flat spaces** (*e.g.,* straight lines, planes, and their higher-dimensional counterparts) that fit a set of points in $\mathbb{R}^d$. The approach uses spherical coordinates in the parameterization of the intended structures. Thus, it constraints the detected elements to flat subspaces and type of input data to points only. This technique can be seen as the generalization of the straight line detection scheme proposed by Duda and Hart (1972) to higher dimensional flat spaces.

The Generalized Hough Transform (GHT) is a Hough-like method for detecting **shapes (in images) that cannot be represented analytically** (BALLARD, 1981). The intended shape is described by a lookup table that stores a list of gradient directions and boundary positions relative to some reference point on the shape. The method allows the identification of the occurrences of the shape regarding changes in location, orientation, and scaling. The extension of the GHT to 3-dimensional shapes is described by Wang and Reeves (1990). As pointed out by Leavers (1993), the GHT is not suitable for the detection of analytical shapes because it does not offer an efficient representation of all such shapes. For instance, straight lines in the plane can be represented with two parameters, while the GHT uses four parameters for any shape in the plane. In order to achieve an efficient representation of straight lines, the parameterization of the GHT must be explicitly changed.

2.2 Random Sample Consensus

The RANSAC (FISCHLER; BOLLES, 1981) identifies the most significant instance of a given model (*i.e.,* a mathematical representation of the intended shape) in a dataset

by consecutive adjustments of the model on randomly selected minimal sets from input data. A minimal set is the smallest number of entries (usually points) required to uniquely define an instance of the model.

The RANSAC is an iterative and non-deterministic technique. It produces reasonable result only with a certain probability, which increases as more iterations are allowed. At each iteration, a candidate instance is created from a random minimal set of input entries. The candidate is tested against all entries in the data to determine a score value indicating how many of the entries are well approximated by the instantiated shape. After a given number of trials, the candidate instance which approximates the most entries is reported as detected and the algorithm continues on the remaining data.

The technique exhibits the following desirable properties:

- It is conceptually simple, which makes it easily extensible and straightforward to implement;
- It can deal with data containing noise and discontinuities; and
- It allows its application in a wide range of settings by specializing the assumed model, the assumed input data type, the computation of the score, and the criteria for testing how much input entries approximate an instance of the model.

Its major deficiency is the considerable computational load if no further optimizations are applied. Raguram *et al.* (2008) provide a comparative analysis of many optimized RANSAC-based techniques.

It is important to notice that to define a generalized model that allows for the concurrent detection of multiple types of shapes with RANSAC is not a simple task. Usually the concurrent detection is performed by assuming more than one model at the same time, and by using specialized testing criteria for each model; with some heuristic to compare the score of candidate instances having different models. Also, the design of a RANSAC for working with heterogeneous datasets (*i.e.,* comprised by many types of input data) is a challenging task. It implies in defining a different comparative function for each combination of input data type and intended model. The random sampling is also affected because not every combination of heterogeneous data can be used while fitting a model. For instance, in the case of detecting the most significant planes in a heterogeneous set comprised by straight lines and points, two random selected lines may not fit a plane, while three points, or one line and one point, do. One of the most general variations of the RANSAC was proposed by Schnabel *et al.* (2007). It allows the detection of planes, spheres, cylinders, cones, and tori in unorganized point clouds.

2.3 Tensor Voting

The TV framework (MEDIONI; LEE; TANG, 2000) is a unified methodology for the robust inference of local features from noisy data. Such features are retrieved in terms of emerging surfaces, curves, and labeled junctions in a given set of points, points with an associated tangent direction, points with an associated normal direction, or any combination of the above. Initially proposed for working in 3-dimensional spaces (GUY; MEDIONI, 1995), the framework was generalized for the n-dimensional case (TANG; MEDIONI; LEE, 2001).

The detection procedure starts by encoding the geometric information associated with each input entry into tensor form. The technique identifies initial local features by spreading the encoded information within a neighborhood via tensor voting fields. This process

refines the input data by assigning some confidence and surface orientation to it, and by discarding noise. After this initial voting step, each input has its confidence and surface orientation encoded into a generic second order symmetric tensor. The input entries vote a second time to propagate their refined information throughout a neighborhood. The result is a dense tensor map which assigns a measure of confidence and saliency to each point in the domain. In the 3-dimensional case, the dense map is decomposed into three other dense maps, each one representing a geometric feature (*i.e.,* surface, curve, and junction), which are analyzed during feature extraction.

The TV approach exhibits the following desirable properties:

- It requires no initial guess or thresholding, the only free parameter is the size of the neighborhood to be considered while processing the entries in a given dataset;
- It can be applied on the detection of features having arbitrary dimensionality; and
- It can robustly deal with high amounts of noise and discontinuities.

The major deficiency of TV is its considerable computational load.

While TV can compete with RANSAC in terms of robustness against noise, it is, however, inherently model-free. As a result, it cannot be efficiently applied to the detection of predefined types of data alignments. By definition, it retrieves, at the same time, all the salient structures (with any dimensionality) embedded in a dataset. A subsequent filtering step is required in order to perform the detection of some intended type of structure.

2.4 Discussion

The three main approaches for detecting the most likely structures in a given dataset were presented in this chapter, namely RANSAC, TV and HT. These approaches are a central component of many computer vision, image processing, and data mining applications (BROWN, 1984). The overall operation of each one of the techniques was described, as well as their strength and weakness. The HT received special attention because it is closely related to the technique proposed in this thesis.

The goal of this work is the development of general technique for detecting data alignments in unordered multidimensional datasets. A technique is general if it can be applied, without changes, to the detection of any structure in heterogeneous data. The voting mechanism of the HT fits better the goals of this work. The generality of a voting-based technique depends on the representational capacity of the assumed model. Ideally, such a model should systematically adapt itself to the intended structure, providing a representation with the smallest set of parameters. In contrast to other approaches, voting-based techniques may be designed to process the input entries individually. By doing so, the mapping and voting procedures can be formulated to handle arbitrary types of data. RANSAC-based techniques, on the other hand, depend on the random selection of input entries in order to define a candidate instance of a given model. As pointed out in Section 2.2, such a procedure may lead to the selection of incompatible types of entries while setting an instance of the model. Finally, the TV approach is quite general. However, the detection of a pre-defined type of data alignment or geometric shape goes against its model-free nature.

I propose a voting-based framework for detecting the occurrences of emerging linear subspaces (with a given dimensionality) in multidimensional datasets. The proposed approach is based on the representation of subspaces as primitives in GA (Chapter 3). By assuming a MOG, such subspaces can be geometrically interpreted as some shape

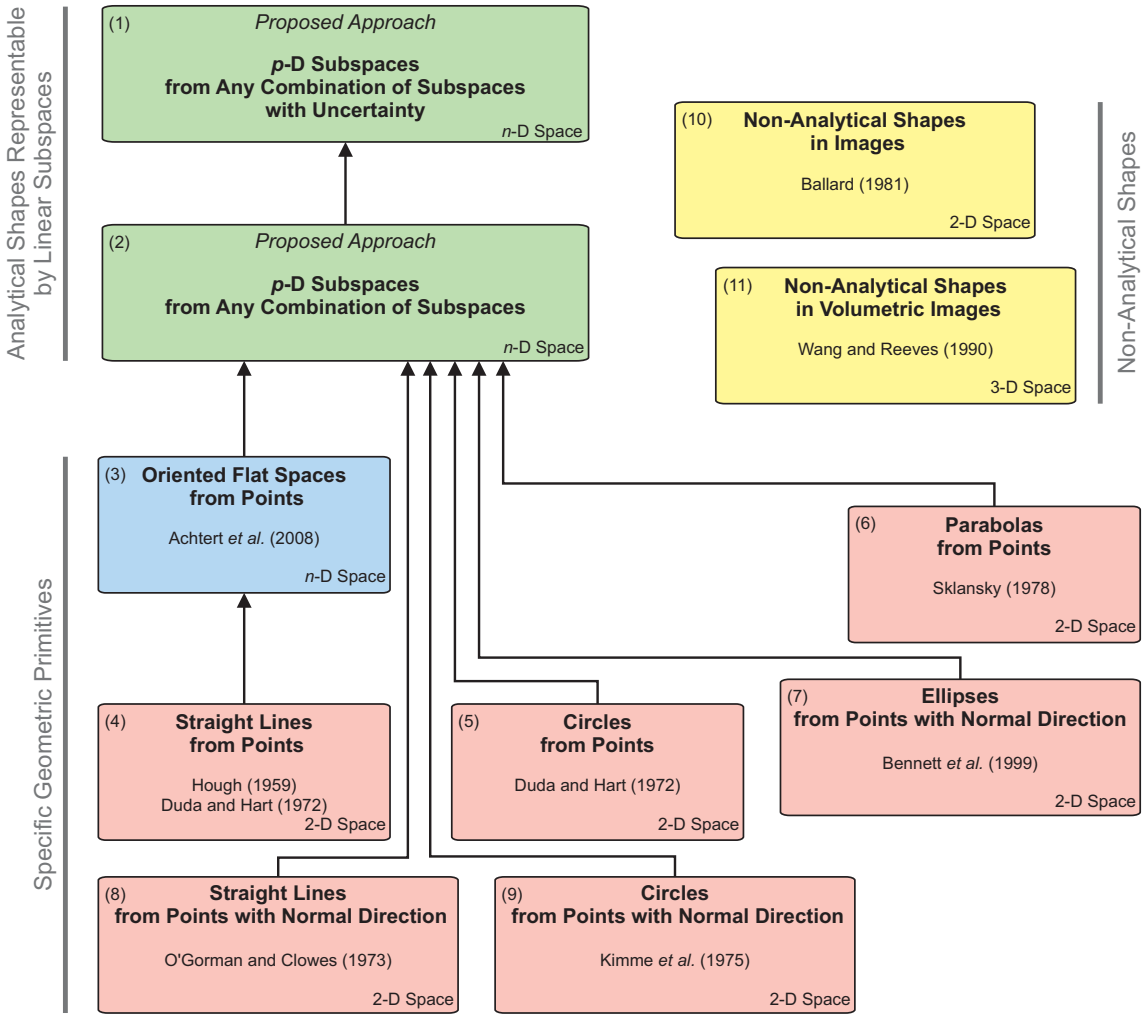


Figure 2.2: Hierarchical representation of the generalization of detection techniques related to the HT. The approaches (1) and (2) are proposed in this thesis. When applied to the detection of geometric shapes they can be seen as the generalization of the HTs for analytic shapes (of arbitrary dimensionality) that can be represented by some linear subspace, *e.g.,* (3-9). It is important to note that the techniques (10) and (11) target a different problem: the detection of shapes (in images) that cannot be represented analytically.

(*e.g.,* straight lines, circles, planes, spheres, among others) or other data alignments (*e.g.,* customer behaviors and preferences in e-commerce databases may emerge as linearly correlated data (BöHM et al., 2004)). Unlike in conventional HTs (HOUGH, 1959, 1962; DUDA; HART, 1972; O'GORMAN; CLOWES, 1973; KIMME; BALLARD; SKLANSKY, 1975; SKLANSKY, 1978; BENNETT; BURRIDGE; SAITO, 1999; BALLARD, 1981; ACHTERT et al., 2008), the parameterization used by the proposed approach is independent of the geometric properties of the structure to be detected. Moreover, the proposed mapping procedures are independent of the input data type. Actually, the framework can be seen as a generalization of the HTs specialized for the detection of structures that can be characterized by linear subspaces. Figure 2.2 shows a hierarchical representation of the generalization of detection techniques related to the HT. The technique depicted as (2) in Figure 2.2 is discussed in Chapters 4, 5 and 7. The technique depicted as (1) extends the former by handling input subspaces with Gaussian distributed uncertainty. It is defined by combining Chapters 4, 6 and 7.

In contrast to Ballard's conceptual algorithm relying on the derivatives of analytical curves (BALLARD, 1981), the proposed approach presents a closed-form solution for the detection of any analytical shape representable by some linear subspace. Also, it is important to note that this thesis targets a different problem than the GHT described by Ballard (1981), which is defined for detecting non-analytical shapes in images. The GHT variations for 2- and 3-dimensional images are depicted in Figure 2.2 as (10) and (11), respectively.

3 GEOMETRIC ALGEBRA

This chapter introduces the concepts of Geometric Algebra (GA) used in this thesis. Sections 3.1 to 3.8 present geometric primitives, transformations, and algebraic operations in a more abstract level. For the sake of simplicity, the mathematical formalisms are kept to a minimum, and the geometrical intuition is illustrated with examples whenever possible. Section 3.9 discusses how one can assume a model of geometry (MOG) in order to give practical geometric interpretation to the elements discussed in previous sections. A more detailed introduction to GA can be found in (FERNANDES; OLIVEIRA, 2009). The books by Dorst *et al.* (2007) and Perwass (2009) provide in-depth treatments to the subject.

3.1 Oriented Subspaces (Blades) as Primitives

By definition, a vector space $\mathbb{R}^n$ consists of a set of elements called vectors, which is closed under finite vector addition and scalar multiplication. By assuming a basis $\{\mathbf{e}_i\}_{i=1}^n$ for $\mathbb{R}^n$, an arbitrary vector (*i.e.,* a 1-dimensional subspace) is written as a linear combination of the basis vectors. Figure 3.1a illustrates a vector

$$\mathbf{a} = \alpha^1 \mathbf{e}_1 + \alpha^2 \mathbf{e}_2 + \alpha^3 \mathbf{e}_3 \in \mathbb{R}^3,$$

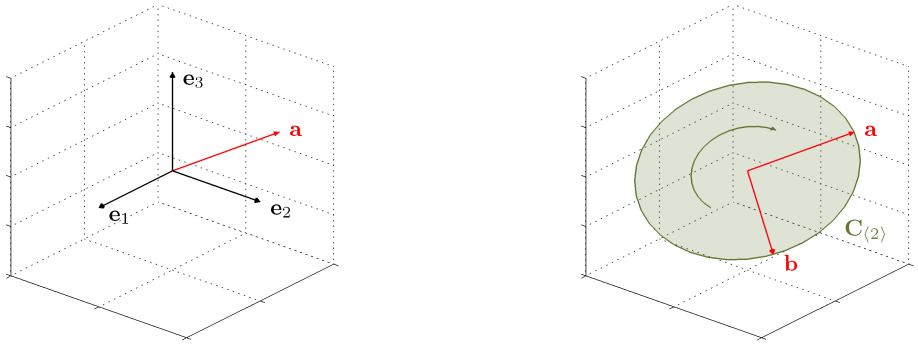
where $\alpha^i \in \mathbb{R}$ is the i-th coefficient of $\mathbf{a}$. In the graphical representation, the 1-dimensional subspace is the oriented straight line that passes through the origin and supports the arrow. The arrow's length represents the weight of the subspace, while the arrow's tip indicates its orientation.

In GA, a 2-dimensional subspace can be spanned as the **outer product** of two linearly independent vectors. The outer product is formally defined in Section 3.3.1. Figure 3.1b shows a 2-dimensional subspace computed as:

$$\mathbf{C}_{\langle 2 \rangle} = \mathbf{a} \wedge \mathbf{b},$$

where $\wedge$ (the *wedge* symbol) denotes the outer product, and $\mathbf{a}$ and $\mathbf{b}$ are two vectors. In Figure 3.1b, $\mathbf{C}_{\langle 2 \rangle}$ is the oriented supporting plane for the disk. The disk's radius and the curved arrow illustrate, respectively, the weight and orientation of the subspace. Note that the orientation of $\mathbf{C}_{\langle 2 \rangle}$ respects the order of the terms in the outer product, *i.e.,* from $\mathbf{a}$ to $\mathbf{b}$.

Using the outer product one can span k-dimensional oriented subspaces, for $0 \leq k \leq n$. In GA, such subspaces are called k-**blades** (the terms blade and subspace are used interchangeably), and k is said to be the **grade** of the blade (the terms grade and dimensionality are also used interchangeably). Thus, for instance, a scalar value $\alpha \in \mathbb{R}$ is a 0-blade, a vector $\mathbf{a} \in \mathbb{R}^n$ is a 1-blade, $\mathbf{C}_{\langle 2 \rangle} = \mathbf{a} \wedge \mathbf{b}$ is a 2-blade, and so on. k-Blades are computational primitives in GA as vectors are primitive elements in vector algebra.



(a) (b)

Figure 3.1: Graphical representation of subspaces in $\mathbb{R}^3$. (a) A 1-dimensional subspace $\mathbf{a}$ is drawn as an arrow, where the arrow's length and the arrow's tip represent, respectively, the weight and the orientation of $\mathbf{a}$. (b) A 2-dimensional subspace $\mathbf{C}_{\langle 2 \rangle}$, spanned from vectors $\mathbf{a}$ and $\mathbf{b}$, is drawn as a disk. Its weight is represented by the radius of the disk, while the orientation is the curved arrow. The basis vectors $\{\mathbf{e}_i\}_{i=1}^3$ are not shown in this image for sake of clarity.

An arbitrary blade $\mathbf{B}_{\langle k \rangle}$ presents the following properties:

attitude The stance of the blade in the surrounding space. The subspaces written in the form $\alpha\,\mathbf{B}_{\langle k \rangle}$, for any $\alpha \in \mathbb{R}$, have the same attitude.

weight The value of α in $\mathbf{B}_{\langle k \rangle} = \alpha\,\mathbf{J}_{\langle k \rangle}$, where $\mathbf{J}_{\langle k \rangle}$ is a reference blade with the same attitude as $\mathbf{B}_{\langle k \rangle}$.

orientation The sign of the weight relative to $\mathbf{J}_{\langle k \rangle}$.

direction The combination of attitude and orientation.

3.2 Multivector Space

In order to treat blades with arbitrary dimensionality as primitives for computation, one needs a basis more robust than the basis of a vector space which, by definition, supports only 1-dimensional subspaces. In GA, $\bigwedge \mathbb{R}^n$ is the **multivector space** built from a vector space $\mathbb{R}^n$. The 2^n basis elements of $\bigwedge \mathbb{R}^n$ are defined as the k-combinations of vectors from the set $\{\mathbf{e}_i\}_{i=1}^n$ (*i.e.,* $\sum_{k=0}^n \binom{n}{k} = 2^n$). As an example, the basis of $\bigwedge \mathbb{R}^3$ is:

$$\Big\{ \underbrace{1,}_{\substack{\text{Scalars} \\ \mathbb{R} = \bigwedge^0 \mathbb{R}^3}} \quad \underbrace{\mathbf{e}_1, \quad \mathbf{e}_2, \quad \mathbf{e}_3,}_{\substack{\text{Vector Space} \\ \mathbb{R}^3 = \bigwedge^1 \mathbb{R}^3}} \quad \underbrace{\mathbf{e}_1 \wedge \mathbf{e}_2, \quad \mathbf{e}_1 \wedge \mathbf{e}_3, \quad \mathbf{e}_2 \wedge \mathbf{e}_3,}_{\substack{\text{Bivector Space} \\ \bigwedge^2 \mathbb{R}^3}} \quad \underbrace{\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3}_{\substack{\text{Trivector Space} \\ \bigwedge^3 \mathbb{R}^3}} \Big\} \tag{3.1}$$

By combining zero elements out of the set $\{\mathbf{e}_i\}_{i=1}^3$, one takes the basis blade "1" for 0-dimensional subspaces, *i.e.,* values in $\mathbb{R}$. As one would expect, the basis blades for 1-dimensional subspaces are the three basis vectors defining $\mathbb{R}^3$. For 2-dimensional subspaces in $\bigwedge^2 \mathbb{R}^3$, the three basis blades are computed by combining two elements out of the set of three basis vectors. Finally, the outer product of all basis vectors define the basis blade for 3-dimensional subspaces. Note that blades in $\bigwedge^3 \mathbb{R}^3$ are a scaled version of

the whole 3-dimensional space. Such blades are often called **pseudoscalar**, and the unit positive pseudoscalar is typically denoted by:

$$\mathbf{I}_{\langle n \rangle} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_n, \tag{3.2}$$

for $\bigwedge \mathbb{R}^n$

A linear combination of the basis elements of a multivector space is called a **multivector**. It is used to encode blades and rotors (rotors are introduced in Section 3.8). The multivector structure for the basis of $\bigwedge \mathbb{R}^3$ (Equation 3.1) is:

$$\begin{aligned} M = \mu^1 1 + \mu^2 \mathbf{e}_1 + \mu^3 \mathbf{e}_2 + \mu^4 \mathbf{e}_3 \\ + \mu^5 \mathbf{e}_1 \wedge \mathbf{e}_2 + \mu^6 \mathbf{e}_1 \wedge \mathbf{e}_3 + \mu^7 \mathbf{e}_2 \wedge \mathbf{e}_3 + \mu^8 \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, \end{aligned}$$

where $\mu^i \in \mathbb{R}$ is the i-th coefficient of M.

Notice that multivectors encode k-blades as the linear combination of the k-dimensional basis elements of the multivector space $\bigwedge \mathbb{R}^n$. However, it is important to notice that not every combination of basis elements in $\bigwedge^k \mathbb{R}^n$ make a k-blade. Only the ones which are factorisable in terms of the outer product do. The simplest example occurs in a multivector space build from a 4-dimensional vector space with basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4\}$. In such a case, the element $(\mathbf{e}_1 \wedge \mathbf{e}_2 + \mathbf{e}_3 \wedge \mathbf{e}_4) \in \bigwedge^2 \mathbb{R}^4$ cannot be written as a 2-blade $\mathbf{C}_{\langle 2 \rangle} = \mathbf{a} \wedge \mathbf{b}$. Such an observation is key in Chapter 4 where it is demonstrated that the proposed parameterization characterizes subspaces by using the smallest set of parameters.

3.3 Some Nonmetric Products

The concept of subspace and its basic properties (Section 3.1) are independent of any metric properties of a vector space $\mathbb{R}^n$, and hence a multivector space $\bigwedge \mathbb{R}^n$ might have (DORST; FONTIJINE; MANN, 2007). GA defines some products which are also independent of the metric of $\bigwedge \mathbb{R}^n$. In this dissertation we are concerned only with the outer product (Section 3.3.1) and the regressive product (Section 3.3.2). In Chapters 4, 5 and 6 these products are used in the definition of the proposed subspace detection scheme.

3.3.1 The Outer Product

Formally, the outer product is a mapping:

$$\wedge : \bigwedge^r \mathbb{R}^n \times \bigwedge^s \mathbb{R}^n \to \bigwedge^{r+s} \mathbb{R}^n.$$

It is defined from a small set of properties:

antisymmetry $\mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a}$, thus $\mathbf{c} \wedge \mathbf{c} = 0$

distributivity $\mathbf{a} \wedge (\mathbf{b} + \mathbf{c}) = \mathbf{a} \wedge \mathbf{b} + \mathbf{a} \wedge \mathbf{c}$

associativity $\mathbf{a} \wedge (\mathbf{b} \wedge \mathbf{c}) = (\mathbf{a} \wedge \mathbf{b}) \wedge \mathbf{c}$

scalars commute $\mathbf{a} \wedge (\beta\, \mathbf{b}) = \beta\, (\mathbf{a} \wedge \mathbf{b})$

The following expression presents a step-by-step algebraic manipulation where the properties of the outer product are used to build a 2-blade $\mathbf{C}_{\langle 2 \rangle} \in \bigwedge^2 \mathbb{R}^3$ from vectors

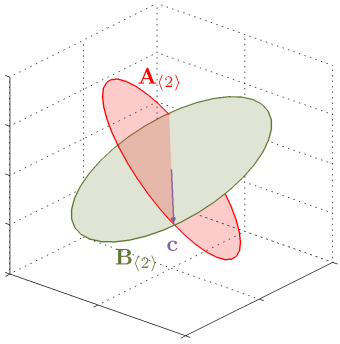


Figure 3.2: The regressive product can be used to build blades as the subspace shared by pseudovectors. For instance, the 1-dimensional subspace $\mathbf{c}$ is computed as the regressive product of $\mathbf{A}_{\langle 2\rangle}$ and $\mathbf{B}_{\langle 2\rangle}$.

$\mathbf{a} = \alpha^1\,\mathbf{e}_1 + \alpha^2\,\mathbf{e}_2 + \alpha^3\,\mathbf{e}_3 \in \mathbb{R}^3$ and $\mathbf{b} = \beta^1\,\mathbf{e}_1 + \beta^2\,\mathbf{e}_2 + \beta^3\,\mathbf{e}_3 \in \mathbb{R}^3$:

$$
\begin{aligned}
\mathbf{C}_{\langle 2\rangle} = \mathbf{a} \wedge \mathbf{b} &= (\alpha^1\,\mathbf{e}_1 + \alpha^2\,\mathbf{e}_2 + \alpha^3\,\mathbf{e}_3) \wedge (\beta^1\,\mathbf{e}_1 + \beta^2\,\mathbf{e}_2 + \beta^3\,\mathbf{e}_3) \\
&= \alpha^1\,\beta^1\,\mathbf{e}_1 \wedge \mathbf{e}_1 + \alpha^1\,\beta^2\,\mathbf{e}_1 \wedge \mathbf{e}_2 + \alpha^1\,\beta^3\,\mathbf{e}_1 \wedge \mathbf{e}_3 && \text{(by distributivity)} \\
&\quad + \alpha^2\,\beta^1\,\mathbf{e}_2 \wedge \mathbf{e}_1 + \alpha^2\,\beta^2\,\mathbf{e}_2 \wedge \mathbf{e}_2 + \alpha^2\,\beta^3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \\
&\quad + \alpha^3\,\beta^1\,\mathbf{e}_3 \wedge \mathbf{e}_1 + \alpha^3\,\beta^2\,\mathbf{e}_3 \wedge \mathbf{e}_2 + \alpha^3\,\beta^3\,\mathbf{e}_3 \wedge \mathbf{e}_3 \\
&= \alpha^1\,\beta^2\,\mathbf{e}_1 \wedge \mathbf{e}_2 + \alpha^1\,\beta^3\,\mathbf{e}_1 \wedge \mathbf{e}_3 && \text{(by antisymmetry)} \\
&\quad - \alpha^2\,\beta^1\,\mathbf{e}_1 \wedge \mathbf{e}_2 + \alpha^2\,\beta^3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \\
&\quad - \alpha^3\,\beta^1\,\mathbf{e}_1 \wedge \mathbf{e}_3 - \alpha^3\,\beta^2\,\mathbf{e}_2 \wedge \mathbf{e}_3 \\
&= (\alpha^1\,\beta^2 - \alpha^2\,\beta^1)\,\mathbf{e}_1 \wedge \mathbf{e}_2 && \text{(by grouping terms)} \\
&\quad + (\alpha^1\,\beta^3 - \alpha^3\,\beta^1)\,\mathbf{e}_1 \wedge \mathbf{e}_3 \\
&\quad + (\alpha^2\,\beta^3 - \alpha^3\,\beta^2)\,\mathbf{e}_2 \wedge \mathbf{e}_3.
\end{aligned}
$$

This example is equivalent to the one depicted in Figure 3.1b. It is important to note that the same set of properties are used while spanning subspaces from two blades having arbitrary grades in $\bigwedge \mathbb{R}^n$.

3.3.2 The Regressive Product

The **regressive product** can be regarded as the correct dual operation to the outer product. While the outer product returns the subspace spanned by linearly independent blades, the regressive product returns the subspace shared by linearly dependent blades that span the whole space.

By using the regressive product one can build k-dimensional oriented subspaces from $(n-k)$ **pseudovectors** (*i.e.,* $(n-1)$-blades in $\bigwedge \mathbb{R}^n$). Figure 3.2 shows a vector $\mathbf{c}$ (*i.e.,* a 1-dimensional subspace) in $\mathbb{R}^3$ computed as:

$$\mathbf{c} = \mathbf{A}_{\langle 2\rangle} \vee \mathbf{B}_{\langle 2\rangle},$$

where $\vee$ (the *vee* symbol) denotes the regressive product, and $\mathbf{A}_{\langle 2\rangle}$ and $\mathbf{B}_{\langle 2\rangle}$ are pseudovectors.

The regressive product is a mapping:

$$\vee : \bigwedge^{n-r} \mathbb{R}^n \times \bigwedge^{n-s} \mathbb{R}^n \to \bigwedge^{n-(r+s)} \mathbb{R}^n.$$

Its properties are similar to the ones from the outer product:

antisymmetry $\mathbf{A}_{\langle n-1\rangle} \vee \mathbf{B}_{\langle n-1\rangle} = -\mathbf{B}_{\langle n-1\rangle} \vee \mathbf{A}_{\langle n-1\rangle}$, thus $\mathbf{C}_{\langle n-1\rangle} \vee \mathbf{C}_{\langle n-1\rangle} = 0$

distributivity $\mathbf{A}_{\langle n-1\rangle} \vee (\mathbf{B}_{\langle n-1\rangle} + \mathbf{C}_{\langle n-1\rangle}) = \mathbf{A}_{\langle n-1\rangle} \vee \mathbf{B}_{\langle n-1\rangle} + \mathbf{A}_{\langle n-1\rangle} \vee \mathbf{C}_{\langle n-1\rangle}$

associativity $\mathbf{A}_{\langle n-1\rangle} \vee (\mathbf{B}_{\langle n-1\rangle} \vee \mathbf{C}_{\langle n-1\rangle}) = (\mathbf{A}_{\langle n-1\rangle} \vee \mathbf{B}_{\langle n-1\rangle}) \vee \mathbf{C}_{\langle n-1\rangle}$

scalars commute $\mathbf{A}_{\langle n-1\rangle} \vee (\beta\,\mathbf{B}_{\langle n-1\rangle}) = \beta\,(\mathbf{A}_{\langle n-1\rangle} \vee \mathbf{B}_{\langle n-1\rangle})$

3.4 Metric and Some Inner Products

In order to compare the weight or the angle between two subspaces having different attitudes, one needs products which depend on the metric of $\bigwedge \mathbb{R}^n$, extending the usual **vector inner product** from vector algebra to higher dimensional subspaces.

The vector inner product (denoted by the *center dot* symbol) multiplies two arbitrary vectors and returns a scalar value that characterizes their metric relation:

$$\mathbf{a} \cdot \mathbf{b} = \mathsf{Q}\,(\mathbf{a},\mathbf{b})\,, \tag{3.3}$$

where Q is a scalar-valued function defining a metric on the vector space $\mathbb{R}^n$. A practical way to implement Q is by using a **metric matrix**:

$$\mathrm{M} = \begin{pmatrix} \mu^{1,1} & \mu^{1,2} & \cdots & \mu^{1,n} \\ \mu^{2,1} & \mu^{2,2} & \cdots & \mu^{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mu^{n,1} & \mu^{n,2} & \cdots & \mu^{n,n} \end{pmatrix},$$

where M is a symmetric positive-definite matrix encoding the inner product of pairs of basis vectors $\{\mathbf{e}_i\}_{i=1}^n$, and $\mu^{i,j} = \mathsf{Q}(\mathbf{e}_i,\mathbf{e}_j)$, for $1 \leq i,j \leq n$. Section 3.9 shows how one can give meaningful geometric interpretations for subspaces and related operations by assuming a metric for $\mathbb{R}^n$. For instance, by letting M be an identity matrix one is assuming Euclidean metric (*i.e.,* $\mathbf{e}_i \cdot \mathbf{e}_j$ is *one* for $i = j$, and *zero* otherwise), and the inner product becomes the well known **vector dot product** from linear algebra. The vector dot product is typically used to measure the smallest angle between two vectors.

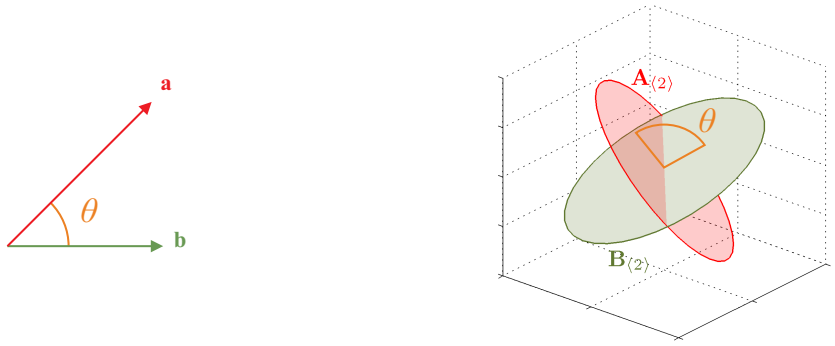
Sections 3.4.1 and 3.4.2 present two extensions of the vector inner product for general blades. See Dorst *et al.* (2007) and Perwass (2009) for other metric products (*i.e.,* products which depend on the metric of $\mathbb{R}^n$).

3.4.1 The Scalar Product

The **scalar product between blades** is denoted by $*$ and extends the vector inner product to arbitrary subspaces having the same dimensionality. The resulting scalar value characterizes the metric relation between the subspaces. When multiplying blades with different dimensionality the outcome is *zero*.

The scalar product is a mapping:

$$* : \bigwedge^k \mathbb{R}^n \times \bigwedge^k \mathbb{R}^n \to \mathbb{R}, \tag{3.4}$$



(a) (b)

Figure 3.3: Using the scalar product to measure angles. (a) In a Euclidean vector space, the vector inner product can be used to compute the smallest angle between two vectors: $\theta = \cos^{-1}\left((\mathbf{a}\cdot\mathbf{b}) \;/\; (\|\mathbf{a}\|\;\|\mathbf{b}\|)\right)$. (b) In the same way, the scalar product introduced in Section 3.4.1 allows one to measure the smallest angle between two arbitrary subspaces having the same dimensionality: $\theta = \cos^{-1}\left(\left(\mathbf{A}_{\langle 2\rangle} * \mathbf{B}_{\langle 2\rangle}\right) \;/\; \left(\left\|\mathbf{A}_{\langle 2\rangle}\right\|\;\left\|\mathbf{B}_{\langle 2\rangle}\right\|\right)\right)$.

and presents the following properties:

symmetry	$\mathbf{A}_{\langle r\rangle} * \mathbf{B}_{\langle s\rangle} = \mathbf{B}_{\langle s\rangle} * \mathbf{A}_{\langle r\rangle}$
distributivity	$\mathbf{A}_{\langle r\rangle} * (\mathbf{B}_{\langle s\rangle} + \mathbf{C}_{\langle t\rangle}) = \mathbf{A}_{\langle r\rangle} * \mathbf{B}_{\langle s\rangle} + \mathbf{A}_{\langle r\rangle} * \mathbf{C}_{\langle t\rangle}$
scalars commute	$\mathbf{A}_{\langle r\rangle} * (\beta\,\mathbf{B}_{\langle s\rangle}) = \beta\,(\mathbf{A}_{\langle r\rangle} * \mathbf{B}_{\langle s\rangle})$

It is important not to confuse the scalar product presented here with the vector dot product from linear algebra, also known as scalar product in vector algebra. The former operates on k-blades, while the latter operates only on vectors in a Euclidean space.

The vector inner product (Equation 3.3) is the special case of the scalar product for 1-blades. Such backward compatibility implies that the metric assumed for the original vector space $\mathbb{R}^n$ automatically extends to blades in $\bigwedge^k \mathbb{R}^n$. Also, such equivalence allows the comparison between two subspaces having higher dimensionality, while using the same basic principles adopted for vectors. For instance, in Euclidean space, the scalar product between k-blades is written as:

$$\mathbf{A}_{\langle k\rangle} * \mathbf{B}_{\langle k\rangle} = \left\|\mathbf{A}_{\langle k\rangle}\right\|\;\left\|\mathbf{B}_{\langle k\rangle}\right\|\cos\theta. \tag{3.5}$$

So, the scalar product can be used for computing the smallest angle between subspaces (under Euclidean metric), as shown in Figure 3.3 for $k = 1$ (Figure 3.3a) and $k = 2$ (Figure 3.3b). In the proposed subspace detection scheme, the scalar product is applied to check whether two blades have the same dimensionality and are not orthogonal.

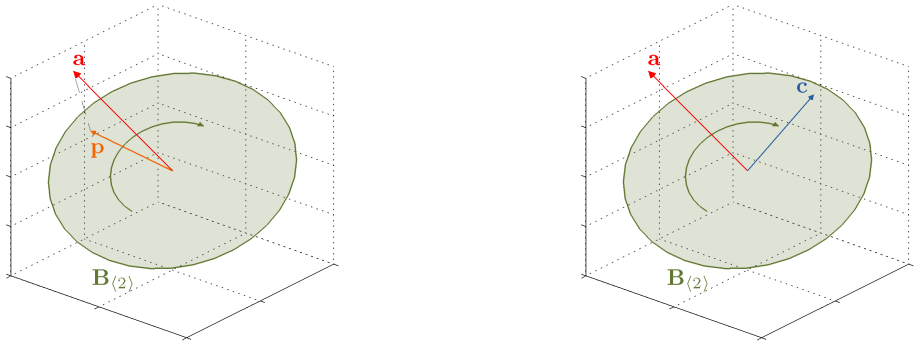
In Equation 3.5, $\left\|\mathbf{A}_{\langle k\rangle}\right\|$ is the **norm** of a blade. The square of the norm is computed as:

$$\left\|\mathbf{A}_{\langle k\rangle}\right\|^2 = \mathbf{A}_{\langle k\rangle} * \widetilde{\mathbf{A}}_{\langle k\rangle}, \tag{3.6}$$

where

$$\widetilde{\mathbf{A}}_{\langle k\rangle} = (-1)^{k(k-1)/2}\,\mathbf{A}_{\langle k\rangle} \tag{3.7}$$

is the **reverse** of a subspace. The reverse operation decomposes a blade $\mathbf{A}_{\langle k\rangle}$ in its vector factors (*e.g.*, $\mathbf{A}_{\langle k\rangle} = \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \cdots \wedge \mathbf{a}_k$) and produces the reverse $\widetilde{\mathbf{A}}_{\langle k\rangle}$ by reversing the order of such factors (*e.g.*, $\widetilde{\mathbf{A}}_{\langle k\rangle} = \mathbf{a}_k \wedge \mathbf{a}_{k-1} \wedge \cdots \wedge \mathbf{a}_1$). Thus, due to the antisymmetry of the outer product, the reverse operation produces a sign change related to the



(a) (b)

Figure 3.4: Geometric intuition for computing the contraction of $\mathbf{a}$ onto $\mathbf{B}_{\langle 2 \rangle}$ in a Euclidean space $\mathbb{R}^3$. (a) First, the 1-blade $\mathbf{a}$ is orthogonally projected onto $\mathbf{B}_{\langle 2 \rangle}$, resulting in vector $\mathbf{p}$. Then, vectors $\mathbf{a}$ and $\mathbf{p}$ are operated under the scalar product, resulting in a scalar value γ. (b) In turn, γ scales the portion of $\mathbf{B}_{\langle 2 \rangle}$ that is orthogonal to $\mathbf{a}$, resulting in $\mathbf{c}$.

grade k of the subspace (Equation 3.7). The sign change under the reversion exhibits a $+\ +\ -\ -\ +\ +\ -\ -\ \cdots$ pattern over the values of k, for $k \geq 0$. It is important to comment that the reverse operation is distributive over addition, so it can be applied to general multivectors.

This chapter does not present an algorithm for computing the scalar product regarding an arbitrary metric space, or a procedure for solving any other particular metric product from GA. Instead, Section 3.5 shows that some bilinear products (*e.g.,* $\wedge$, $*$, and $\rfloor$ introduced in the Section 3.4.2) can be defined in terms of a more general product, namely geometric product, for which an implementation is described.

3.4.2 The Left Contraction

One of the most important metric products for blades is the **left contraction**:

$$\mathbf{C}_{\langle s-r \rangle} = \mathbf{A}_{\langle r \rangle} \rfloor \mathbf{B}_{\langle s \rangle}, \tag{3.8}$$

denoted by $\rfloor$. Its geometric interpretation can be described as removing from $\mathbf{B}_{\langle s \rangle}$ (the *contractee*) the part that is "like" $\mathbf{A}_{\langle r \rangle}$ (the *contractor*), returning the portion $\mathbf{C}_{\langle s-r \rangle} \subseteq \mathbf{B}_{\langle s \rangle}$ that is "unlike" $\mathbf{A}_{\langle r \rangle}$ in the assumed metric. In Equation 3.8, if $\mathbf{A}_{\langle r \rangle}$ includes at least one vector factor that is orthogonal to $\mathbf{B}_{\langle s \rangle}$ (*e.g.,* when $r > s$) then the outcome is *zero*, because the condition $\mathbf{C}_{\langle s-r \rangle} \subseteq \mathbf{B}_{\langle s \rangle}$ cannot be satisfied. Figure 3.4 illustrates the case of a vector $\mathbf{a}$ contracted on $\mathbf{B}_{\langle 2 \rangle}$, resulting in the $\mathbf{c}$ vector. Euclidean space $\mathbb{R}^3$ is assumed for this example. In Figure 3.4a, the vector $\mathbf{p}$ is the projection of $\mathbf{a}$ onto $\mathbf{B}_{\langle 2 \rangle}$. It is the part on $\mathbf{B}_{\langle 2 \rangle}$ that is most like $\mathbf{a}$. When the left contraction is applied, $\mathbf{a}$ and $\mathbf{p}$ are multiplied together under the scalar product (Section 3.4.1), resulting in a scalar value γ. In Figure 3.4b, the part of $\mathbf{B}_{\langle 2 \rangle}$ orthogonal to $\mathbf{a}$ is then scaled by γ, resulting in vector $\mathbf{c}$. Note that the orientation of $\mathbf{c}$ is consistent with the orientation of both $\mathbf{p}$ and $\mathbf{B}_{\langle 2 \rangle}$. It is important to comment that $\mathbf{p}$ does not need to be explicitly computed while performing the left contraction, because the projection of $\mathbf{a}$ is implicitly handled by the product.

The left contraction defines a mapping:

$$\rfloor : \bigwedge^r \mathbb{R}^n \times \bigwedge^s \mathbb{R}^n \to \bigwedge^{s-r} \mathbb{R}^n.$$

Note that such mapping generalizes Equation 3.4 when $r = s$. Therefore, the scalar product is a particular case of the left contraction:

$$\mathbf{A}_{\langle k \rangle} \rfloor \mathbf{B}_{\langle k \rangle} = \mathbf{A}_{\langle k \rangle} * \mathbf{B}_{\langle k \rangle},$$

and the three metric products presented so far are backward compatible for 1-blades:

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a} * \mathbf{b} = \mathbf{a} \rfloor \mathbf{b}.$$

The properties of the left contraction are:

symmetry	$\mathbf{A}_{\langle r \rangle} \rfloor \mathbf{B}_{\langle s \rangle} = \mathbf{B}_{\langle s \rangle} \rfloor \mathbf{A}_{\langle r \rangle}$, iif $r = s$
distributivity	$\mathbf{A}_{\langle r \rangle} \rfloor (\mathbf{B}_{\langle s \rangle} + \mathbf{C}_{\langle t \rangle}) = \mathbf{A}_{\langle r \rangle} \rfloor \mathbf{B}_{\langle s \rangle} + \mathbf{A}_{\langle r \rangle} \rfloor \mathbf{C}_{\langle t \rangle}$
scalars commute	$\mathbf{A}_{\langle r \rangle} \rfloor (\beta \mathbf{B}_{\langle s \rangle}) = \beta (\mathbf{A}_{\langle r \rangle} \rfloor \mathbf{B}_{\langle s \rangle})$

3.5 The Geometric Product

The **geometric product** is the fundamental product in GA for two reasons: (i) unlike the other products, the geometric product is invertible; and (ii) many of the products presented so far (*i.e.,* $\wedge$, $*$, and $\rfloor$) can be defined in terms of it. By combining these two features, the geometric product is an invaluable tool for solving geometric problems. From our experience, the clear geometrical meaning of the other products is very helpful while designing a solution to a problem. However, we develop our final (and simpler) solutions by replacing the conventional products by the geometric product whenever possible, and by taking advantage of its invertibility while performing algebraic manipulations.

The geometric product has no special symbol and it is denoted by a *thin space*. For real values (*i.e.,* 0-dimensional subspaces) it is equivalent to the standard multiplication operation. For vectors, it is defined as the linear combination of the vector inner product (Equation 3.3) and the outer product (Section 3.3.1):

$$\mathbf{a}\,\mathbf{b} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}. \tag{3.9}$$

In Equation 3.9, when $\mathbf{a} \cdot \mathbf{b} = 0$, the outcome of the geometric product is the subspace spanned as the outer product of $\mathbf{a}$ and $\mathbf{b}$. On the other hand, when $\mathbf{a} \wedge \mathbf{b} = 0$, the result is the scalar value computed as the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$. In both cases, the geometric product of vectors returns a subspace. However, when both $\mathbf{a} \cdot \mathbf{b}$ and $\mathbf{a} \wedge \mathbf{b}$ are not zero, the geometric product produces a multivector of mixed grade. Section 3.8 shows that elements computed as the geometric product of an even number of unit invertible vectors are called **rotors**, and encode rotations under Euclidean metric.

The geometric product extends to elements of higher dimensionality. Unfortunately, for arbitrary dimensions, it is hard to provide an intuitive and/or illustrated description of the product. For this reason the general definition of the geometric product is provided in Appendix A. Such equations are the most conservative way to implement it, and can be used within any metric space. As an alternative, Dorst *et al.* (2007) describe how to implement the geometric product using matrix multiplication. However, the matrix approach suffers from numerical instability. Dorst *et al.* (2007) also describe how one can compute the geometric product of basis blades in an orthogonal metric space. Moreover, they show how one can temporarily switch the actual basis vectors of a nonorthogonal basis to a new basis that is orthonormal. This way, one can compute the geometric product

in the general case using a single algorithm. However, I believe that the set of equations presented in Appendix A provides a more elegant definition.

The **inverse geometric product** is denoted by $/$ (the *forward slash* symbol). In the following expression:

$$C = A \,/\, B = A\, B^{-1}, \tag{3.10}$$

A, B, and C are arbitrary multivectors. In the context of this thesis, however, such multivectors only encode blades or rotors (Section 3.8). Note from Equation 3.10 that having the inverse allows us to divide by multivectors. Therefore, the "divisor" B must be invertible. A blade (or rotor) B is invertible if $B\,\widetilde{B} \neq 0 \in \mathbb{R}$. The invertibility of arbitrary multivectors is discussed in (DORST; FONTIJINE; MANN, 2007). The **inverse** $\mathbf{B}_{\langle k \rangle}^{-1}$ of a blade $\mathbf{B}_{\langle k \rangle}$ satisfies $\mathbf{B}_{\langle k \rangle}\,\mathbf{B}_{\langle k \rangle}^{-1} = \mathbf{B}_{\langle k \rangle}^{-1}\,\mathbf{B}_{\langle k \rangle} = 1$, and is computed as:

$$\mathbf{B}_{\langle k \rangle}^{-1} = \frac{\widetilde{\mathbf{B}}_{\langle k \rangle}}{\left\|\mathbf{B}_{\langle k \rangle}\right\|^2}. \tag{3.11}$$

Since the geometric product can produce multivectors of mixed grade, it defines a mapping:

$$\bigwedge \mathbb{R}^n \times \bigwedge \mathbb{R}^n \to \bigwedge \mathbb{R}^n.$$

The properties of the geometric product are:

distributivity	$A\,(B + C) = A\,B + A\,C$
associativity	$A\,(B\,C) = (A\,B)\,C$
scalars commute	$A\,(\beta\,B) = \beta\,(A\,B)$
neither fully symmetric nor fully antisymmetric	$\exists\, A, B \in \bigwedge \mathbb{R}^n : A\,B \neq B\,A$

From the last property, one must be aware that the order of the elements matter while performing a sequence of geometric products. The only exceptions are when: the geometric product can be replaced by a scalar product (which is commutative); one of the operands is a scalar value (*i.e.,* scalars commute); or it multiplies together two orthogonal transformations (*e.g.,* two rotations in disjoint planes). Notice that the non-commutability of the geometric product implies that the division in Equation 3.10 is also non-commutative. Thus, recall that the operator $/$ is a division on the right by B.

3.5.1 Subspace Products from the Geometric Product

The outer product (Section 3.3.1), the scalar product (Section 3.4.1), and the left contraction (Section 3.4.2) are special cases of the geometric product. These products can be computed as:

$$\mathbf{A}_{\langle r \rangle} \wedge \mathbf{B}_{\langle s \rangle} = \left\langle \mathbf{A}_{\langle r \rangle}\,\mathbf{B}_{\langle s \rangle} \right\rangle_{r+s}, \tag{3.12}$$

$$\mathbf{A}_{\langle r \rangle} * \mathbf{B}_{\langle s \rangle} = \left\langle \mathbf{A}_{\langle r \rangle}\,\mathbf{B}_{\langle s \rangle} \right\rangle_{0}, \text{ and} \tag{3.13}$$

$$\mathbf{A}_{\langle r \rangle} \rfloor\, \mathbf{B}_{\langle s \rangle} = \left\langle \mathbf{A}_{\langle r \rangle}\,\mathbf{B}_{\langle s \rangle} \right\rangle_{s-r}, \tag{3.14}$$

where $\langle M \rangle_k$ denotes the **grade extraction** operation (or **grade projection**), which retrieves the k-grade part of a multivector M. Blades of negative grade or with grade higher than n are equal to *zero*. Therefore, the outer product is zero when $(r + s) > n$ (Equation 3.12), and the left contraction is zero when $r > s$ (Equation 3.14).

3.6 Dual Representation of Subspaces

The number of basis blades in each graded part of a multivector space $\bigwedge \mathbb{R}^n$ (Section 3.2) suggests the existence of a relation between k-blades and $(n-k)$-blades. Such symmetry in the number of basis elements of $\bigwedge^k \mathbb{R}^n$ and $\bigwedge^{n-k} \mathbb{R}^n$ is related to the equivalence on the combinations of basis vectors (*i.e.,* $\binom{n}{k} = \binom{n}{n-k}$). In GA, the **dual** representation of a k-blade $\mathbf{A}_{\langle k \rangle}$ is its $(n-k)$-dimensional orthogonal complement, with the same absolute weight of $\mathbf{A}_{\langle k \rangle}$, and a well-defined orientation. The dual of $\mathbf{A}_{\langle k \rangle}$ is computed as:

$$\mathbf{A}^*_{\langle k \rangle} = \mathbf{A}_{\langle k \rangle} \rfloor \mathbf{I}^{-1}_{\langle n \rangle}, \tag{3.15}$$

where $\square^*$ (the *superscripted asterisk* symbol) denotes the **taking the dual** operation, and $\mathbf{I}_{\langle n \rangle}$ is the unit pseudoscalar of the n-dimensional space (Equation 3.2). Optionally, one can replace the left contraction in Equation 3.15 by a geometric product because they are equivalent in this context, since $\mathbf{A}_{\langle k \rangle} \subseteq \mathbf{I}^{-1}_{\langle n \rangle}$. However, the use of the left contraction provides to Equation 3.15 a clear geometric interpretation: returning the portion of $\mathbf{I}^{-1}_{\langle n \rangle}$ that is orthogonal to $\mathbf{A}_{\langle k \rangle}$.

It is important to emphasize that $\mathbf{A}^*_{\langle k \rangle}$ is an $(n-k)$-blade, as suggests the outcome of $\mathbf{A}_{\langle k \rangle}$ contracted on $\mathbf{I}^{-1}_{\langle n \rangle}$ in Equation 3.15. Therefore, the dualization defines a mapping:

$$\square^* : \bigwedge^k \mathbb{R}^n \to \bigwedge^{n-k} \mathbb{R}^n.$$

At first sight, the dual of the dual representation of a blade should result in the direct representation of the blade. However, as demonstrated in Equation 3.16, $\left(\mathbf{A}^*_{\langle k \rangle}\right)^* = \mathbf{A}_{\langle k \rangle}$ does not hold in the general case:

$$\left(\mathbf{A}^*_{\langle k \rangle}\right)^* = \left(\mathbf{A}_{\langle k \rangle} \rfloor \mathbf{I}^{-1}_{\langle n \rangle}\right) \rfloor \mathbf{I}^{-1}_{\langle n \rangle} = \mathbf{A}_{\langle k \rangle} \mathbf{I}^{-1}_{\langle n \rangle} \mathbf{I}^{-1}_{\langle n \rangle} = (-1)^{n(n-1)/2} \mathbf{A}_{\langle k \rangle}. \tag{3.16}$$

Equation 3.16 shows that the successive application of two dualization operations may change the orientation of the resulting blade according to the value of n. It is because $\mathbf{I}^{-1}_{\langle n \rangle} \mathbf{I}^{-1}_{\langle n \rangle} = (-1)^{n(n-1)/2}$. In order to correct this issue, it is necessary to define the **taking the undual** operation as:

$$\mathbf{D}^{-*}_{\langle n-k \rangle} = \mathbf{D}_{\langle n-k \rangle} \rfloor \mathbf{I}_{\langle n \rangle}. \tag{3.17}$$

This way, the dual representation of a blade can be correctly mapped back to its direct representation:

$$\left(\mathbf{A}^*_{\langle k \rangle}\right)^{-*} = \left(\mathbf{A}_{\langle k \rangle} \rfloor \mathbf{I}^{-1}_{\langle n \rangle}\right) \rfloor \mathbf{I}_{\langle n \rangle} = \mathbf{A}_{\langle k \rangle} \mathbf{I}^{-1}_{\langle n \rangle} \mathbf{I}_{\langle n \rangle} = \mathbf{A}_{\langle k \rangle}. \tag{3.18}$$

In Equation 3.18, $\mathbf{I}^{-1}_{\langle n \rangle} \mathbf{I}_{\langle n \rangle} = 1$ ensures that the relative orientation of $\mathbf{A}_{\langle k \rangle}$ will be preserved for all n.

3.7 Meet and Join of Subspaces

The **meet** and **join** products are the GA analogs of intersection and union operators from set theory. For any two blades $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$ one can factor out a blade $\mathbf{M}_{\langle t \rangle}$ from both $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$:

$$\mathbf{A}_{\langle r \rangle} = \mathbf{A}'_{\langle r-t \rangle} \wedge \mathbf{M}_{\langle t \rangle} \text{ and } \mathbf{B}_{\langle s \rangle} = \mathbf{M}_{\langle t \rangle} \wedge \mathbf{B}'_{\langle s-t \rangle}.$$

Meet returns the subspace shared by $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$:

$$\mathbf{A}_{\langle r \rangle} \cap \mathbf{B}_{\langle s \rangle} = \mathbf{M}_{\langle t \rangle}, \tag{3.19}$$

while the join is the subspace spanned by the disjoint and by the common parts of $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$:

$$\mathbf{A}_{\langle r \rangle} \cup \mathbf{B}_{\langle s \rangle} = \mathbf{A}'_{\langle r-t \rangle} \wedge \mathbf{M}_{\langle t \rangle} \wedge \mathbf{B}'_{\langle s-t \rangle}. \tag{3.20}$$

Both meet (Equation 3.19) and join (Equation 3.20) are independent of the particular metric since they are based on factorization by the nonmetric outer product.

In the implementations produced for this thesis the join of two arbitrary subspaces (*i.e.,* $\mathbf{J}_{\langle r+s-t \rangle} = \mathbf{A}_{\langle r \rangle} \cup \mathbf{B}_{\langle s \rangle}$) is computed by using the efficient algorithm proposed by Fontijne (2008). Once one has the join, the meet can be computed using:

$$\mathbf{M}_{\langle t \rangle} = \mathbf{A}_{\langle r \rangle} \cap \mathbf{B}_{\langle s \rangle} = \left(\mathbf{B}^*_{\langle s \rangle} \wedge \mathbf{A}^*_{\langle r \rangle}\right)^{-*}$$

by recalling that the dual (Equation 3.15) and undual (Equation 3.17) operations, in this case, are not relative to the pseudoscalar of the total space ($\mathbf{I}_{\langle n \rangle}$), but of the pseudoscalar $\mathbf{J}_{\langle r+s-t \rangle}$ of the space within which the intersection problem resides.

Meet and join are nonlinear products. However, if $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$ are disjoint, then $\mathbf{M}_{\langle t \rangle}$ is a scalar value (*i.e.,* $\mathbf{A}_{\langle r \rangle} \cap \mathbf{B}_{\langle s \rangle} = \mathbf{M}_{\langle 0 \rangle} = 1$, a 0-blade) and the join reduces to the outer product (*i.e.,* $\mathbf{A}_{\langle r \rangle} \cup \mathbf{B}_{\langle s \rangle} \equiv \mathbf{A}_{\langle r \rangle} \wedge \mathbf{B}_{\langle s \rangle}$), which is linear. Similarly, if the join of $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$ is the total space (*i.e.,* $\mathbf{A}_{\langle r \rangle} \cup \mathbf{B}_{\langle s \rangle} = \mathbf{I}_{\langle n \rangle}$, the pseudoscalar) the meet reduces to the regressive product (*i.e.,* $\mathbf{A}_{\langle r \rangle} \cap \mathbf{B}_{\langle s \rangle} \equiv \mathbf{A}_{\langle r \rangle} \vee \mathbf{B}_{\langle s \rangle}$), which is also linear and nonmetric.

3.8 Encoding Rotations with Rotors

The **reflection** of an arbitrary vector $\mathbf{a}$ with respect to an invertible vector $\mathbf{v}$ is obtained using a **sandwiching construction** involving the geometric product:

$$\mathbf{a}' = -\mathbf{v}\,\mathbf{a} \,/\, \mathbf{v} = -\mathbf{v}\,\mathbf{a}\,\mathbf{v}^{-1},$$

where $\mathbf{a}'$ denotes the transformed vector, $/$ denotes the inverse geometric product (Equation 3.10), and $\square^{-1}$ is the inverse of a blade (Equation 3.11). A sequence of reflection operations results in an **orthogonal transformation** applied to $\mathbf{a}$. For instance, an even number of reflections under Euclidean metric gives a rotation while an odd number of reflections represent a reflection followed by a rotation, or a rotation followed by a reflection. Figure 3.5 shows how the double reflection of $\mathbf{a}$ in vectors $\mathbf{p}$ and $\mathbf{q}$ is equivalent to the rotation of $\mathbf{a}$ in the $\mathbf{p} \wedge \mathbf{q}$ plane. In such a case, the rotation angle ϕ is twice the angle between $\mathbf{p}$ and $\mathbf{q}$. The sense of rotation is given by the orientation of the 2-blade (*i.e.,* from $\mathbf{p}$ to $\mathbf{q}$) and by the sign of ϕ.

A **rotor** is defined as the geometric product of an even number of unit invertible vectors. Under Euclidean metric, rotors encode rotations. In Euclidean 3-dimensional spaces, they are closely related to unit **quaternions** (see (DORST; FONTIJINE; MANN, 2007; PERWASS, 2009) for details). In fact, rotors are the generalization of quaternions to n-dimensional spaces.

The transformation encoded by a rotor $\boldsymbol{T}$ is applied to a k-blade $\mathbf{A}_{\langle k \rangle}$ by using the sandwiching construction:

$$\mathbf{A}'_{\langle k \rangle} = \boldsymbol{T}\,\mathbf{A}_{\langle k \rangle} \,/\, \boldsymbol{T} = \boldsymbol{T}\,\mathbf{A}_{\langle k \rangle}\,\boldsymbol{T}^{-1} = \boldsymbol{T}\,\mathbf{A}_{\langle k \rangle}\,\widetilde{\boldsymbol{T}},$$

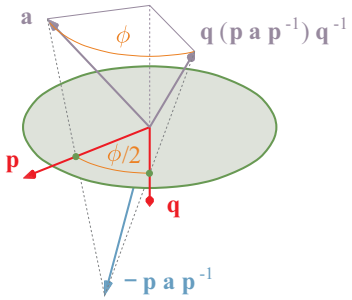


Figure 3.5: Rotation as double reflection. Let $\mathbf{a}$ be a vector in a Euclidean space. By reflecting $\mathbf{a}$ in vector $\mathbf{p}$ and, in turn, in vector $\mathbf{q}$, one gets the same result as by rotating $\mathbf{a}$ in the $\mathbf{p} \wedge \mathbf{q}$ plane by ϕ radians, where $\phi/2$ is the angle from $\mathbf{p}$ to $\mathbf{q}$.

where $\widetilde{\square}$ is the reverse (Equation 3.7). As for the inverse of blades, the inverse of $\boldsymbol{T}$ is computed using Equation 3.11. However, since rotors have unit positive squared norm, their inverse reduces to the reverse (*i.e.,* $\boldsymbol{T}^{-1} = \widetilde{\boldsymbol{T}}$).

As orthogonal transformations, rotors preserve both the symmetric inner product and the outer product. This way, the structure preservation of rotors holds for the geometric product (Equation 3.9), and hence to all other products in GA:

$$\boldsymbol{T}\ (A \circ B)\ /\ \boldsymbol{T} = (\boldsymbol{T}\ A\ /\ \boldsymbol{T}) \circ (\boldsymbol{T}\ B\ /\ \boldsymbol{T})\,, \tag{3.21}$$

where the $\circ$ symbol represents *any* product of GA, and, as a consequence, any operation defined from the products (*e.g.,* inversion and duality):

$$\boldsymbol{T}\ (A^{\circ})\ /\ \boldsymbol{T} = (\boldsymbol{T}\ A\ /\ \boldsymbol{T})^{\circ}\,.$$

An alternative and more practical way to define rotors is to use the exponential of 2-blades. Under Euclidean metric, the rotor $\boldsymbol{R}$ encoding a rotation of ϕ radians on the unit plane $\mathbf{P}_{\langle 2 \rangle}$ is given by:

$$\boldsymbol{R} = \exp\left(-\frac{\phi}{2}\mathbf{P}_{\langle 2 \rangle}\right) = \cos\left(\frac{\phi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\mathbf{P}_{\langle 2 \rangle}. \tag{3.22}$$

By using the exponential form one can easily define a rotation in an arbitrary plane without being concerned about the handedness of the space. This is because the sense of rotation is related only to the given angle and to the rotation plane. Also, the exponential form allows the definition of rotors directly from the 2-blades, instead of from pairs of vectors spanning them. In Chapter 4, I propose the use of rotations in the parameterization of p-dimensional subspaces.

3.9 Models of Geometry

So far, this chapter has presented the most fundamental concepts of GA. This section disscusses four MOGs that can be used in daily computations. A MOG provides a practical geometric interpretation to blades, rotors, and other elements in GA. Such interpretation is achieved by embedding the d-dimensional **base space** $\mathbb{R}^d$ (*i.e.,* space where the geometric interpretation happens) into an n-dimensional **representational space** $\mathbb{R}^n$ (*i.e.,* the total vector space), and by defining a metric to the representational space. The geometric properties of the space depend on the chosen metric. Appendix B presents a quick reference for defining geometric primitives as blades in the discussed MOGs, from parameters that are typically used with linear algebra.

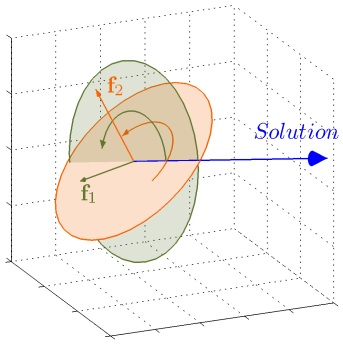


Figure 3.6: The outer product can be used to solve homogeneous systems of linear equations. Here, $\mathbf{f}_1$ and $\mathbf{f}_2$ are the normal vectors of the planes related to the equations of the system presented in Equation 3.23. The solution set is the vector defined by the intersection of the planes. It is compute as $(\mathbf{f}_1 \wedge \mathbf{f}_2)^{-*}$.

3.9.1 The Euclidean Model

As the name suggests, in the Euclidean model one assumes Euclidean metric for $\mathbb{R}^n$. This way, k-blades are geometrically interpreted as k-dimensional Euclidean subspaces, *i.e.,* oriented **flats** (*e.g.,* straight lines, planes, and their higher-dimensional counterparts) that pass through the origin of the vector space. The Euclidean model has been used to illustrate all examples given so far (Figures 3.1 to 3.5).

Euclidean subspaces in $\mathbb{R}^n$ are important because they represent the solution set to any homogeneous system of linear equations with n variables. For instance, consider the following system:

$$\begin{cases} 2\,\mathbf{e}_1 - 3\,\mathbf{e}_2 = 0 \\ \mathbf{e}_1 - 2\,\mathbf{e}_2 + 3\,\mathbf{e}_3 = 0 \end{cases} \tag{3.23}$$

Each equation of the system is associated with a plane that passes through the origin of $\mathbb{R}^3$. As depicted in Figure 3.6, the vectors $\mathbf{f}_1 = 2\,\mathbf{e}_1 - 3\,\mathbf{e}_2$ and $\mathbf{f}_2 = \mathbf{e}_1 - 2\,\mathbf{e}_2 + 3\,\mathbf{e}_3$ are the normal vectors (*i.e.,* the dual representation) of such planes. The solution set is the intersection of the planes, which can be computed using the outer product:

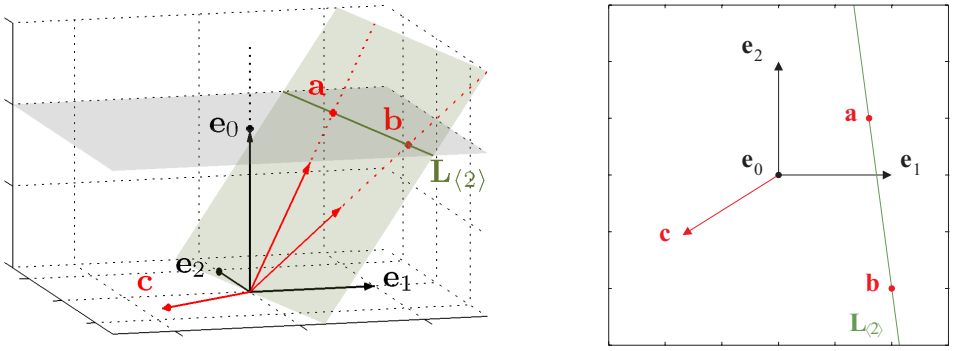
$$(\mathbf{f}_1 \wedge \mathbf{f}_2)^{-*} = 9\,\mathbf{e}_1 + 6\,\mathbf{e}_2 + \mathbf{e}_3. \tag{3.24}$$

Here, the 2-blade spanned as the outer product of $\mathbf{f}_1$ and $\mathbf{f}_2$ is the dual of the solution. The final solution is obtained by taking its undual. The undual operation is defined in Equation 3.17. The resulting subspace will be *zero* when the system has no solution. Note that the technique presented in Equation 3.24 can also be used to solve underdetermined systems (*i.e.* systems with more variables than the number of homogeneous linear equations plus one). In such a case, the result is a subspace whose dimensionality is higher than one.

3.9.2 The Homogeneous Model

The homogeneous (or projective) model (HESTENES; ZIEGLER, 1991) is similar to the use of homogeneous coordinates in linear algebra. It assumes Euclidean metric and a representational space $\mathbb{R}^{d+1}$ with basis $\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_d\}$. In this MOG, the d-dimensional base space is embedded in $\mathbb{R}^{d+1}$ in such a way that the extra basis vector $\mathbf{e}_0$ is interpreted as the origin of the base space. In Figure 3.7a, the plane parallel to $\mathbf{e}_1 \wedge \mathbf{e}_2$ is the homogeneous representation of the 2-dimensional base space in Figure 3.7b.

In the homogeneous model, vectors are geometrically interpreted as points. A **proper point** is a vector defining a finite location $(\alpha^1, \alpha^2, \cdots, \alpha^d)$ in the base space. Such a



(a) Representational space $\mathbb{R}^{2+1}$ (b) Base space $\mathbb{R}^2$

Figure 3.7: Geometric interpretation of blades in the homogeneous model. In (a), the plane parallel to $\mathbf{e}_1 \wedge \mathbf{e}_2$ is the homogeneous representation of the 2-dimensional base space in (b). The geometric interpretation of blades is given by their intersection with the base space. For instance, vectors $\mathbf{e}_0$, $\mathbf{a}$, and $\mathbf{b}$ in (a) are interpreted as proper points in (b), while vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{c}$ in (a) are interpreted as improper points, or directions, in (b). The straight line in (b) is defined by the intersection of a 2-blade with the base space in (a). In such a case, $\mathbf{L}_{\langle 2 \rangle} = \mathbf{a} \wedge \mathbf{b}$.

location is given by the intersection of the 1-blade with the base space (see $\mathbf{e}_0$, $\mathbf{a}$, and $\mathbf{b}$ in Figure 3.7). Unit proper points are written in the form:

$$\mathbf{p} = \mathbf{e}_0 + \alpha^1 \mathbf{e}_1 + \alpha^2 \mathbf{e}_2 + \cdots + \alpha^d \mathbf{e}_d. \tag{3.25}$$

Note that the coefficient assigned to $\mathbf{e}_0$ in Equation 3.25 is equal to *one*. A general proper point $\gamma \mathbf{p}$ is a weighted version of a unit point, and it is interpreted as having the same location (*i.e.,* the weight does not change the geometric interpretation of a blade).

When a vector is parallel to the base space (as $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{c}$ in Figure 3.7) it is called an **improper point**, or a point at infinity. Improper points can be seen as directions, because they are in the purely directional space $\mathbb{R}^d$ of the representational space $\mathbb{R}^{d+1}$. Unlike proper points, directions have the coefficient of $\mathbf{e}_0$ equal to *zero*:

$$\mathbf{d} = \beta^1 \mathbf{e}_1 + \beta^2 \mathbf{e}_2 + \cdots + \beta^d \mathbf{e}_d. \tag{3.26}$$

Higher dimensional oriented flat subspaces, like straight lines and planes, are spanned as the outer product of proper and improper points. For instance, the line in Figure 3.7 is defined as $\mathbf{L}_{\langle 2 \rangle} = \mathbf{a} \wedge \mathbf{b}$. Optionally, one can create a line from a proper point and a direction by using the same construction. The result will be a 2-blade to be used as a computing primitive.

In the homogeneous model, 3-blades are geometrically interpreted as planes. As one would expect, they are defined in d-dimensional base spaces (for $d \geq 3$) as the outer product of: (i) three proper points; (ii) two proper points and one direction; or (iii) one proper point and two directions; as far the vectors are linearly independent. As one can see, the definition of k-flats (for $0 \leq k < d$) in the homogeneous model is straightforward. It is based on the outer product of $(k+1)$ vectors. Blades spanned exclusively from improper points are geometrically interpreted as k-flats at infinity.

3.9.3 The Conformal Model

The conformal model (HESTENES, 1987, 2001) improves and extends the homogeneous model. Unlike the homogeneous model, in the conformal model blades can be

geometrically interpreted not only as directions and flats, but also as **rounds** (*e.g.,* point pairs, circles, spheres, and their higher-dimensional counterparts) and **tangent subspaces**.

The representational vector space $\mathbb{R}^{d+2}$ of the conformal model is defined from the basis vectors $\{\mathbf{o}, \mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_d, \boldsymbol{\infty}\}$, where the d-dimensional base space is enhanced with two extra dimensions: $\mathbf{o}$, a null vector interpreted as the origin point (pronounced "no"); and $\boldsymbol{\infty}$, a null vector interpreted as the point at infinity (pronounced "ni"). They are null vectors due to the special metric assumed in the conformal model. Equation 3.27 shows the multiplication table for the vector inner product of the basis vectors:

$$\begin{array}{c|cccccc} \cdot & \mathbf{o} & \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_d & \boldsymbol{\infty} \\ \hline \mathbf{o} & 0 & 0 & 0 & \cdots & 0 & -1 \\ \mathbf{e}_1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \mathbf{e}_2 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{e}_d & 0 & 0 & 0 & \cdots & 1 & 0 \\ \boldsymbol{\infty} & -1 & 0 & 0 & \cdots & 0 & 0 \end{array} \tag{3.27}$$

Note that $\mathbf{o} \cdot \mathbf{o} = \boldsymbol{\infty} \cdot \boldsymbol{\infty} = 0$, while $\mathbf{o} \cdot \boldsymbol{\infty} = -1$. One consequence of this definition is that the inner product of two unit finite points (*i.e.,* points at a finite distance from the origin) is given in terms of the square of the Euclidean distance between them:

$$\mathbf{p} \cdot \mathbf{q} = -\frac{1}{2} \sum_{i=1}^{d} \left(\alpha^i - \beta^i\right)^2, \tag{3.28}$$

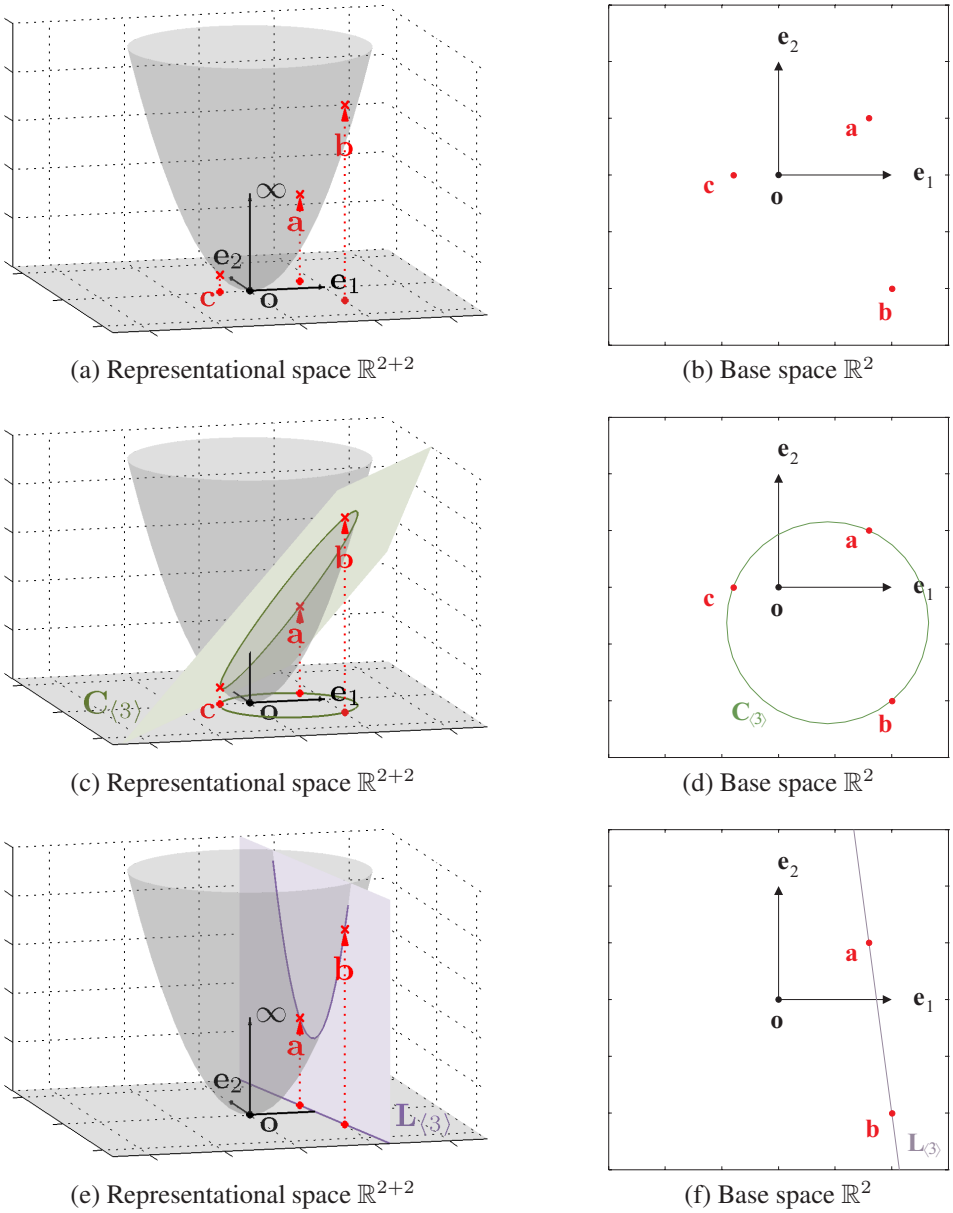
where $(\alpha^1, \alpha^2, \cdots, \alpha^d)$ and $(\beta^1, \beta^2, \cdots, \beta^d)$ are the location of points $\mathbf{p}$ and $\mathbf{q}$, respectively. This way, finite points are also null vectors (*i.e.,* $\mathbf{p} \cdot \mathbf{p} = 0$). Having the inner product related to the Euclidean distance of points is an interesting feature of the conformal model. That allows the definition of a coordinate-free solution, because the comparison of points is independent of their location relative to an assumed origin. In the homogeneous model, the outcome of the inner product of proper points depends on how far they are from $\mathbf{e}_0$.

Unit finite points are written in the form:

$$\mathbf{p} = \mathbf{o} + \alpha^1 \mathbf{e}_1 + \alpha^2 \mathbf{e}_2 + \cdots + \alpha^d \mathbf{e}_d + \frac{1}{2} \sum_{i=1}^{d} \left((\alpha^i)^2\right) \boldsymbol{\infty}, \tag{3.29}$$

while general finite points are weighted points ($\gamma\,\mathbf{p}$) having the same location. Figure 3.8a shows that the set of all unit finite points in a 2-dimensional base space defines a paraboloid in the $\boldsymbol{\infty}$-direction of the representational space. In this example, the total 4-dimensional vector space is presented as a 3-dimensional homogeneous space, where $\mathbf{o}$ is treated as the homogeneous coordinate. The base space shown in Figure 3.8b corresponds to the plane at the bottom of Figure 3.8a. Note that the paraboloid touches the base space at its origin, and that the location of finite points (*e.g.,* $\mathbf{o}$, $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$) is given by their orthogonal projection onto the base space.

From the outer product of two, three, and four finite points one builds, respectively, point pairs, circles, and spheres. So, the construction of k-spheres (for $0 \leq k < d$) is straightforward. It is achieved from the outer product of $(k+2)$ points. Figure 3.8d shows a circle defined as $\mathbf{C}_{\langle 3 \rangle} = \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$. Note in Figure 3.8c that such circle is a cross



(a) Representational space $\mathbb{R}^{2+2}$

(b) Base space $\mathbb{R}^2$

(c) Representational space $\mathbb{R}^{2+2}$

(d) Base space $\mathbb{R}^2$

(e) Representational space $\mathbb{R}^{2+2}$

(f) Base space $\mathbb{R}^2$

Figure 3.8: Geometric interpretation of blades in the conformal model. The representation of a 2-dimensional base space is shown on the left. In such a representation, the basis vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and ∞ are seen as a homogeneous space having the basis vector $\mathbf{o}$ as homogeneous coordinate. The 2-dimensional base space (on the right) corresponds to the plane at the bottom of the images on the left. Points on the paraboloid are interpreted as finite points in the base space (see $\mathbf{o}$, $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ in (a) and (b)). As depicted in (c) and (d), the circle defined by $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ is computed as $\mathbf{C}_{\langle 3 \rangle} = \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$. In (e) and (f), the straight line that passes thought $\mathbf{a}$ and $\mathbf{b}$ is defined as $\mathbf{L}_{\langle 3 \rangle} = \mathbf{a} \wedge \mathbf{b} \wedge \infty$.

section of the paraboloid, orthogonally projected onto the base space. Appendix B includes formulas for defining k-spheres from their usual center-radius parameterization.

As commented in Section 3.9.2, improper points in the homogeneous model are characterized as points at infinity in such a way that each direction is one of these points. In the conformal model, however, the base space is "closed". It means that ∞ is the unique point at infinity, with a well defined location that one can approach from *any* direction.

So, $\boldsymbol{\infty}$ is common to all flat subspaces, because they stretch to infinity. This way, straight lines and planes are built as the outer product of $\boldsymbol{\infty}$ with, respectively, two and three finite points. For instance, the line passing through $\mathbf{a}$ to $\mathbf{b}$ in Figure 3.8e is computed as $\mathbf{L}_{\langle 3 \rangle} = \mathbf{a} \wedge \mathbf{b} \wedge \boldsymbol{\infty}$. It is important to note that all the equations for construction of flat subspaces in the conformal model are backward compatible with the ones from the homogeneous model, but including $\boldsymbol{\infty}$.

In Figure 3.8e, the cross section defined by $\mathbf{L}_{\langle 3 \rangle}$ is a parabola whose orthogonal projection on the base space is a straight line (Figure 3.8f). Now, note how similar $\mathbf{C}_{\langle 3 \rangle}$ and $\mathbf{L}_{\langle 3 \rangle}$ are (Figures 3.8c and 3.8e). Both are 3-blades which define cross sections in the paraboloid. In fact, the projection of both cross sections can be interpreted as circles in the base space, where $\mathbf{L}_{\langle 3 \rangle}$ (Figure 3.8f) is a circle with infinite radius. Such a generality on the geometric interpretation of blades is explored by the proposed detection framework in order to perform concurrent detection of subspaces with different interpretations but with the same dimensionality.

In order to be interpreted as a direction, a blade must have only directional properties and no locational aspects. The location of a blade is defined in terms of the assumed origin point $\mathbf{o}$. Therefore, directions (also called **free blades**) are built as $\mathbf{A}_{\langle k \rangle} \wedge \boldsymbol{\infty}$, where $\mathbf{A}_{\langle k \rangle} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$. This is the natural extension of blades, which are interpreted as directions in the homogeneous model, to the conformal model.

The final type of conformal blade is the **tangent subspace**. As the name suggests, such primitives are tangent to something. In such a case, they encode the subspace tangent to rounds or flats at a given location. Therefore, tangent subspaces have a point-like interpretation, and also direction information assigned to them. For a given round (or flat) $\mathbf{X}_{\langle k \rangle}$ passing thought the point $\mathbf{p}$, the tangent subspace at the location of $\mathbf{p}$ is $\mathbf{T}_{\langle k-1 \rangle} = \mathbf{p} \rfloor \widehat{\mathbf{X}}_{\langle k \rangle}$. The general equation for tangents subspaces is:

$$\mathbf{T}_{\langle k+1 \rangle} = \mathbf{p} \wedge (-\mathbf{p} \rfloor (\widehat{\mathbf{A}}_{\langle k \rangle} \boldsymbol{\infty})),$$

where $\mathbf{A}_{\langle k \rangle} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ defines the direction.

3.9.4 The Conic Model

In a recent work, Perwass and Forstner (2006) show how to embed a 2-dimensional base space $\mathbb{R}^2$ in a 6-dimensional representation space $\mathbb{R}^6$ in order to encode **conic sections** (*e.g.,* circle, ellipse, straight line, hyperbola, parallel line pair, and intersecting line pair) as blades.

This MOG assumes Euclidean metric and the basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5, \mathbf{e}_6\}$ for $\mathbb{R}^6$. Unit finite points are written in the form:

$$\mathbf{p} = \alpha\,\mathbf{e}_1 + \beta\,\mathbf{e}_2 + \frac{1}{\sqrt{2}}\,\mathbf{e}_3 + \frac{1}{\sqrt{2}}\,(\alpha)^2\,\mathbf{e}_4 + \frac{1}{\sqrt{2}}\,(\beta)^2\,\mathbf{e}_5 + \alpha\,\beta\,\mathbf{e}_6,$$

where α and β define the location of a point in the base space.

From the outer product of two, three, and four finite points one builds point pairs, point triplets, and point quadruplets. The outer product of five distinct points defines a 5-blade interpreted as one of the types of conic sections.

3.10 Discussion

The main concepts of GA used in this thesis were introduced in the current chapter. GA is a mathematical formalism that allows the development of solutions whose geomet-

rical meaning can be written directly as algebraic equations. Also, it treats subspaces as primitives for computation. As such, GA is an appropriate tool for modeling a generalized subspace detection framework.

In GA, each MOG assumes a particular metric for the representational space $\mathbb{R}^n$. Such a metric is key for the geometrical interpretation of blades. Table 3.1 summarizes the geometric primitives that can be represented as blades in the MOGs discussed in Sections 3.9.1 to 3.9.4. This table can be extended to higher dimensions, to other geometric shapes, and to other MOGs. See (DORST; FONTIJINE; MANN, 2007; PERWASS, 2009) for in-depth discussions on the subject.

The proposed subspace detection scheme is based on nonmetric definition of GA. As a result, the actual metric can be replaced by any convenient metric. In Chapters 4, 5 and 6, Euclidean metric is assumed for $\mathbb{R}^n$ in all computations in order to prescind the interpretation of blades in the actual MOG. This way, blades can be (temporally) interpreted as Euclidean subspaces rather than specific geometric shapes.

Table 3.1: Geometric interpretation of subspaces according to four MOGs (E: Euclidean, H: Homogeneous, C: Conformal, and N: Conic). n indicates the dimensionality of the **representational space**, and d is the dimensionality of the **base space**. The extra dimensions (added to d) are imposed by the MOG. A 2-dimensional base space is assumed. The only exceptions are for shapes marked with †. In such cases, the base space is 3-dimensional. The rows group shapes according to their class (frees, flats, rounds, tangents, point sets, and conic sections). The table entries show the dimensionality of the subspaces.

		E	**H**	**C**	**N**
	$n \rightarrow$	d	$d+1$	$d+2$	6
Frees	1-dimensional direction	1	1	2	
	2-dimensional direction	2	2	3	
	3-dimensional direction†	3	3	4	
Flats	Flat point		1	2	
	Straight line		2	3	
	Plane†		3	4	
Rounds	Point pair			2	
	Circle			3	
	Sphere†			4	
Tangents	Point			1	
	1-dimensional tangent direction			2	
	2-dimensional tangent direction†			3	
Point Sets	Point				1
	Point pair				2
	Point triplet				3
	Point quadruplet				4
Conics	Circle, Ellipse, Straight line, Hyperbola, Parallel line pair, Intersecting line pair				5

4 PARAMETERIZATION OF SUBSPACES

This chapter provides the derivation for a model function that characterizes an arbitrary p-blade thought a set of $p\,(n-p)$ angular parameters, where n is the dimensionality of the representational vector space $\mathbb{R}^n$ from which a multivector space $\bigwedge \mathbb{R}^n$ is built. The basic properties of subspaces are revisited. A parameterization of such properties is described for vectors (*i.e.,* 1-blades in $\bigwedge \mathbb{R}^n$) and pseudovectors (*i.e.,* $(n-1)$-blades in $\bigwedge \mathbb{R}^n$). The parameterization of blades with arbitrary dimensionality is comprised by the parameters characterizing the vectors (or pseudovectors) used in their construction. The choice for one of the two constructions is based on the value of p and n, and will be discussed later in Section 4.1.

As pointed out in Section 3.1, an arbitrary subspace presents four basic properties: **attitude**, **weight**, **orientation**, and **direction**. These properties are intrinsic to the construction of the blade by the outer product of its vector factors, or by the regressive product of pseudovectors. Both the outer and the regressive products are independent of the metric of $\mathbb{R}^n$. Therefore, the basic properties are also independent of the assumed MOG and its metric. The nonmetric nature of these definitions allows the replacement of the actual metric by any convenient metric while defining a parameterization. In the following derivations Euclidean metric is assumed for $\mathbb{R}^n$ in all computations in order to prescind the interpretation of subspaces in the context of a given MOG. This way, blades are interpreted as Euclidean subspaces rather than as specific geometric shapes.

Despite the attitude to be a nonmetric property, it affects the geometric interpretation of the subspace. This is illustrated in Figure 3.7 for $\mathbf{L}_{\langle 2\rangle} = \mathbf{a} \wedge \mathbf{b}$, where the interpretation of $\mathbf{L}_{\langle 2\rangle}$ as a straight line is shown in Figure 3.7b according to the homogeneous MOG. The attitude of $\mathbf{L}_{\langle 2\rangle}$ is its stance in the surrounding 3-dimensional space (Figure 3.7a). By changing the sign of the subspace (*i.e.,* $-\mathbf{L}_{\langle 2\rangle}$) one changes the orientation of the line from $\mathbf{a} \to \mathbf{b}$ to $\mathbf{b} \to \mathbf{a}$. Both lines defined by $\mathbf{L}_{\langle 2\rangle}$ and $-\mathbf{L}_{\langle 2\rangle}$ determine the same set of points stretching to infinity because they have the same attitude. Multiplying the subspace by a scalar value (*e.g.,* $3\,\mathbf{L}_{\langle 2\rangle}$) produces a blade with a different weight. Again, both $\mathbf{L}_{\langle 2\rangle}$ and $3\,\mathbf{L}_{\langle 2\rangle}$ determine, essentially, the same line. The later could be said to pass through its points three times faster than the former. In order to get a different straight line, one has to change the attitude of $\mathbf{L}_{\langle 2\rangle}$. Thus, a parameterization must describe the attitude of a blade in order to be independent of the geometric interpretation of the parameterized subspace and also be capable of characterizing different instances of a given type of data alignment.

An arbitrary vector $\mathbf{a}$ can be expressed in a n-dimensional Euclidean space by $(n-1)$ angles and a scalar value. Assuming a basis $\{\mathbf{e}_i\}_{i=1}^{n}$ for $\mathbb{R}^n$, and taking $\mathbf{e}_n$ as reference

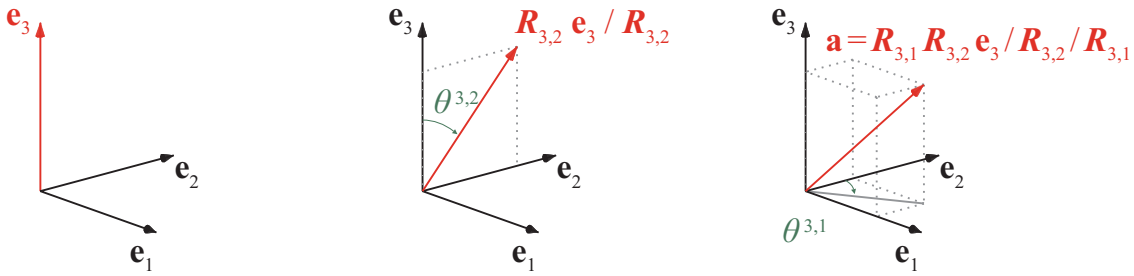


Figure 4.1: A sequence of two rotation operations in well defined planes describe the attitude of an arbitrary vector in $\mathbb{R}^3$. By assuming $\mathbf{e}_3$ as reference vector (a), the vector $\mathbf{a}$ in (c) can be computed from one rotation of $\theta^{3,2} = \pi/4$ radians on plane $\mathbf{e}_3 \wedge \mathbf{e}_2$ (b), followed by a rotation of $\theta^{3,1} = \pi/3$ radians on plane $\mathbf{e}_2 \wedge \mathbf{e}_1$ (c).

unit vector, $\mathbf{a}$ can be written as:

$$\mathbf{a} = \gamma \, \boldsymbol{S}_n \, \mathbf{e}_n \, / \, \boldsymbol{S}_n, \tag{4.1}$$

where

$$\boldsymbol{S}_n = \boldsymbol{R}_{n,1} \cdots \boldsymbol{R}_{n,n-2} \, \boldsymbol{R}_{n,n-1} \tag{4.2}$$

is a rotor encoding a sequence of rotations of $\theta^{n,j}$ radians on the unit planes $\mathbf{e}_{j+1} \wedge \mathbf{e}_j$, for

$$\boldsymbol{R}_{n,j} = \cos\left(\frac{\theta^{n,j}}{2}\right) - \sin\left(\frac{\theta^{n,j}}{2}\right) (\mathbf{e}_{j+1} \wedge \mathbf{e}_j) \tag{4.3}$$

(see Equation 3.22), and $j \in \{n-1, n-2, \cdots, 1\}$. Note that j is consistent with the order of the rotations applied to $\mathbf{e}_n$: the rotors $\boldsymbol{R}_{n,j}$ in Equation 4.2 are applied to $\mathbf{e}_n$ in Equation 4.1 from inside to outside of the sandwiching construction. Thus, the first rotation applied is $\boldsymbol{R}_{n,n-1}$, followed by $\boldsymbol{R}_{n,n-2}$, and so on. By assuming $\theta^{n,j} \in [-\pi/2, \pi/2)$, it is ensured that $\boldsymbol{S}_n \, \mathbf{e}_n \, / \, \boldsymbol{S}_n$ (Equation 4.1) is inside the hemisphere defined by $+\mathbf{e}_n$. Such a condition guarantees that the rotation angles encode $\mathbf{a}$'s attitude. In Equation 4.1, $\gamma \in \mathbb{R}$ characterizes the weight, and γ's sign characterizes the orientation of $\mathbf{a}$. A sequence of two rotation operations characterizing a vector in $\mathbb{R}^3$ is illustrated in Figure 4.1. In this example, $\mathbf{e}_3$ is the reference vector (Figure 4.1a), $\mathbf{e}_3 \wedge \mathbf{e}_2$ and $\mathbf{e}_2 \wedge \mathbf{e}_1$ are the rotation planes, $\theta^{3,2}$ and $\theta^{3,1}$ are, respectively, $\pi/4$ and $\pi/3$ radians (Figures 4.1b and 4.1c), and $\gamma = 1$.

In the notational convention of this work, $\boldsymbol{S}_n$ (Equation 4.2) encodes a sequence of $(n-1)$ rotations in an n-dimensional space (note the subscript n). $\boldsymbol{R}_{n,j}$ (Equation 4.3), in turn, represents one of such rotations on the plane $\mathbf{e}_{j+1} \wedge \mathbf{e}_j$.

The parameterization of vectors naturally extends to pseudovectors through the dual relationship between 1-dimensional and $(n-1)$-dimensional subspaces. By making the parameterized pseudovector $\mathbf{A}_{\langle n-1 \rangle} = \mathbf{a}^*$ and the reference unit blade $\mathbf{E}_{\langle n-1 \rangle} = \mathbf{e}_n^*$ ($\square^*$ is the dual operation, defined in Equation 3.15), Equation 4.1 becomes:

$$\mathbf{A}_{\langle n-1 \rangle} = \gamma \, \boldsymbol{S}_n \, \mathbf{E}_{\langle n-1 \rangle} \, / \, \boldsymbol{S}_n. \tag{4.4}$$

A parameterization that is equivalent for both 1-dimensional and $(n-1)$-dimensional subspaces is convenient due the possibility to build p-blades from these subspaces while using the smallest number of parameters. For instance, when $p < (n-p)$, spanning a

p-blade as the outer product of p vectors uses less parameters than spanning it as the regressive product of $(n-p)$ pseudovectors. However, when $p > (n-p)$, the best choice is to use $(n-p)$ pseudovectors. In the proposed parameterization the **reference unit subspace for p-blades** is built as:

$$\mathbf{E}_{\langle p \rangle} = \begin{cases} \bigwedge_{v \in \mathcal{V}} \mathbf{e}_v & \text{for } p \neq q \\ \bigvee_{v \in \mathcal{V}} \mathbf{e}_v^* & \text{for } p = q \end{cases}, \tag{4.5}$$

where $q = \max(p, n-p)$, $\bigwedge_{v \in \mathcal{V}}$ denotes the outer product of vectors $\mathbf{e}_v$, and $\bigvee_{v \in \mathcal{V}}$ is the regressive product of pseudovectors $\mathbf{e}_v^*$, for

$$\mathcal{V} = \{2\,(q+i) - n\}_{i=1}^{n-q}. \tag{4.6}$$

As for the parameterization of vectors and pseudovectors (Equations 4.1 and 4.4, respectively), the weight and orientation of an arbitrary blade $\mathbf{B}_{\langle p \rangle}$ are expressed by a scalar value (γ), while its **attitude is characterized by a set of rotations ($\boldsymbol{T}$)** applied to the reference blade $\mathbf{E}_{\langle p \rangle}$:

$$\mathbf{B}_{\langle p \rangle} = \gamma\, \boldsymbol{T}\, \mathbf{E}_{\langle p \rangle} \,/\, \boldsymbol{T}, \tag{4.7}$$

where

$$\boldsymbol{T} = \boldsymbol{S}_n\, \boldsymbol{S}_{n-2} \cdots \boldsymbol{S}_{2\,(q+1)-n}, \tag{4.8}$$

and $\boldsymbol{S}_v$ are rotors computed according to Equation 4.2. Note that the rotor $\boldsymbol{T}$ is computed as a sequence of rotors $\boldsymbol{S}_v$. Each one of such rotors $\boldsymbol{S}_v$ is related to the parameterization of the attitude of a reference vector (or pseudovector) used in the construction of the reference blade $\mathbf{E}_{\langle p \rangle}$ (Equation 4.5) for $\mathbf{B}_{\langle p \rangle}$. Note that $\mathbf{e}_v$, $\mathbf{e}_v^*$, and $\boldsymbol{S}_v$ have the same index v.

The geometric interpretation of blades is not affected by γ (*i.e.,* the weight and orientation of the blade), only by the rotation operations. Therefore, one can safely assume $\gamma = 1$ in Equation 4.7, leading to:

$$\mathbf{B}_{\langle p \rangle} = \boldsymbol{T}\, \mathbf{E}_{\langle p \rangle} \,/\, \boldsymbol{T}, \tag{4.9}$$

and define the parameterization of subspaces regarding

$$m = \sum_{v \in \mathcal{V}} (v-1) = p\,(n-p)$$

rotation angles $\theta^{v,j}$. Such a practice is consistent with existing HTs. For example, Duda and Hart's approach (DUDA; HART, 1972) retrieves lines having unit weight and positive orientation.

Equation 4.9 is regarded as **the proposed function for modeling subspaces** of dimensionality p, where $0 \leq p \leq n$. The m rotation angles related to the sequence of rotation operations in Equation 4.9 define a **parameter space** for p-blades:

$$\mathbb{P}^m = \{(\theta^1, \theta^2, \cdots, \theta^m) \mid \theta^t \in [-\pi/2, \pi/2)\}, \tag{4.10}$$

where each parameter vector $(\theta^1, \theta^2, \cdots, \theta^m) \in \mathbb{P}^m$ characterizes an instance of a p-blade, and θ^t is the rotation angle of the t-th rotation $\boldsymbol{R}_{v,j}$ (Equation 4.3) applied to $\mathbf{E}_{\langle p \rangle}$ in Equation 4.9, for $1 \leq t \leq m$. Notice that the double-index notation used so far (*i.e.,* n and j in $\theta^{n,j}$) has been simplified to single index (*i.e.,* t in θ^t, Equation 4.10) because it is more convenient for the definition of the parameter vector.

4.1 Avoiding Ambiguous Representations of Subspaces

Consecutive values for $v \in \mathcal{V}$ (Equation 4.6) used in Equations 4.5 and 4.8 are spaced two units apart. This avoids ambiguous sets of vectors/pseudovectors while defining $\mathbf{B}_{\langle p \rangle}$ (Equations 4.7 and 4.9), and leads to a more compact parameterization. The example shown in Figure 4.2 illustrates this. Note that the example is described in terms of pseudovectors. Therefore, it immediately extends to pairs of pseudovectors in spaces of any dimensionality and to vectors by duality. The construction of blades with arbitrary dimensionality is achieved by the successive application of this example on representational spaces with dimensionality n, $(n-2)$, $(n-4)$, and so on. For this example, consider the parameterization of a straight line in 3-dimensional base space under the homogeneous MOG. In such a case, the representational space has 4 dimensions (*i.e.,* $n = 3+1 = 4$) and the straight line ($\mathbf{B}_{\langle 2 \rangle}$) is a 2-blade. The homogeneous MOG in GA is analogous to using homogeneous coordinates in projective geometry. In this example, the basis vectors are $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4\}$ and $\mathbf{e}_4$ is geometrically interpreted as the point at the origin.

$\mathbf{B}_{\langle 2 \rangle}$ can be written in terms of the intersection (regressive product) of pseudovectors $\mathbf{F}_{\langle 3 \rangle}$ and $\mathbf{G}_{\langle 3 \rangle}$, geometrically interpreted as planes in Figure 4.2a:

$$\mathbf{B}_{\langle 2 \rangle} = \mathbf{F}_{\langle 3 \rangle} \vee \mathbf{G}_{\langle 3 \rangle}. \tag{4.11}$$

The choice of arbitrary pairs of planes can lead to ambiguous representations for $\mathbf{B}_{\langle 2 \rangle}$. For instance, by rotating $\mathbf{F}_{\langle 3 \rangle}$ and $\mathbf{G}_{\langle 3 \rangle}$ around the line $\mathbf{B}_{\langle 2 \rangle}$ in Figure 4.2a, one gets different pairs of planes, and hence different parameterizations for the attitude of $\mathbf{B}_{\langle 2 \rangle}$. Recall that rotations parameterizing the attitude of $\mathbf{B}_{\langle 2 \rangle}$ come from the parameters of the planes (pseudovectors) defining it (see the sequence of rotors $\boldsymbol{S}_v$ in Equation 4.8). In the proposed parameterization, such ambiguity is avoided by choosing $\mathbf{G}_{\langle 3 \rangle}$ and $\mathbf{F}_{\langle 3 \rangle}$ such that they satisfy some constraints:

1. $\mathbf{G}_{\langle 3 \rangle}$ is the plane whose smallest distance to $\mathbf{e}_4$ is the same as the smallest distance from $\mathbf{B}_{\langle 2 \rangle}$ to $\mathbf{e}_4$. In such a situation, $\mathbf{p}$ is the closest point to the origin $\mathbf{e}_4$ for both $\mathbf{B}_{\langle 2 \rangle}$ and $\mathbf{G}_{\langle 3 \rangle}$. It can be computed as the orthogonal projection of $\mathbf{e}_4$ onto $\mathbf{G}_{\langle 3 \rangle}$, and hence onto $\mathbf{B}_{\langle 2 \rangle}$.
2. $\mathbf{F}_{\langle 3 \rangle}$ is chosen as the plane passing through $\mathbf{B}_{\langle 2 \rangle}$ as well as through the point $\mathbf{q} = \mathbf{G}_{\langle 3 \rangle}^{-*}$. Note that the undual operation $\square^{-*}$ (Equation 3.17) makes $\mathbf{q}$ be orthogonal to $\mathbf{G}_{\langle 3 \rangle}$ in the 4-dimensional representational space. Therefore, it is guaranteed that $\mathbf{F}_{\langle 3 \rangle}$ includes one vector factor that is orthogonal to $\mathbf{G}_{\langle 3 \rangle}$.

By definition, $\mathbf{p}$ and $\mathbf{q}$ are orthogonal vectors in the representational space, because $\mathbf{p} \subset \mathbf{G}_{\langle 3 \rangle}$ and $\mathbf{q}$ is orthogonal to $\mathbf{G}_{\langle 3 \rangle}$. It is important to emphasize that the same two constraints can be imposed to pairs of pseudovectors in representational spaces of arbitrary dimensionality n. Also, that the actual geometrical interpretation of these pseudovectors can be cast as hyperplanes in homogeneous MOG, where $\mathbf{e}_n$ is geometrically interpreted as the point at the origin.

Now, let's replace $\mathbf{G}_{\langle 3 \rangle}$ in Equation 4.11 by the pseudovector parameterization in Equation 4.4:

$$\mathbf{B}_{\langle 2 \rangle} = \mathbf{F}_{\langle 3 \rangle} \vee \left(\beta \, \boldsymbol{S}_4 \, \mathbf{E}_{\langle 3 \rangle} \, / \, \boldsymbol{S}_4 \right). \tag{4.12}$$

where $\mathbf{E}_{\langle 3 \rangle} = \mathbf{e}_4^* = \mathbf{e}_4 \rfloor \mathbf{I}_{\langle 4 \rangle}^{-1} = -\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ is the reference blade for $\mathbf{G}_{\langle 3 \rangle}$. Note that $\boldsymbol{S}_4$ is computed from rotations (see Equation 4.2) in such a way that any vector in the 4-dimensional space is affected by it. Therefore, $\mathbf{F}_{\langle 3 \rangle}$ is also affected by $\boldsymbol{S}_4$. One can

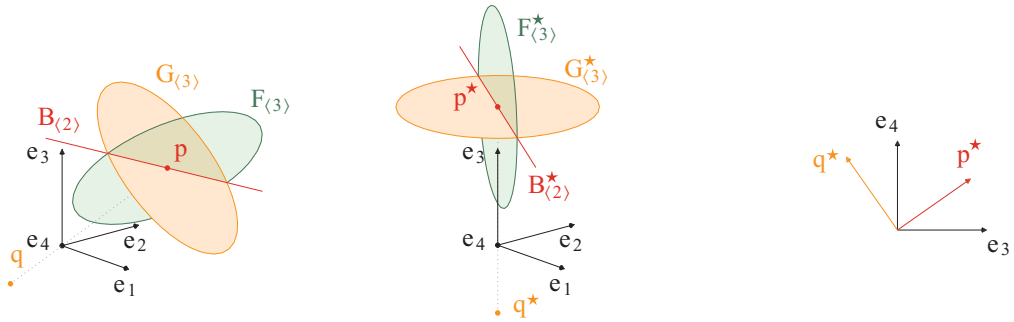


Figure 4.2: Pseudovectors used in the construction of a 2-blade in $\bigwedge \mathbb{R}^{3+1}$. By assuming the homogeneous MOG and a 3-dimensional base space: (a) $\mathbf{B}_{\langle 2\rangle}$ is the blade (interpreted as a straight line) resulting from the intersection of pseudovectors $\mathbf{F}_{\langle 3\rangle}$ and $\mathbf{G}_{\langle 3\rangle}$ (interpreted as planes). $\mathbf{G}_{\langle 3\rangle}$ defines the distance from $\mathbf{B}_{\langle 2\rangle}$ to the origin $\mathbf{e}_4$. $\mathbf{p}$ is the closest point to $\mathbf{e}_4$ for both $\mathbf{B}_{\langle 2\rangle}$ and $\mathbf{G}_{\langle 3\rangle}$. $\mathbf{F}_{\langle 3\rangle}$ includes $\mathbf{q} = \mathbf{G}^{-*}_{\langle 3\rangle}$, a vector that is orthogonal to $\mathbf{G}_{\langle 3\rangle}$ in the 4-dimensional representational space. (b) Configuration obtained after rolling back rotations $\boldsymbol{R}_{4,1}$ and $\boldsymbol{R}_{4,2}$ from the blades $\mathbf{p}$, $\mathbf{q}$, $\mathbf{B}_{\langle 2\rangle}$, $\mathbf{F}_{\langle 3\rangle}$, and $\mathbf{G}_{\langle 3\rangle}$. Here, the $\star$ symbol indicates that $\mathbf{A}^{\star}_{\langle k\rangle} = (\boldsymbol{R}^{-1}_{4,2}\,\boldsymbol{R}^{-1}_{4,1})\,\mathbf{A}_{\langle k\rangle} \,/\, (\boldsymbol{R}^{-1}_{4,2}\,\boldsymbol{R}^{-1}_{4,1})$, where $\mathbf{A}_{\langle k\rangle}$ is some blade in (a). (c) The same situation illustrated in (c) for $\mathbf{e}_3$, $\mathbf{e}_4$, $\mathbf{p}^{\star}$ and $\mathbf{q}^{\star}$, but disregarding $\mathbf{e}_1$ and $\mathbf{e}_2$.

write $\mathbf{F}_{\langle 3\rangle} = \boldsymbol{S}_4\,\mathbf{F}'_{\langle 3\rangle} \,/\, \boldsymbol{S}_4$ and replace it in Equation 4.12:

$$\mathbf{B}_{\langle 2\rangle} = \left(\boldsymbol{S}_4\,\mathbf{F}'_{\langle 3\rangle} \,/\, \boldsymbol{S}_4\right) \vee \left(\beta\,\boldsymbol{S}_4\,\mathbf{E}_{\langle 3\rangle} \,/\, \boldsymbol{S}_4\right).$$

From the structure preservation property of rotors (Equation 3.21):

$$\mathbf{B}_{\langle 2\rangle} = \beta\,\boldsymbol{S}_4\,\left(\mathbf{F}'_{\langle 3\rangle} \vee \mathbf{E}_{\langle 3\rangle}\right) \,/\, \boldsymbol{S}_4. \tag{4.13}$$

In Equation 4.13, $\boldsymbol{S}_4$ encodes a sequence of three rotation operations in well defined planes and $\mathbf{E}_{\langle 3\rangle}$ is the reference pseudovector for $\mathbf{G}_{\langle 3\rangle}$. Thus, three parameters of $\mathbf{B}_{\langle 2\rangle}$ have been defined from the rotations in $\boldsymbol{S}_4$. One must write $\mathbf{F}'_{\langle 3\rangle}$ in terms of Equation 4.4 in order to retrieve the one parameter that is missing (*i.e.,* the attitude of a 2-blade in $\bigwedge \mathbb{R}^4$ is expressed by $m = 2\,(4-2) = 4$ parameters). Figure 4.2b shows that, rolling back $\boldsymbol{R}_{4,1}$ and $\boldsymbol{R}_{4,2}$ (the latest transformations in $\boldsymbol{S}_4$) from blades in Figure 4.2a, one gets $\mathbf{p}^{\star} = (\boldsymbol{R}^{-1}_{4,2}\,\boldsymbol{R}^{-1}_{4,1})\,\mathbf{p} \,/\, (\boldsymbol{R}^{-1}_{4,2}\,\boldsymbol{R}^{-1}_{4,1})$ and $\mathbf{q}^{\star} = (\boldsymbol{R}^{-1}_{4,2}\,\boldsymbol{R}^{-1}_{4,1})\,\mathbf{q} \,/\, (\boldsymbol{R}^{-1}_{4,2}\,\boldsymbol{R}^{-1}_{4,1})$ in the space spanned by $\{\mathbf{e}_3, \mathbf{e}_4\}$. This situation is also illustrated in Figure 4.2c, where $\mathbf{e}_3$, $\mathbf{e}_4$, $\mathbf{p}^{\star}$ and $\mathbf{q}^{\star}$ are drawn in a 2-dimensional vector space, since the coordinates related to $\mathbf{e}_1$ and $\mathbf{e}_2$ are zero for all these elements. By also rolling back $\boldsymbol{R}_{4,3}$ (a rotation on the 2-blade $\mathbf{e}_4 \wedge \mathbf{e}_3$), all the transformations defining $\boldsymbol{S}_4$ are removed from $\mathbf{p}$ and $\mathbf{q}$. A rotation on $\mathbf{e}_4 \wedge \mathbf{e}_3$ applied to vectors in the space $\{\mathbf{e}_3, \mathbf{e}_4\}$ (such as $\mathbf{p}^{\star}$ and $\mathbf{q}^{\star}$) is interpreted in Figure 4.2b as a translation along the line through $\mathbf{e}_4$ with direction $\mathbf{e}_3$. As a result, $\boldsymbol{R}^{-1}_{4,3}\,\mathbf{q}^{\star} \,/\, \boldsymbol{R}^{-1}_{4,3}$ translates $\mathbf{q}^{\star}$ to the origin and makes it equal to $\mathbf{e}_4$ up to a scaling factor (as one would expect from Equation 4.1 and Figure 4.2c). Also, $\boldsymbol{R}^{-1}_{4,3}\,\mathbf{p}^{\star} \,/\, \boldsymbol{R}^{-1}_{4,3}$ translates $\mathbf{p}^{\star}$ to the infinity, until it becomes $\mathbf{e}_3$ up to a scaling factor. Since $\{\mathbf{p}, \mathbf{q}\} \subset \mathbf{F}_{\langle 3\rangle}$, we can state that $\{\mathbf{e}_3, \mathbf{e}_4\} \subset \mathbf{F}'_{\langle 3\rangle}$ and write

$$\mathbf{F}'_{\langle 3\rangle} = \mathbf{F}'_{\langle 1\rangle} \wedge \mathbf{e}_3 \wedge \mathbf{e}_4, \tag{4.14}$$

where $\mathbf{F}'_{\langle 1\rangle}$ is the weighted portion of $\mathbf{F}'_{\langle 3\rangle}$ that is enclosed in the space spanned by $\{\mathbf{e}_1, \mathbf{e}_2\}$.

By rolling back $\boldsymbol{S}_4$ from the blades in Figure 4.2a, $\mathbf{G}_{\langle 3\rangle}$ has become the reference subspace $\mathbf{E}_{\langle 3\rangle} = -\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$. According to Equation 4.14, $\mathbf{F}_{\langle 3\rangle}$ has become $\mathbf{F}'_{\langle 3\rangle}$, which can be factored in terms of the outer product as two well defined vector factors (*i.e.,* $\mathbf{e}_3$ and $\mathbf{e}_4$) and one free factor (*i.e.,* $\mathbf{F}'_{\langle 1\rangle}$). By definition, $\mathbf{F}'_{\langle 1\rangle}$ is orthogonal to $\mathbf{e}_3$ and $\mathbf{e}_4$. Thus, it must be in $\mathbb{R}^2$. Using the pseudovector parameterization from Equation 4.4 over $\mathbf{F}'_{\langle 1\rangle}$, Equation 4.14 becomes:

$$\mathbf{F}'_{\langle 3\rangle} = \left(\alpha\, \boldsymbol{S}_2\, \mathbf{E}'_{\langle 1\rangle} \,/\, \boldsymbol{S}_2\right) \wedge \mathbf{e}_3 \wedge \mathbf{e}_4, \tag{4.15}$$

where $\mathbf{E}'_{\langle 1\rangle} = \mathbf{e}_2 \lrcorner \mathbf{I}^{-1}_{\langle 2\rangle} = -\mathbf{e}_1$ is the reference blade for $\mathbf{F}'_{\langle 1\rangle}$, a pseudovector in the $\{\mathbf{e}_1, \mathbf{e}_2\}$ space, and $\mathbf{I}_{\langle 2\rangle}$ is the pseudoscalar of such bidimensional space. Replacing Equation 4.15 in Equation 4.13:

$$\mathbf{B}_{\langle 2\rangle} = \beta\, \boldsymbol{S}_4 \left(\left(\left(\alpha\, \boldsymbol{S}_2\, \mathbf{E}'_{\langle 1\rangle} \,/\, \boldsymbol{S}_2\right) \wedge \mathbf{e}_3 \wedge \mathbf{e}_4\right) \vee \mathbf{E}_{\langle 3\rangle}\right) \,/\, \boldsymbol{S}_4. \tag{4.16}$$

Note that rotation planes from $\boldsymbol{S}_2$ do not affect subspace $\mathbf{e}_3 \wedge \mathbf{e}_4$ nor $\mathbf{E}_{\langle 3\rangle}$, because such rotation planes are orthogonal to the former and they are contained by the latter. As a result, Equation 4.16 can be rewritten as

$$\mathbf{B}_{\langle 2\rangle} = \gamma\, \left(\boldsymbol{S}_4\, \boldsymbol{S}_2\right) \left(\left(\mathbf{E}'_{\langle 1\rangle} \wedge \mathbf{e}_3 \wedge \mathbf{e}_4\right) \vee \mathbf{E}_{\langle 3\rangle}\right) \,/\, \left(\boldsymbol{S}_4\, \boldsymbol{S}_2\right). \tag{4.17}$$

Since $\mathbf{E}'_{\langle 1\rangle} \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 = -\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 = \mathbf{e}^*_2$ and $\mathbf{E}_{\langle 3\rangle} = \mathbf{e}^*_4$ (from Equation 4.12), Equation 4.17 can be simplified to

$$\mathbf{B}_{\langle 2\rangle} = \gamma\, \left(\boldsymbol{S}_4\, \boldsymbol{S}_2\right) \left(\mathbf{e}^*_2 \vee \mathbf{e}^*_4\right) \,/\, \left(\boldsymbol{S}_4\, \boldsymbol{S}_2\right). \tag{4.18}$$

Recall that reference pseudovectors for $\mathbf{F}_{\langle 3\rangle}$ and $\mathbf{G}_{\langle 3\rangle}$ are the dual representations of vectors $\mathbf{e}_2$ and $\mathbf{e}_4$, respectively (as one would expect from Equation 4.5, for $\mathcal{V} = \{2, 4\}$ and $p = q = 2$), and $\gamma = \alpha\, \beta$.

In Equation 4.18, $\boldsymbol{S}_2$ and $\boldsymbol{S}_4$ describe two sequences of rotations. The former consists of 1 rotation. It is similar to the case depicted in Equation 4.2, but in a dimensionality lower than $n = 4$. The latter consists of 3 rotations, exactly like in Equation 4.2. Together, the 4 rotation angles describe the attitude of $\mathbf{B}_{\langle 2\rangle}$.

4.2 Rotation Angles as a Coordinate Chart for the Grassmannian

The **Grassmannian** $G(k,n)$ is the set of all k-dimensional linear subspaces of a vector space $\mathbb{R}^n$ (HARRIS, 1992). In GA, the multivector representation of k-blades resides in $\bigwedge^k \mathbb{R}^n$, *i.e.,* the portion of the multivector space $\bigwedge \mathbb{R}^n$ with k-dimensional basis elements (Section 3.2). For example, in Equation 3.1, the basis of $\bigwedge^2 \mathbb{R}^3$ is $\{\mathbf{e}_1 \wedge \mathbf{e}_2, \mathbf{e}_1 \wedge \mathbf{e}_3, \mathbf{e}_2 \wedge \mathbf{e}_3\}$. The linear combination of basis elements in $\bigwedge^k \mathbb{R}^n$ is called a k-vector. However, an arbitrary k-vector is not necessarily a k-blade. The k-blades are the k-vector which can be factored in terms of the outer product of k linearly independent vectors. Thus, $G(k,n)$ corresponds to a subset of k-vectors (*i.e.,* the k-blades) in $\bigwedge^k \mathbb{R}^n$.

The Grassmannian defines a projective variety of dimension $k\,(n-k)$ in the $\binom{n}{k}$-dimensional space of $\bigwedge^k \mathbb{R}^n$ (HARRIS, 1992). Therefore, an arbitrary k-dimensional subspace requires at least $k\,(n-k)$ coordinates to be addressed in such variety. By choosing a reference subspace one may define an open affine covering $\mathbb{A}^{k\,(n-k)}$ for $G(k,n)$. The covering is open because the k-dimensional subspaces orthogonal to the reference are not

properly represented in the affine space $\mathbb{A}^{k\,(n-k)}$ (*i.e.,* they reside at infinity). The remaining k-dimensional subspaces in $G(k,n)$, on the other hand, are represented uniquely as points in $\mathbb{A}^{k\,(n-k)}$, where the reference subspace is related to the point at the origin. To see this in coordinates, let's assume the subspace spanned by $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k\}$ as the reference. It follows that any k-dimensional subspace in the open affine covering of $G(k,n)$ may be represented as the row space of a unique matrix of the form:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & \alpha^{1,1} & \alpha^{1,2} & \cdots & \alpha^{1,n-k} \\ 0 & 1 & \cdots & 0 & \alpha^{2,1} & \alpha^{2,2} & \cdots & \alpha^{2,n-k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & \alpha^{k,1} & \alpha^{k,2} & \cdots & \alpha^{k,n-k} \end{pmatrix}, \tag{4.19}$$

where the entries $\alpha^{i,j}$ define a location in $\mathbb{A}^{k\,(n-k)}$ (HARRIS, 1992). Thus, a subspace may be mapped from a point in $\mathbb{A}^{k\,(n-k)}$ to a blade $\mathbf{B}_{\langle k \rangle} \in \bigwedge^k \mathbb{R}^n$ through:

$$\mathbf{B}_{\langle k \rangle} = \bigwedge_{i=1}^{k} \left(\mathbf{e}_i + \sum_{j=1}^{n-k} \left(\alpha^{i,j}\, \mathbf{e}_{k+j} \right) \right).$$

A blade $\mathbf{B}_{\langle k \rangle}$ may be mapped from $\bigwedge^k \mathbb{R}^n$ to $\mathbb{A}^{k\,(n-k)}$ by decomposing $\mathbf{B}_{\langle k \rangle}$ into vector factors and, in turn, computing the row reduced echelon form of its $k \times n$ matrix representation. It leads to Equation 4.19 when $\mathbf{B}_{\langle k \rangle}$ is not orthogonal to $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_k$.

The parameter space $\mathbb{P}^m$ (Equation 4.10) introduced in this chapter provides a coordinate chart for $G(p,n)$. In such a coordinate system, a p-dimensional subspace is addressed by a set of $p\,(n-p)$ rotation angles in the $[-\pi/2, \pi/2)$ range (*i.e.,* the parameter vector). In contrast to the open affine covering $\mathbb{A}^{k\,(n-k)}$ of $G(k,n)$, the proposed parameterization can represent *all* the p-dimensional linear subspaces in $\bigwedge^p \mathbb{R}^n$. The drawback of the parameterization is that, for some p-blades, the mapping between $\mathbb{P}^m$ and $\bigwedge^p \mathbb{R}^n$ is not unique. It follows that all parameter vector in $\mathbb{P}^m$ map to a single p-blade in $\bigwedge^p \mathbb{R}^n$, but there are p-blades in $\bigwedge^p \mathbb{R}^n$ that map to an infinite set of parameter vectors in $\mathbb{P}^m$.

One example of many parameter vectors in $\mathbb{P}^m$ characterizing the same blade in $\bigwedge^p \mathbb{R}^n$ is the parameterization of the vector $\mathbf{e}_3 \in \bigwedge^1 \mathbb{R}^3$. In such a case, $p = 1$, $n = 3$ and $m = 1\,(3-1) = 2$, leading to:

$$\mathbb{P}^2 = \{(\theta^{3,2}, \theta^{3,1}) \mid \theta^{3,2}, \theta^{3,1} \in [-\pi/2, \pi/2)\}.$$

From Equation 4.5, the reference blade is $\mathbf{E}_{\langle 1 \rangle} = \mathbf{e}_3$. According to Equation 4.2, the two rotations applied to $\mathbf{E}_{\langle 1 \rangle}$ are, respectively, on planes $\mathbf{e}_3 \wedge \mathbf{e}_2$ and $\mathbf{e}_2 \wedge \mathbf{e}_1$. Notice that the parameterized blade ($\mathbf{e}_3$) is equal to the reference blade. So the rotation on $\mathbf{e}_3 \wedge \mathbf{e}_2$ (*i.e.,* from $\mathbf{e}_3$ to $\mathbf{e}_2$) must be of *zero* radians, leading to $\theta^{3,2} = 0$. The second rotation is on a plane orthogonal $\mathbf{e}_3$. As a result, $\theta^{3,1}$ may assume any value in $[-\pi/2, \pi/2)$. Thus, $\mathbf{e}_3$ may be characterized by any parameter vector in the form $(0, \theta^{3,1})$ for $\theta^{3,1} \in [-\pi/2, \pi/2)$. Such an ambiguity (related to the so-called *gimbal lock*) is inherent to the description of the attitude of a rigid body (*i.e.,* a body in which the relative position of all its points is constant) in some Euclidean space though a set of rotations (HOFFMAN; RAFFENETTI; RUEDENBERG, 1972). Fortunately, the gimbal lock does not compromise the proposed subspace detection framework. Chapters 5 and 6 describe general procedures for mapping subspaces of arbitrary dimensionality to $\mathbb{P}^m$ even in the presence of gimbal lock.

4.3 The Proposed Parameterization as a Subset of Euler Angles

Euler (1765) showed that the orientation of a rigid body in 3-dimensional Euclidean space requires exactly three parameters to be described. When these parameters are defined in the form of angles they are called *the Euler angles*. Hoffman *et al.* (1972) had generalized the Euler angles to n-dimensional spaces. They presented an algorithm whereby n-dimensional orthogonal matrices, and hence rigid bodies with n-dimensional coordinate systems, can be represented in terms of $n\,(n-1)/2$ independent parameters, for $n \geq 2$. Appendix C presents the derivations proposed by Hoffman *et al.* (1972), originally with linear algebra, converted here to GA notation. In GA, it is equivalent to setting the parameterization/decomposition of rotors in terms of generalized Euler angles.

The parameterization proposed in the current chapter targets a different problem than (HOFFMAN; RAFFENETTI; RUEDENBERG, 1972): to express the attitude of p-blades through a set of rotation angles. In such a case, the number of parameters required (*i.e.,* $p\,(n-p)$) is smaller than the number of parameters for describing a n-dimensional coordinate system with respect to a reference basis. The only exceptions is for vectors in $\mathbb{R}^2$, where a single parameter is required in both approaches. However, the parameterization of p-blades can be cast as the parameterization of the rotor $\boldsymbol{T}$ that transforms a reference blade $\mathbf{E}_{\langle p \rangle}$ into a given subspace (Equation 4.9). In such a case, one may set an n-dimensional coordinate system where most of the rotation operations (from Euler angles) happen on planes orthogonal to, or contained by, the reference subspace. The remaining rotations (*i.e.,* the subset of generalized Euler angles that affect $\mathbf{E}_{\langle p \rangle}$) are equivalent to the proposed parameterization.

4.4 Discussion

The function for modeling p-dimensional linear subspaces in an n-dimensional total space was introduced in this chapter, as well as the parameter space defined from such a function. The idea behind the proposed parameterization is conceptually quite simple: the attitude of an arbitrary p-blade can be expressed by a sequence of $p\,(n-p)$ rotation operations applied to a canonical subspace with the same dimensionality, while its weight and orientation can be expressed by a scalar value. Since the geometric interpretation of blades is related only to its attitude, the rotation angles are sufficient to define a parameter space where every point characterizes a p-dimensional linear subspace.

In contrast to model functions from conventional HTs (HOUGH, 1959, 1962; DUDA; HART, 1972; O'GORMAN; CLOWES, 1973; KIMME; BALLARD; SKLANSKY, 1975; SKLANSKY, 1978; BENNETT; BURRIDGE; SAITO, 1999; ACHTERT et al., 2008), the proposed model function is independent of the type of data alignment that is being detected, as well as of the type of data comprising the input dataset. It depends only on the dimensionality of the subspace representing the intended data alignment (p) and the dimensionality of the representational space (n). Such a feature is key for the definition of a general framework for data alignment detection. It allows the development of general procedures for mapping arbitrary types of data to the parameter space, and hence for the design of generally applicable voting schemes, such as the ones presented in Chapters 5 and 6.

In the proposed subspace detection framework, an accumulator array is created as the discrete representation of the parameter space for p-blades. Therefore, another desirable feature of such parameter space is to be representable in a finite range of parameter values.

By definition, the rotation angles used as parameters are restricted to the $[-\pi/2, \pi/2)$ domain. The use of the open affine covering of the Grassmannian, by contrast, would require handling coordinates in the $(-\infty, +\infty)$ range, with the further drawback of not being able to properly represent all the p-dimensional subspaces. A discussion on the discretization of the proposed parameter space is presented in the next chapter.

5 VOTING PROCESS FOR INPUT SUBSPACES

The voting process is composed of two stages: (i) mapping the entries in the input dataset to a (continuous) parameter space (using the function described in Figure 5.2), and (ii) spreading the "votes" on the bins of the accumulator array (*i.e.,* the discrete representation of the parameter space) according to such a mapping. Chapter 4 presented a parameter space ($\mathbb{P}^m$) for characterizing p-blades, *i.e.,* the linear subspaces that represent the intended type of data alignment. The current chapter presents a mapping procedure that takes each r-dimensional subspace $\mathbf{X}_{\langle r \rangle}$ in the input dataset (for $0 \leq r \leq n$) and identifies the parameters (coordinates in $\mathbb{P}^m$) of all p-blades related to it. This chapter also describes a voting procedure that uses this mapping scheme.

The proposed mapping procedure is based on three key observations. The *first observation* is that the dimensionality of an arbitrary input blade $\mathbf{X}_{\langle r \rangle}$ defines a **containment relationship** between $\mathbf{X}_{\langle r \rangle}$ and $\mathbf{C}_{\langle p \rangle} \in \mathcal{C}$ (*i.e.,* $\mathbf{X}_{\langle r \rangle}$ is contained or it contains $\mathbf{C}_{\langle p \rangle}$), where $\mathcal{C}$ is the set of all p-blades related to $\mathbf{X}_{\langle r \rangle}$. In Chapter 4, $\mathbf{B}_{\langle p \rangle}$ denotes an arbitrary p-blade. In the current chapter, $\mathbf{C}_{\langle p \rangle}$ has a more restricted definition, denoting a p-blade related to $\mathbf{X}_{\langle r \rangle}$.

Since p-blades are expressed as orthogonal transformations applied to a reference blade $\mathbf{E}_{\langle p \rangle}$ (Equation 4.9), the containment relationships between $\mathbf{X}_{\langle r \rangle}$ and $\mathbf{C}_{\langle p \rangle}$ can be extended through the sequence of transformations:

$$\begin{cases} \mathbf{X}^{(t)}_{\langle r \rangle} \subseteq \mathbf{C}^{(t)}_{\langle p \rangle} & \text{for } r \leq p \\ \mathbf{X}^{(t)}_{\langle r \rangle} \supseteq \mathbf{C}^{(t)}_{\langle p \rangle} & \text{for } r \geq p \end{cases}, \tag{5.1}$$

where

$$\mathbf{X}^{(t)}_{\langle r \rangle} = \boldsymbol{R}^{-1}_{t+1}\, \mathbf{X}^{(t+1)}_{\langle r \rangle} \,/\, \boldsymbol{R}^{-1}_{t+1} \tag{5.2}$$

and

$$\mathbf{C}^{(t)}_{\langle p \rangle} = \boldsymbol{R}_t\, \mathbf{C}^{(t-1)}_{\langle p \rangle} \,/\, \boldsymbol{R}_t$$

for $\mathbf{X}^{(m+1)}_{\langle r \rangle} = \mathbf{X}_{\langle r \rangle}$, $\mathbf{C}^{(0)}_{\langle p \rangle} = \mathbf{E}_{\langle p \rangle}$, $\boldsymbol{R}_{m+1} = 1$, and $1 \leq t \leq m$. Here, $\boldsymbol{R}_t$ encodes the t-th rotation applied to $\mathbf{E}_{\langle p \rangle}$. In Chapter 4, a double-index notation (*e.g.,* v and j in $\boldsymbol{R}_{v,j}$) was used to emphasize that rotations $\boldsymbol{R}_{v,j}$ are related to a rotor $\boldsymbol{S}_v$, and hence to a reference vector $\mathbf{e}_v$ or pseudovector $\mathbf{e}^*_v$. In this chapter the notation has been changed to a single index (t) because it is more convenient for the following derivations. Thus, one can think of the model function (Equation 4.9) by replacing the rotor $\boldsymbol{T}$ by its component rotors $\boldsymbol{S}_v$ and, in turn, by replacing each $\boldsymbol{S}_v$ by its component rotors $\boldsymbol{R}_{v,j}$, leading to:

$$\mathbf{C}_{\langle p \rangle} = \boldsymbol{R}_m \left(\boldsymbol{R}_{m-1} \cdots \left(\boldsymbol{R}_2 \left(\boldsymbol{R}_1 \mathbf{E}_{\langle p \rangle} \,/\, \boldsymbol{R}_1\right) \,/\, \boldsymbol{R}_2\right) \cdots \,/\, \boldsymbol{R}_{m-1}\right) \,/\, \boldsymbol{R}_m. \tag{5.3}$$

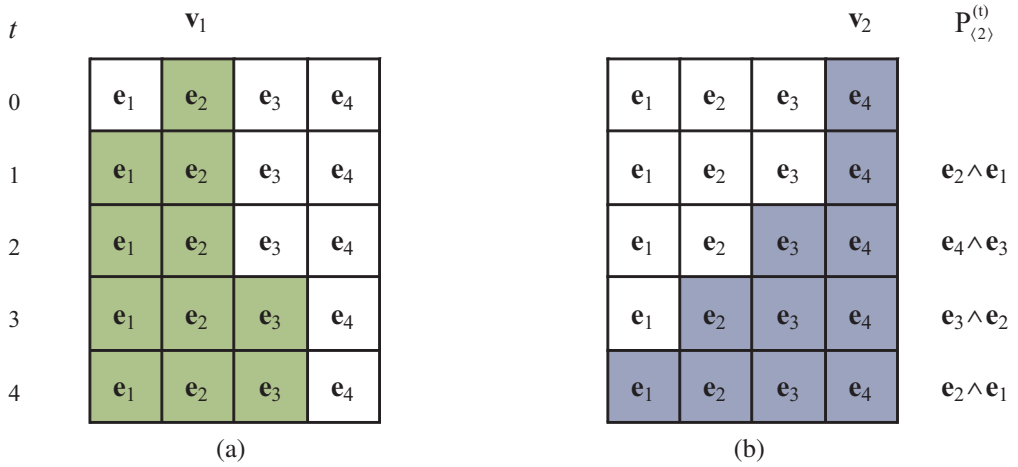


Figure 5.1: Trees of possibilities for $\mathbf{v}_1 = \mathbf{e}_2$ (a) and $\mathbf{v}_2 = \mathbf{e}_4$ (b) in a 4-dimensional representational space ($n = 4$ and $p = 2$). Vectors $\mathbf{e}_2$ and $\mathbf{e}_4$ at the first row are the vectors spanning a reference blade. Rows $t = 1$ to $t = 4$ represent the rotation of reference vectors on the planes $\mathbf{e}_2 \wedge \mathbf{e}_1$, $\mathbf{e}_4 \wedge \mathbf{e}_3$, $\mathbf{e}_3 \wedge \mathbf{e}_2$, and $\mathbf{e}_2 \wedge \mathbf{e}_1$, respectively. From Equations 4.8 and 4.2 one should note that the first rotation plane comes from the single rotation defining $\boldsymbol{S}_2$, and the other three planes came from the rotations defining $\boldsymbol{S}_4$. The colored squares at each row correspond to the space of possibilities (Equation 5.5) related to vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively, after applying the rotations up to a given row.

In the adopted notation, the superscript in parentheses (*e.g.,* t in $\mathbf{X}^{(t)}_{\langle r \rangle}$ and $\mathbf{C}^{(t)}_{\langle p \rangle}$, Equation 5.1) indicates that the element is an intermediate variable in the context of the t-th rotation operation.

The *second observation* is related to the rotation of basis vectors spanning $\mathbf{E}_{\langle p \rangle}$. They comprise the set $\mathcal{E}$:

$$\mathcal{E} = \begin{cases} \{\mathbf{e}_v\}_{v \in \mathcal{V}} & \text{for } p \neq q \\ \{\mathbf{e}_v\}_{v \in \mathcal{V}} \backslash \{\mathbf{e}_i\}_{i=1}^{n} & \text{for } p = q \end{cases}, \tag{5.4}$$

where $\mathcal{A} \backslash \mathcal{B}$ denotes the relative complement of $\mathcal{A}$ in $\mathcal{B}$. **As the rotation operations are applied to vectors $\mathbf{v}_l \in \mathcal{E}$ (for $1 \leq l \leq |\mathcal{E}|$, where $|\mathcal{E}| = p$ denotes the cardinality of $\mathcal{E}$), the dimensionality of the regions of $\mathbb{R}^n$ that can be reached by vectors $\mathbf{v}_l$ increases**. These regions are called **spaces of possibilities**. Figure 5.1 illustrates the **tree of possibilities** of reference vectors $\mathbf{v}_1 = \mathbf{e}_2$ (Figure 5.1a) and $\mathbf{v}_2 = \mathbf{e}_4$ (Figure 5.1b), for $\mathcal{E} = \{\mathbf{e}_2, \mathbf{e}_4\}$ and $n = 4$. In Figure 5.1, each row of the grid is related to a rotation on the plane $\mathbf{P}^{(t)}_{\langle 2 \rangle}$. The values of index t are indicated on the left side, and the rotation planes (*e.g.,* $\mathbf{e}_i \wedge \mathbf{e}_j$) are indicated on the right. The set of colored squares at each row corresponds to the region that can be reached by the reference vector after applying the rotations up to the corresponding row (*i.e.,* the space of possibilities $\mathbf{F}^{(t)}_l$, defined in Equation 5.5). For instance, after applying the three first rotations to $\mathbf{v}_1 = \mathbf{e}_2$, it can becomes a vector in the space spanned by $\mathbf{F}^{(3)}_1 = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$.

The spaces of possibilities are computed as:

$$\mathbf{F}^{(t)}_l = \begin{cases} \mathbf{F}^{(t-1)}_l \cup \mathbf{P}^{(t)}_{\langle 2 \rangle} & \text{for grade}\,(\mathbf{F}^{(t-1)}_l \cap \mathbf{P}^{(t)}_{\langle 2 \rangle}) = 1 \\ \mathbf{F}^{(t-1)}_l & \text{otherwise} \end{cases}, \tag{5.5}$$

where $\mathbf{F}^{(t)}_l$ is the space reachable by vector $\mathbf{v}_l \in \mathcal{E}$ after the application of the first t ro-

tations. Therefore, $\mathbf{F}_l^{(0)} = \mathbf{v}_l$. $\mathbf{P}_{\langle 2 \rangle}^{(t)}$ is the plane where the t-th rotation happens, $\cup$ and $\cap$ denote, respectively, the join and the meet operations (Section 3.7), and the grade function retrieves the dimensionality of a subspace. Notice that Equation 5.5 determines how a tree of possibilities grows: $\mathbf{F}_l^{(t)}$ will always be equal to or have one more dimension than its predecessor, $\mathbf{F}_l^{(t-1)}$. $\mathbf{F}_l^{(t)}$ and $\mathbf{F}_l^{(t-1)}$ are equal when the rotation plane $\mathbf{P}_{\langle 2 \rangle}^{(t)}$ is contained by or is orthogonal to $\mathbf{F}_l^{(t-1)}$. For instance, in Figure 5.1a, $\mathbf{F}_1^{(4)} = \mathbf{F}_1^{(3)} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ because a rotation on $\mathbf{P}_{\langle 2 \rangle}^{(4)} = \mathbf{e}_2 \wedge \mathbf{e}_1$ does not affect $\mathbf{F}_1^{(3)}$. By the same token, $\mathbf{F}_l^{(t)}$ will have one more dimension than $\mathbf{F}_l^{(t-1)}$ when only one of the dimensions of $\mathbf{F}_l^{(t-1)}$ is affected by the t-th rotation operation. For example, $\mathbf{F}_2^{(3)} = \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4$ in Figure 5.1b. By applying a rotation on plane $\mathbf{P}_{\langle 2 \rangle}^{(4)} = \mathbf{e}_2 \wedge \mathbf{e}_1$, the portion $\mathbf{e}_2$ in $\mathbf{F}_2^{(3)}$ can reach any vector in the $\mathbf{e}_2 \wedge \mathbf{e}_1$ plane, leading to $\mathbf{F}_2^{(4)} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4$.

The spaces of possibilities are important because they impose a set of restrictions for the vector factors of a given blade $\mathbf{C}_{\langle p \rangle} \in \mathcal{C}$. By writing $\mathbf{C}_{\langle p \rangle}$ in terms of the outer product of its orthonormal vector factors $\mathbf{c}_l$:

$$\mathbf{C}_{\langle p \rangle} = \mathbf{c}_1 \wedge \mathbf{c}_2 \wedge \cdots \wedge \mathbf{c}_p,$$

one should notice that a parameter vector $(\theta^1, \theta^2, \cdots, \theta^m) \in \mathbb{P}^m$ defines the transformation of each $\mathbf{v}_l \in \mathcal{E}$ into a vector $\mathbf{c}_l^{(t)} \subseteq \mathbf{F}_l^{(t)}$, where

$$\mathbf{c}_l^{(t)} = \boldsymbol{R}_t \, \mathbf{c}_l^{(t-1)} \, / \, \boldsymbol{R}_t,$$

for $\mathbf{c}_l^{(0)} = \mathbf{v}_l$ and $\mathbf{c}_l^{(m)} = \mathbf{c}_l$. Thus, a given $\mathbf{c}_l^{(t)}$ is "confined" in a well defined tree of possibilities.

The *third observation* is that rotations do not commute. Therefore, **one needs to respect the sequence of rotations while computing the parameter vectors** of blades in $\mathcal{C}$. Since $\mathbf{X}_{\langle r \rangle}$ and $\mathbf{E}_{\langle p \rangle}$ are the only data available to compute the elements in $\mathcal{C}$, **the proposed approach calculates the parameter vectors starting from the last to the first** θ^t (*i.e.,* from θ^m to θ^1). Thus, using $\mathbf{X}_{\langle r \rangle}^{(t)}$ (Equation 5.2) as input, the t-th rotation angle is computed. In the case of Figure 5.1, $\mathbf{X}_{\langle r \rangle}^{(4)} = \mathbf{X}_{\langle r \rangle}$ is related to the last row of the trees of possibilities. By computing the last parameter one is able to find the rotation that takes $\mathbf{X}_{\langle r \rangle}$ into the previous row (*i.e.,* one finds $\mathbf{X}_{\langle r \rangle}^{(3)}$) and so on, until all the θ^t values have been computed, and finally reach $\mathbf{X}_{\langle r \rangle}^{(0)}$. $\mathbf{X}_{\langle r \rangle}^{(0)}$ is then related to the canonical reference $\mathbf{E}_{\langle p \rangle}$.

5.1 Mapping Procedure for $r \geq p$

The procedure for mapping a given input blade $\mathbf{X}_{\langle r \rangle}$ to parameter space $\mathbb{P}^m$ is presented in Figure 5.2. The algorithm assumes that $r \geq p$. The case involving $r \leq p$ is discussed in Section 5.2. By making $r \geq p$, the condition depicted in Equation 5.1 and the second observation guarantees the existence of vectors $\mathbf{c}_l^{(t)} \subseteq (\mathbf{X}_{\langle r \rangle}^{(t)} \cap \mathbf{F}_l^{(t)})$ for all $1 \leq l \leq |\mathcal{E}|$. This holds because $\mathbf{C}_{\langle p \rangle}^{(t)}$ can be factorized as

$$\mathbf{C}_{\langle p \rangle}^{(t)} = \mathbf{c}_1^{(t)} \wedge \mathbf{c}_2^{(t)} \wedge \cdots \wedge \mathbf{c}_p^{(t)}, \tag{5.6}$$

where each factor $\mathbf{c}_l^{(t)}$ is related to a space of possibilities $\mathbf{F}_l^{(t)}$, and $\mathbf{X}_{\langle r \rangle}^{(t)}$ includes the entire blade $\mathbf{C}_{\langle p \rangle}^{(t)}$. In Figure 5.1, it means that the transformed input blade $\mathbf{X}_{\langle r \rangle}^{(t)}$ will always share

Require: An input r-blade $\mathbf{X}_{\langle r \rangle}$

1: $\mathcal{P}^{(m)} \leftarrow \left\{(\mathbf{X}_{\langle r \rangle}, \varnothing)\right\}$
2: **for** $t = m$ **down to** 1 **do**
3: Let $\mathbf{P}^{(t)}_{\langle 2 \rangle}$ be the rotation plane of the t-th rotation applied to $\mathbf{E}_{\langle p \rangle}$ in Equation 5.3
4: $\mathcal{P}^{(t-1)} \leftarrow \varnothing$
5: **for all** $(\mathbf{X}^{(t)}_{\langle r \rangle}, \Theta^{(t)}) \in \mathcal{P}^{(t)}$ **do**
6: $\mathcal{T} \leftarrow CalculateParameter(\mathbf{X}^{(t)}_{\langle r \rangle})$
7: $\mathcal{P}^{(t-1)} \leftarrow \mathcal{P}^{(t-1)} \cup \left\{(\boldsymbol{R}_t^{-1}\,\mathbf{X}^{(t)}_{\langle r \rangle} \,/\, \boldsymbol{R}_t^{-1}, (\theta^t, \Theta^{(t)}_1, \Theta^{(t)}_2, \cdots, \Theta^{(t)}_{m-t})) \mid \right.$
$\left. \boldsymbol{R}_t = \cos\left(\frac{\theta^t}{2}\right) - \sin\left(\frac{\theta^t}{2}\right) \mathbf{P}^{(t)}_{\langle 2 \rangle}, \text{ and } \theta^t \in \mathcal{T}\right\}$
8: **end for**
9: **end for**
10: **return** $\left\{\Theta^{(0)} \mid (\mathbf{X}^{(0)}_{\langle r \rangle}, \Theta^{(0)}) \in \mathcal{P}^{(0)}\right\}$

Figure 5.2: The algorithm used to map an input r-blade $\mathbf{X}_{\langle r \rangle}$ to $\mathbb{P}^m$ (Equation 4.10). The procedure returns a set of parameter vectors $\Theta^{(0)} \in \mathbb{P}^m$ characterizing the p-blades that are contained by $\mathbf{X}_{\langle r \rangle}$.

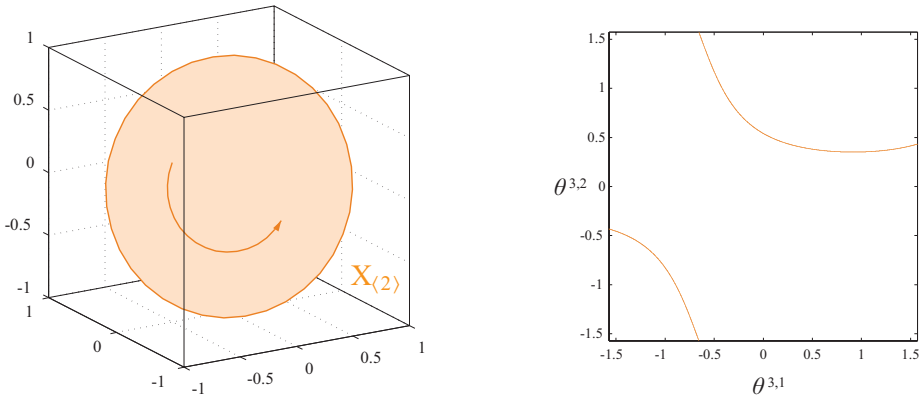
at least one vector factor with each space of possibilities at each row of the trees (not necessarily a basis vector, but a vector in the space of possibilities). At the first row of Figure 5.1, $\mathbf{X}^{(0)}_{\langle r \rangle}$ must include $\mathbf{v}_1 = \mathbf{e}_2$ and $\mathbf{v}_2 = \mathbf{e}_4$.

In its first step (Figure 5.2, line 1), the mapping procedure initializes a set $\mathcal{P}^{(m)}$ with a 2-tuple comprised by the input blade $\mathbf{X}_{\langle r \rangle}$ and an empty set ($\varnothing$) denoting that no parameter was calculated yet. At each iteration (lines 2 to 9, for $t = m$ down to 1) the 2-tuples in $\mathcal{P}^{(t)}$ are processed and a new set $\mathcal{P}^{(t-1)}$ is created. For each 2-tuples in $\mathcal{P}^{(t)}$ (inner loop, lines 5 to 8), the procedure $CalculateParameter$ (defined in Figure 5.4) calculates the parameter θ^t for the $\mathbf{C}^{(t)}_{\langle p \rangle}$ blades related to $\mathbf{X}^{(t)}_{\langle r \rangle}$. Recall that a given input blade $\mathbf{X}_{\langle r \rangle}$ can lead to one or more parameter vectors, and hence one or more blades $\mathbf{C}_{\langle p \rangle} \in \mathcal{C}$. It depends on how many p-blades are related to $\mathbf{X}_{\langle r \rangle}$. Calculating the t-th parameter implies identifying the $\mathbf{c}^{(t)}_l$ vectors in Equation 5.6, for the current t, and computing the $\boldsymbol{R}_t$ that ensures the existence of vectors

$$\mathbf{c}^{(t-1)}_l = \boldsymbol{R}_t^{-1}\,\mathbf{c}^{(t)}_l \,/\, \boldsymbol{R}_t^{-1}$$

inside their respective $\mathbf{F}^{(t-1)}_l$ spaces. In other words, computing the value for θ^t consists of guaranteeing that each tree of possibilities includes at least one vector of $\mathbf{X}^{(t)}_{\langle r \rangle}$ for all t values.

When θ^t is being calculated it can assume a single value (*i.e.,* it is computed from input data) or assume all values in $[-\pi/2,\pi/2)$ (*i.e.,* it is arbitrated). Given the discrete nature of the accumulator array, to assume all values in $[-\pi/2,\pi/2)$ means replicate the current 2-tuple being processed and assign a discrete value in the $[-\pi/2,\pi/2)$ range to each one of the replicas. The possible values for θ^t define the set $\mathcal{T}$ and are computed by the $CalculateParameter$ function (line 6). Once θ^t is known, **its related rotation must be rolled back** from the input blade in order to not affect the computation of θ^{t-1} (see the sandwiching construction $\boldsymbol{R}_t^{-1}\,\mathbf{X}^{(t)}_{\langle r \rangle} \,/\, \boldsymbol{R}_t^{-1}$ in line 7, where $\mathbf{P}^{(t)}_{\langle 2 \rangle}$ is the rotation plane of the t-th rotation applied to $\mathbf{E}_{\langle p \rangle}$ in Equation 5.3). Also, the parameter vector



(a) Representational space $\mathbb{R}^3$

(b) Parameter space $\mathbb{P}^{1\,(3-1)}$

Figure 5.3: A 2-dimensional Euclidean subspace $\mathbf{X}_{\langle 2\rangle}$ (a) mapped to the parameter space (b) characterizing vectors in $\mathbb{R}^3$. Each point of the curve in (b) represents a vector contained by $\mathbf{X}_{\langle 2\rangle}$ in (a).

must be updated (see $(\theta^t, \Theta_1^{(t)}, \Theta_2^{(t)}, \cdots, \Theta_{m-t}^{(t)})$ in line 7) by including the new parameter value θ^t with the other parameters $\{\Theta_k^{(t)}\}_{k=1}^{m-t}$ computed so far. At the end of the process (line 10), $\mathbf{X}_{\langle r\rangle}^{(0)} \supseteq \mathbf{E}_{\langle p\rangle}$ for all $(\mathbf{X}_{\langle r\rangle}^{(0)}, \Theta^{(0)}) \in \mathcal{P}^{(0)}$, where $\Theta^{(0)}$ is a parameter vectors resulting from mapping $\mathbf{X}_{\langle r\rangle}$ to $\mathbb{P}^m$. Note that $\boldsymbol{R}_t^{-1}\,\mathbf{X}_{\langle r\rangle}^{(t)}$ / $\boldsymbol{R}_t^{-1}$ in line 7 can be further simplified to $\widetilde{\boldsymbol{R}}_t\,\mathbf{X}_{\langle r\rangle}^{(t)}\,\boldsymbol{R}_t$. The inverse ($\boldsymbol{R}_t^{-1}$) is used in the algorithm in order to emphasizes that a rotation is being rolled back from $\mathbf{X}_{\langle r\rangle}^{(t)}$.

Figure 5.3 shows a set of parameter vectors (the points defining the curve in Figure 5.3b) resulting from mapping a 2-blade ($\mathbf{X}_{\langle 2\rangle}$ in Figure 5.3a) to the parameter space for 1-blades in $\mathbb{R}^3$, thus $r = 2$, $p = 1$, $n = 3$, and $m = 1\,(3-1) = 2$. In this case, an input blade maps to several points in the parameter space because a 2-dimensional linear space is comprised by an infinite set of vectors. Note that the parameter space is defined onto a periodic domain. Therefore, curves resulting from mapping input data can appear discontinuous regarding a single $[-\pi/2,\pi/2) \times [-\pi/2,\pi/2)$ period. In this example, the parameter $\theta^{3,1}$ (or θ^2 in single-index notation) was arbitrated by the $CalculateParameter$ function in the first iteration of the mapping procedure (the outer loop in Figure 5.2). Note that there are points covering the entire $[-\pi/2,\pi/2)$ range of $\theta^{3,1}$. The parameter $\theta^{3,2}$ (or θ^1) was computed in the second iteration of the algorithm as a single value for each possible $\theta^{3,1}$ value arbitrated previously.

In conventional HTs the input data type is known *a priori*. Thus, conventional mapping procedures predefine which parameters must be arbitrated and which ones must be computed. The proposed approach, on the other hand, does not have prior information about input data. It decides at runtime how to treat each parameter θ^t. Such a behavior is key for the generality of the proposed subspace detection framework. Another practical advantage is the ability to check at runtime which parameters can be arbitrated and which ones need to be computed to handle the gimbal lock problem. The rotations at gimbal lock are always arbitrated because they do not affect the attitude of related input and resulting blades. For the case where the input data type is known *a priori*, one can predict how the mapping procedure handles the parameters and apply the same treatment to each input blade. This way, the computational load of the technique can be reduced.

The $CalculateParameter$ function is presented in Figure 5.4. It takes as input the blade $\mathbf{Y}^{(t)}$, computed in Figure 5.2 as $\mathbf{X}_{\langle r\rangle}^{(t)}$. In this algorithm, the input blade is denoted

by $\mathbf{Y}^{(t)}$ instead of $\mathbf{X}^{(t)}_{\langle r\rangle}$ because its dimensionality may be changed while executing the procedure.

The function in Figure 5.4 is iterative (see the loop in lines 4 to 16). In line 5 it creates a set $\mathcal{M}$ containing the intersection (meet) of $\mathbf{Y}^{(t)}$ with the spaces of possibilities $\mathbf{F}^{(t)}_l$. Here we are concerned with the intersections ($\mathbf{M}^{(t)}_l$) whose vector factors can be associated to well defined $\mathbf{F}^{(t)}_l$'s at current t. These intersections define the set $\mathcal{N}$ in line 6, where $\mathcal{S}$ is a set comprised by blades $\mathbf{M}^{(t)}_h$ that are not orthogonal to $\mathbf{M}^{(t)}_l$ and have the same dimensionality as $\mathbf{M}^{(t)}_l$. As a result, a 1-blade $\mathbf{M}^{(t)}_l \in \mathcal{N}$ is related (exclusively) to the l-th space of possibilities, while a 2-blade $\mathbf{M}^{(t)}_l \in \mathcal{N}$ is related to two well-defined spaces of possibilities and so on. **When $\mathcal{N}$ is empty (line 7), θ^t assumes all value in $[-\pi/2, \pi/2)$ and the procedure stops**. Assuming all values in $[-\pi/2, \pi/2)$ means that all rotation values keep at least one vector factor of $\mathbf{X}^{(t-1)}_{\langle r\rangle} = \boldsymbol{R}_t^{-1}\,\mathbf{X}^{(t)}_{\langle r\rangle} \,/\, \boldsymbol{R}_t^{-1}$ (Equation 5.2) inside the $(t-1)$-th spaces of possibilities. Therefore, the condition depicted in Equation 5.1 will be respected. However, when $\mathcal{N}$ is not empty it must be ensured that 1-blades $\mathbf{M}^{(t)}_l \in \mathcal{N}$ contained in $\mathbf{F}^{(t)}_l$ will also be contained in $\mathbf{F}^{(t-1)}_l$ after rolling back $\boldsymbol{R}_t$ from them. Thus, the set $\mathcal{O}$ (line 10) is defined as containing the vectors in $\mathcal{N}$ that can potentially "leave" $\mathbf{F}^{(t-1)}_l$. Notice that it may happens only when the dimensionality of $\mathbf{F}^{(t-1)}_l$ is smaller than the dimensionality of $\mathbf{F}^{(t)}_l$. The vector

$$\mathbf{r}^{(t)}_l = \mathbf{F}^{(t-1)}_l \rfloor \mathbf{F}^{(t)}_l \tag{5.7}$$

in Figure 5.4 (line 3) represents the additional dimension of $\mathbf{F}^{(t)}_l$. An example of $\mathbf{r}^{(t)}_l$ in Figure 5.1a is vector $\mathbf{e}_3$ at $t = 2$ and $t = 3$.

In line 11, the set $\mathcal{Q}$ is comprised by the nonzero vectors $\mathbf{q}^{(t)}_l$ resulting from the contraction of vectors $\mathbf{m}^{(t)}_l \in \mathcal{O}$ onto the rotation plane $\mathbf{P}^{(t)}_{\langle 2\rangle}$. When $\mathbf{q}^{(t)}_l$ is zero, it means that $\mathbf{m}^{(t)}_l$ is orthogonal to $\mathbf{P}^{(t)}_{\langle 2\rangle}$ and, thus, it is not affected by a rotation in $\mathbf{P}^{(t)}_{\langle 2\rangle}$ and cannot leave $\mathbf{F}^{(t-1)}_l$. However, when $\mathbf{q}^{(t)}_l$ is not zero there is a single rotation angle θ^t that makes $\boldsymbol{R}_t^{-1}\,\mathbf{m}^{(t)}_l \,/\, \boldsymbol{R}^{-1}$ be inside $\mathbf{F}^{(t-1)}_l$. Such angle is computed in line 13 from the outer product:

$$\mathbf{q}^{(t)}_l \wedge \mathbf{r}^{(t)}_l = -\alpha\, \sin\left(\theta^t\right)\, \mathbf{P}^{(t)}_{\langle 2\rangle} \tag{5.8}$$

and the scalar product:

$$\mathbf{q}^{(t)}_l * \mathbf{r}^{(t)}_l = \alpha\, \cos\theta^t \tag{5.9}$$

of $\mathbf{q}^{(t)}_l$ and $\mathbf{r}^{(t)}_l$. The vector $\mathbf{q}^{(t)}_l$ is orthogonal to $\mathbf{m}^{(t)}_l$ and is contained in plane $\mathbf{P}^{(t)}_{\langle 2\rangle}$, as one would expect from the left contraction in line 11. α is the product of the weights of $\mathbf{q}^{(t)}_l$ and $\mathbf{r}^{(t)}_l$, and θ^t is the smallest angle between $\mathbf{q}^{(t)}_l$ and $\mathbf{r}^{(t)}_l$. **The rotation angle θ^t that makes $\mathbf{r}^{(t)}_l$ parallel to $\mathbf{q}^{(t)}_l$ is the only rotation angle that respects the condition in Equation 5.1 after rolling back rotation R_t from $\mathbf{X}^{(t)}_{\langle r\rangle}$.** This is because it ensures that $\boldsymbol{R}_t^{-1}\,\mathbf{m}^{(t)}_l \,/\, \boldsymbol{R}_t^{-1}$ will be included in $\mathbf{F}^{(t-1)}_l$. The vector $\mathbf{r}^{(t)}_l$ is included in $\mathbf{P}^{(t)}_{\langle 2\rangle}$ and $\mathbf{F}^{(t)}_l$, but it is not included in $\mathbf{F}^{(t-1)}_l$. By orthogonality, as $\mathbf{q}^{(t)}_l$ gets parallel to $\mathbf{r}^{(t)}_l$ and goes out of the tree of possibilities, $\mathbf{m}^{(t)}_l$ keeps inside of the related tree. As in line 13:

$$\theta^t = \tan^{-1}\left(\frac{\left(\mathbf{q}^{(t)}_l \wedge \mathbf{r}^{(t)}_l\right) * \mathbf{P}^{(t)}_{\langle 2\rangle}}{\mathbf{q}^{(t)}_l * \mathbf{r}^{(t)}_l}\right). \tag{5.10}$$

Require: $\mathbf{Y}^{(t)}$, the current input blade
1: Let $\mathbf{P}^{(t)}_{\langle 2\rangle}$ be the rotation plane of the t-th rotation applied to $\mathbf{E}_{\langle p\rangle}$ in Equation 5.3
2: Let $\mathbf{F}^{(t)}_{l}$ be a space of possibilities as defined in Equation 5.5
3: Let $\mathbf{r}^{(t)}_{l} \leftarrow \mathbf{F}^{(t-1)}_{l} \rfloor \mathbf{F}^{(t)}_{l}$, *i.e.,* the vector factor in $\mathbf{F}^{(t)}_{l}$ that is not in $\mathbf{F}^{(t-1)}_{l}$
4: **loop**
5: $\quad \mathcal{M} \leftarrow \left\{(\mathbf{Y}^{(t)} \cap \mathbf{F}^{(t)}_{l}) \mid l \in \mathbb{Z}, \text{and } 1 \leq l \leq |\mathcal{E}|\right\}$
6: $\quad \mathcal{N} \leftarrow \left\{\mathbf{M}^{(t)}_{l} \mid \mathbf{M}^{(t)}_{l} \in \mathcal{M}, \text{and } \operatorname{grade}(\mathbf{M}^{(t)}_{l}) = |\mathcal{S}|\right.,$
$\qquad\qquad$ where $\left.\mathcal{S} \leftarrow \left\{\mathbf{M}^{(t)}_{h} \mid \mathbf{M}^{(t)}_{h} \in \mathcal{M}, \text{and } \mathbf{M}^{(t)}_{l} * \mathbf{M}^{(t)}_{h} \neq 0\right\}\right\}$
7: $\quad$ **if** $\mathcal{N} = \varnothing$ **then**
8: $\quad\quad$ **return** $\left\{\theta^t \mid \theta^t \in [-\pi/2, \pi/2)\right\}$
9: $\quad$ **end if**
10: $\quad \mathcal{O} \leftarrow \left\{\mathbf{M}^{(t)}_{l} \mid \mathbf{M}^{(t)}_{l} \in \mathcal{N}, \text{and } \operatorname{grade}(\mathbf{M}^{(t)}_{l}) = 1, \text{and } \operatorname{grade}(\mathbf{r}^{(t)}_{l}) = 1\right\}$
11: $\quad \mathcal{Q} \leftarrow \left\{\mathbf{q}^{(t)}_{l} \mid \mathbf{q}^{(t)}_{l} = (\mathbf{m}^{(t)}_{l} \rfloor \mathbf{P}^{(t)}_{\langle 2\rangle}), \text{and } \mathbf{m}^{(t)}_{l} \in \mathcal{O}, \text{and } \mathbf{q}^{(t)}_{l} \neq 0\right\}$
12: $\quad$ **if** $\mathcal{Q} \neq \varnothing$ **then**
13: $\quad\quad$ **return** $\left\{\theta^t \mid \theta^t = \tan^{-1}\left(((\mathbf{q}^{(t)}_{l} \wedge \mathbf{r}^{(t)}_{l}) * \mathbf{P}^{(t)}_{\langle 2\rangle}) \,/\, (\mathbf{q}^{(t)}_{l} * \mathbf{r}^{(t)}_{l})\right),\right.$
$\qquad\qquad\qquad\qquad\qquad\qquad$ where $\left.\mathbf{q}^{(t)}_{l}$ is one of the vectors in $\mathcal{Q}\right\}$
14: $\quad$ **end if**
15: $\quad \mathbf{Y}^{(t)} \leftarrow (\mathbf{M}^{(t)}_{l})^{-1} \rfloor \mathbf{Y}^{(t)}$, where $\mathbf{M}^{(t)}_{l}$ is the blade with the highest dimensionality in the set $\mathcal{N}$
16: **end loop**

Figure 5.4: Function $CalculateParameter$. It complements the mapping procedure in Figure 5.2. The algorithm takes as input a r-blade $\mathbf{Y}^{(t)}$ and determines if the t-th parameter in $\Theta^{(0)}$ (Figure 5.2, line 10) can be computed from $\mathbf{Y}^{(t)}$ or if it must be arbitrated.

By replacing Equations 5.8 and 5.9 in Equation 5.10, it follows that:

$$\theta^t = \tan^{-1}\left(\frac{\alpha\, \sin(\theta^t)\, \left(-\mathbf{P}^{(t)}_{\langle 2\rangle} * \mathbf{P}^{(t)}_{\langle 2\rangle}\right)}{\alpha\, \cos\theta^t}\right) = \tan^{-1}\left(\frac{\alpha\, \sin\theta^t}{\alpha\, \cos\theta^t}\right) = \tan^{-1}\left(\frac{\sin\theta^t}{\cos\theta^t}\right).$$

If the set $\mathcal{Q}$ is empty, then there is no vector $\mathbf{q}^{(t)}_{l}$ to be used to compute θ^t. In such a case, the input blade $\mathbf{Y}^{(t)}$ is updated (line 15) by removing from $\mathbf{Y}^{(t)}$ its highest dimensional portion that certainly will not "leave" the related spaces of possibilities after rolling back $\boldsymbol{R}_t$ for any value of θ^t.

The procedure in Figure 5.4 executes p iterations in the worst case. This happens when $\mathbf{Y}^{(t)}$ is updated, at each iteration, by contracting (in line 15) only one of the factors shared with $\mathbf{C}^{(t)}_{\langle p\rangle}$.

5.2 Mapping Procedure for $r \leq p$

For the case involving $r \leq p$, one can explore the dual relationship between k-dimensional and $(n-k)$-dimensional subspaces (Section 3.6) in order to compute the parameter vectors of blades in $\mathcal{C}$. By taking $\mathbf{X}^{*}_{\langle r\rangle}$ as input and $\mathbf{E}^{*}_{\langle p\rangle}$ as reference blade, the containment relationship between $\mathbf{X}_{\langle r\rangle}$ and the elements in $\mathcal{C}$ changes from:

$$\mathbf{X}_{\langle r\rangle} \subseteq \mathbf{C}_{\langle p\rangle}$$

to

$$\mathbf{X}^*_{\langle r\rangle} \supseteq \mathbf{C}^*_{\langle p\rangle},$$

reducing the mapping problem to the case described in Section 5.1. It is important to notice that when $r \leq p$ one needs to use the set $\mathcal{E}\backslash\{\mathbf{e}_i\}_{i=1}^n$ while defining the trees of possibilities, because it contains the basis vectors spanning $\mathbf{E}^*_{\langle p\rangle}$.

5.3 Case Study: Mapping a Point to the Parameter Space for Straight Lines in 3-Dimensional Base Space

Section 4.1 presents how the proposed function (Equation 4.9) for modeling p-dimensional subspaces in some n-dimensional space is defined in order to avoid ambiguous representations for modeled subspaces. It is illustrated with an example where the homogeneous MOG (Section 3.9.2) over a 3-dimensional base space is assumed, leading to $n = 3 + 1 = 4$ (the dimensionality of the representational space). In Section 4.1, it is shown that the parameterization of 2-blades (thus, $p = 2$) geometrically interpreted as straight lines in the MOG is defined from the parameters of pairs of pseudovectors (3-blades interpreted as planes) containing them. The current section extends such an example by discussing the case study of mapping the vector

$$\mathbf{x} = 0.5\,\mathbf{e}_1 + 0.3\,\mathbf{e}_2 + \mathbf{e}_4 \in \mathbb{R}^4 \tag{5.11}$$

to the parameter space $\mathbb{P}^4$ (Equation 4.10) defined for $p = 2$ and $n = 4$. Recall from Section 4.1 that $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4\}$ are the vectors of the representational space, and $\mathbf{e}_4$ is geometrically interpreted as the point at the origin of the 3-dimensional base space. Under the homogeneous MOG, $\mathbf{x}$ (Equation 5.11) is geometrically interpreted as a proper point at the location $(0.5, 0.3, 0.0)$ of the base space.

The algorithm that maps input r-blades to $\mathbb{P}^m$ is presented in Figures 5.2 and 5.4, and it is described in Section 5.1. The mapping algorithm is defined for input subspaces having dimensionality greater or equal to the dimensionality of intended subspaces. However, in the current example, the input subspace ($\mathbf{x}$) is *one*-dimensional, while the intended subspaces interpreted as straight lines are bidimensional in the assumed MOG. Thus, according to Section 5.2, one needs to take the dual of $\mathbf{x}$ as the used input ($\mathbf{X}_{\langle r\rangle}$ in Figure 5.2), and to use the dual of $\mathbf{e}_2^* \vee \mathbf{e}_4^*$ (*i.e.,* the reference subspace for 2-blades, see Equation 4.18) as reference blade ($\mathbf{E}_{\langle p\rangle}$ in Figures 5.2 and 5.4). By doing so:

$$\mathbf{X}_{\langle 4-1\rangle} = \mathbf{x}^* = 0.5\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.3\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \tag{5.12}$$

and

$$\mathbf{E}_{\langle 4-2\rangle} = (\mathbf{e}_2^* \vee \mathbf{e}_4^*)^* = -\mathbf{e}_2 \wedge \mathbf{e}_4, \tag{5.13}$$

reducing the mapping procedure to the case described in Section 5.1 (*i.e.,* $r \geq p$, because $r = 4 - 1 = 3$ in Equation 5.12, and $p = 4 - 2 = 2$ in Equation 5.13). From the basis vectors spanning the reference blade $\mathbf{E}_{\langle 4-2\rangle}$ (Equation 5.13), it follows that the set $\mathcal{E} = \{\mathbf{e}_2, \mathbf{e}_4\}$. In this case, the spaces of possibilities $\mathbf{F}_l^{(t)}$ of vectors $\mathbf{v}_l \in \mathcal{E}$ (used in Figure 5.4) are the ones depicted in Figure 5.1.

In the first step of the mapping algorithm (Figure 5.2, line 1), the set $\mathcal{P}^{(4)}$ is initialized:

$$\mathcal{P}^{(4)} = \big\{(\mathbf{X}_{\langle 4-1\rangle}, \varnothing)\big\},$$

where $\mathbf{X}_{\langle 4-1\rangle}$ is given by Equation 5.12. The (single) 2-tuple in $\mathcal{P}^{(4)}$ is processed during the first iteration of the loop (line 2 to 9, for $t = 4$), resulting in the set $\mathcal{P}^{(3)}$. In line 6, the $CalculateParameter$ function is called in order to determine if the 4th coordinate of the resulting parameter vectors (θ^4) can be computed from $\mathbf{X}_{\langle 4-1\rangle}$ or if it must be arbitrated. In this case, the parameter is arbitrated. Notice, by looking to the line $t = 4$ of Figure 5.1, that there is no rotation on plane $\mathbf{P}^{(4)}_{\langle 2\rangle} = \mathbf{e}_2 \wedge \mathbf{e}_1$ that, when applied to $\mathbf{X}_{\langle 4-1\rangle}$, makes the transformed input subspace leave some of the trees of possibilities by complete at $t = 3$. Actually, there is no 3-blade that can be spanned outside some of the spaces of possibilities at $t = 3$. The space of possibilities $\mathbf{F}^{(3)}_1 = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$, for instance, does not include only one of the dimensions of the total 4-dimensional space (*i.e.,* $\mathbf{e}_4$, in Figure 5.1a, for $t = 3$). The same happens to $\mathbf{F}^{(3)}_2 = \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4$, where only the dimension $\mathbf{e}_1$ is missing. Also, the dimension not included by $\mathbf{F}^{(3)}_1$ is included by $\mathbf{F}^{(3)}_2$, and vice-versa.

In this case study, the step for defining the linear discretization of the parameter space is $\pi/6$. By arbitrating the value of the 4th parameter (θ^4):

$$\mathcal{T} = \left\{-\tfrac{\pi}{2}, -\tfrac{\pi}{3}, -\tfrac{\pi}{6}, 0, \quad \tfrac{\pi}{6}, \quad \tfrac{\pi}{3}\right\},$$

in line 6, and $\mathcal{P}^{(3)}$ is comprised by six 2-tuples:

$$\begin{aligned}
\mathcal{P}^{(3)} = \big\{ & (\ \ 0.30\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.50\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (-\tfrac{\pi}{2})), \\
& (\ \ 0.51\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.28\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (-\tfrac{\pi}{3})), \\
& (\ \ 0.58\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.01\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (-\tfrac{\pi}{6})), \\
& (\ \ 0.50\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.30\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (\ \ 0)), \\
& (\ \ 0.28\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.51\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (\ \ \tfrac{\pi}{6})), \\
& (-0.01\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.58\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (\ \ \tfrac{\pi}{3}))\big\}.
\end{aligned} \tag{5.14}$$

It is important to comment that the coefficients of the blades (the first entry of the tuples in $\mathcal{P}^{(3)}$) were rounded to two digits after the decimal point in order to fit the page. In practice, 32-bit floating-point representations have being used in the computations.

In the second iteration of the outer loop (Figure 5.2, line 2 to 9), $t = 3$. Each element in $\mathcal{P}^{(3)}$ is processed during the inner loop (line 5 to 8), resulting in the set $\mathcal{P}^{(2)}$. For all these elements, the current parameter (θ^3) is arbitrated by the $CalculateParameter$ function for the same reasons as in previous iteration: there is no 3-blade that can be spanned outside the spaces of possibilities at $t = 2$, and $\mathbf{F}^{(2)}_1 = \mathbf{e}_1 \wedge \mathbf{e}_2$ and $\mathbf{F}^{(2)}_2 = \mathbf{e}_3 \wedge \mathbf{e}_4$ are complementary with respect to the whole space. As a result, $\mathcal{P}^{(2)}$ is comprised by 36 tuples (*i.e.,* each one of the six elements in $\mathcal{P}^{(3)}$ creates six elements in $\mathcal{P}^{(2)}$). The six new tuples related to the first 2-tuple in Equation 5.14 are presented next. The other tuples are omitted for sake of simplicity:

$$\begin{aligned}
\mathcal{P}^{(2)} = \big\{ & \\
& (0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.50\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (-\tfrac{\pi}{2}, -\tfrac{\pi}{2})), \\
& (0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.25\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.43\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (-\tfrac{\pi}{3}, -\tfrac{\pi}{2})), \\
& (0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.43\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.25\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (-\tfrac{\pi}{6}, -\tfrac{\pi}{2})), \\
& (0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.50\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (\ \ 0, -\tfrac{\pi}{2})), \\
& (0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.43\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.25\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (\ \ \tfrac{\pi}{6}, -\tfrac{\pi}{2})), \\
& (0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.25\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.43\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, (\ \ \tfrac{\pi}{3}, -\tfrac{\pi}{2})), \cdots \big\}.
\end{aligned}$$

It is important to emphasize that the $CalculateParameter$ procedure (Figure 5.4) is capable to identify that parameters θ^4 and θ^3 must be arbitrated in the current case study. However, in order to avoid the tedious repetition of the procedure, such an identification was performed up to this point by using some geometric intuition about spaces of possibilities and the dimensionality of the given input blade rather than by running the algorithm in Figure 5.4 step-by-step. For the next iteration in Figure 5.2, the $CalculateParameter$ function must be evaluated in order to compute a single value for θ^2. It is because there is no guarantee that all possible rotations on plane $\mathbf{P}^{(2)}_{\langle 2\rangle} = \mathbf{e}_4 \wedge \mathbf{e}_3$ (Figure 5.1, $t = 2$) make the transformed input subspace keep at least one vector factor inside each space of possibilities for $t = 1$ ($\mathbf{F}^{(1)}_1 = \mathbf{e}_1 \wedge \mathbf{e}_2$, and $\mathbf{F}^{(1)}_2 = \mathbf{e}_4$ in Figure 5.1). For instance, by making $\theta^2 = -\pi/2$, the blade related to the first tuple in $\mathcal{P}^{(2)}$ would be transformed from:

$$\mathbf{X}^{(2)}_{\langle 4-1\rangle} = 0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.50\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, \tag{5.15}$$

to:

$$\begin{aligned}\mathbf{X}^{(1)}_{\langle 4-1\rangle} &= \boldsymbol{R}_2^{-1}\,\mathbf{X}^{(2)}_{\langle 4-1\rangle} \,/\, \boldsymbol{R}_2^{-1} \\ &= 0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 0.5\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 + \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3,\end{aligned} \tag{5.16}$$

which has no common vector factor with $\mathbf{F}^{(1)}_2 = \mathbf{e}_4$ (Figure 5.1b, $t = 1$). In Equation 5.16,

$$\boldsymbol{R}_2 = \cos\left(-\frac{\pi}{4}\right) - \sin\left(-\frac{\pi}{4}\right)\,\mathbf{e}_4 \wedge \mathbf{e}_3.$$

In the third iteration of the mapping procedure, $t = 2$, and the inner loop processes each element in $\mathcal{P}^{(2)}$. The first tuple in $\mathcal{P}^{(2)}$ is $\mathbf{X}^{(2)}_{\langle 4-1\rangle}$ (Equation 5.15). In line 6 of Figure 5.2, the $CalculateParameter$ function is called having $\mathbf{X}^{(2)}_{\langle 4-1\rangle}$ as argument, making:

$$\mathbf{Y}^{(2)} = \mathbf{X}^{(2)}_{\langle 4-1\rangle} = 0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.50\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4 - \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \tag{5.17}$$

in Figure 5.4. The set $\mathcal{M}$ is created in line 5 of Figure 5.4 as the set of subspaces resulting from the intersection of $\mathbf{Y}^{(2)}$ (Equation 5.17) with the spaces of possibilities $\mathbf{F}^{(2)}_1 = \mathbf{e}_1 \wedge \mathbf{e}_2$ and $\mathbf{F}^{(2)}_2 = \mathbf{e}_3 \wedge \mathbf{e}_4$:

$$\begin{aligned}\mathcal{M} = \big\{&\mathbf{Y}^{(2)} \cap \mathbf{F}^{(2)}_1 = \mathbf{e}_2, \\ &\mathbf{Y}^{(2)} \cap \mathbf{F}^{(2)}_2 = 0.8\,\mathbf{e}_3 - 0.4\,\mathbf{e}_4\big\}.\end{aligned} \tag{5.18}$$

Both subspaces in $\mathcal{M}$ are associated with well defined spaces of possibilities. Notice that $\mathbf{e}_2$ is contained only in $\mathbf{F}^{(2)}_1$, while $0.8\,\mathbf{e}_3 - 0.4\,\mathbf{e}_4$ is related exclusively to $\mathbf{F}^{(2)}_2$. Thus, according to line 6 of Figure 5.4, the set $\mathcal{N}$ is defined as $\mathcal{N} = \mathcal{M}$. Since $\mathcal{N}$ is not empty, it follows from line 10 that:

$$\mathcal{O} = \big\{0.8\,\mathbf{e}_3 - 0.4\,\mathbf{e}_4\big\}, \tag{5.19}$$

because the only tree of possibilities whose spaces of possibilities change the dimensionality from $t = 2$ to $t = 1$ is the one in Figure 5.1b. This way, the set $\mathcal{Q}$ is computed in line 11 as:

$$\mathcal{Q} = \big\{(0.8\,\mathbf{e}_3 - 0.4\,\mathbf{e}_4) \rfloor (\mathbf{e}_4 \wedge \mathbf{e}_3) = -0.4\,\mathbf{e}_3 - 0.8\,\mathbf{e}_4\big\}. \tag{5.20}$$

Finally, the 2nd parameter (θ^2) can be computed in line 13 from the vector in $\mathcal{Q}$ and the reference vector $\mathbf{r}_2^{(2)} = -\mathbf{e}_3$ as:

$$\begin{aligned}\theta^2 &= \tan^{-1}\left(\frac{(\mathbf{q}_l^{(2)} \wedge \mathbf{r}_l^{(2)}) * \mathbf{P}_{\langle 2\rangle}^{(2)}}{\mathbf{q}_l^{(2)} * \mathbf{r}_2^{(2)}}\right)\\ &= \tan^{-1}\left(\frac{((-0.4\,\mathbf{e}_3 - 0.8\,\mathbf{e}_4) \wedge (-\mathbf{e}_3)) * (\mathbf{e}_4 \wedge \mathbf{e}_3)}{(-0.4\,\mathbf{e}_3 - 0.8\,\mathbf{e}_4) * (-\mathbf{e}_3)}\right)\\ &= \tan^{-1}\left(-\frac{0.8}{0.4}\right) = -1.1071.\end{aligned} \tag{5.21}$$

The current parameter (θ^2) is computed for each element in $\mathcal{P}^{(2)}$ in the same way as presented for $\mathbf{X}_{\langle 4-1\rangle}^{(2)}$ in Equations 5.17 to 5.21. After the tuples in $\mathcal{P}^{(2)}$ have being processed by the mapping procedure (Figure 5.2), the resulting set $\mathcal{P}^{(1)}$ is:

$$\begin{aligned}\mathcal{P}^{(1)} = \big\{&\\ &(0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.12\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (-1.11, -\tfrac{\pi}{2}, -\tfrac{\pi}{2})),\\ &(0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.25\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.09\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (-1.16, -\tfrac{\pi}{3}, -\tfrac{\pi}{2})),\\ &(0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.43\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.03\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (-1.33, -\tfrac{\pi}{6}, -\tfrac{\pi}{2})),\\ &(0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.50\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.00\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (-1.57, \quad 0, -\tfrac{\pi}{2})),\\ &(0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.43\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 1.03\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (\ \ 1.33, \quad \tfrac{\pi}{6}, -\tfrac{\pi}{2})),\\ &(0.3\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 0.25\,\mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 1.09\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (\ \ 1.16, \quad \tfrac{\pi}{3}, -\tfrac{\pi}{2})), \cdots\big\}.\end{aligned} \tag{5.22}$$

At this point, the three last parameters of the 2-dimensional subspaces (interpreted as straight lines) related to the input blade $\mathbf{x}$ were calculated. Section 4.1 has been shown that those parameters are related to the pseudovector defining the distance of the line to the origin of the base space (*i.e.,* they are related to the plane $\mathbf{G}_{\langle 3\rangle}$ in Figure 4.2a). Those parameters are also related to the direction of the intended lines, up to one degree of freedom defining $\mathbf{F}_{\langle 3\rangle}$. Such a degree of freedom is represented by the parameter θ^1, computed at the last iteration of the mapping procedure.

In the last iteration, the 2-tuples in $\mathcal{P}^{(1)}$ (Equation 5.22) are processed by the mapping algorithm in order to create the set $\mathcal{P}^{(0)}$ with blades $\mathbf{X}_{\langle 4-1\rangle}^{(0)}$ and the resulting parameter vectors. In this case study, the parameter θ^1 assumes a single value for each one of the blades in $\mathcal{P}^{(1)}$. After the end of the last iteration, the set $\mathcal{P}^{(0)}$ is given by:

$$\begin{aligned}\mathcal{P}^{(0)} = \big\{&(0.30\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.12\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (0.00, -1.11, -\tfrac{\pi}{2}, -\tfrac{\pi}{2})),\\ &(0.39\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.09\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (0.69, -1.16, -\tfrac{\pi}{3}, -\tfrac{\pi}{2})),\\ &(0.53\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.03\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (0.96, -1.33, -\tfrac{\pi}{6}, -\tfrac{\pi}{2})),\\ &(0.58\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 + 1.00\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (1.03, -1.57, \quad 0, -\tfrac{\pi}{2})),\\ &(0.53\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 1.03\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (0.96, \quad 1.33, \quad \tfrac{\pi}{6}, -\tfrac{\pi}{2})),\\ &(0.39\,\mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 - 1.09\,\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4, (0.69, \quad 1.16, \quad \tfrac{\pi}{3}, -\tfrac{\pi}{2})), \cdots\big\},\end{aligned}$$

and the resulting parameter vectors (Figure 5.2, line 10) are:

$$\begin{aligned}\big\{&(0.00, -1.11, -\tfrac{\pi}{2}, -\tfrac{\pi}{2}), (0.69, -1.16, -\tfrac{\pi}{3}, -\tfrac{\pi}{2}), (0.96, -1.33, -\tfrac{\pi}{6}, -\tfrac{\pi}{2}),\\ &(1.03, -1.57, \quad 0, -\tfrac{\pi}{2}), (0.96, \quad 1.33, \quad \tfrac{\pi}{6}, -\tfrac{\pi}{2}), (0.69, \quad 1.16, \quad \tfrac{\pi}{3}, -\tfrac{\pi}{2}), \cdots\big\}.\end{aligned}$$

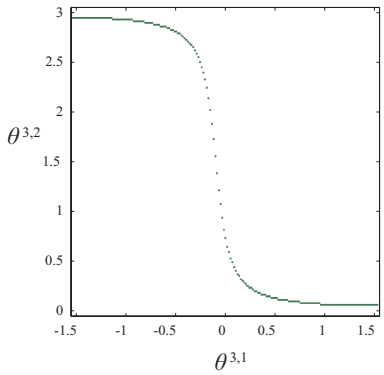
It is important to emphasize that the actual geometric interpretation of the input blades or of the intended subspaces were not taken into account while performing the mapping procedure. From our experience, it is being difficult to express the relation among the computation of the parameters used in the proposed subspace detection scheme and the computation of the ones from conventional parameterizations. It was observed that it is better to always think about Euclidean subspaces rather than specific geometric shapes in order to understand the computational chain or while extending the proposed framework.

5.4 Voting Procedure

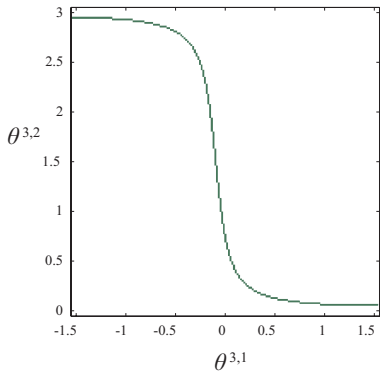
Mapping an input blade $\mathbf{X}_{\langle r \rangle}$ to parameter space $\mathbb{P}^m$ would result in a continuous curve/surface in $\mathbb{P}^m$. Thus, the use of a discrete representation of the parameter space requires a voting procedure for rasterizing the continuous structure and incrementing the related bins in the accumulator array. The simplest way to perform the rasterization is by setting the values of arbitrated parameters according to the discretization of $\mathbb{P}^m$, and by rounding the computed parameters to their closest discrete values. In this way, a continuous structure is approximated by a set of discrete point samples, and the incrementing is performed by adding the importance value ω of $\mathbf{X}_{\langle r \rangle}$ to the bins addressed by the coordinates of the samples. However, an improper choice of discretization values may lead to the identification of sparse bins in the accumulator array. Figures 5.5a and 5.5c illustrate this situation in the accumulator arrays produced for, respectively, straight line detection and circle detection from a single input point. In Figure 5.5a the homogeneous MOG (Section 3.9.2) over a 2-dimensional base space is assumed, leading to $n = 2 + 1 = 3$ (the dimensionality of the representational space), $p = 2$ (the dimensionality of blades interpreted as straight lines in the MOG), $m = 2\,(3 = 2) = 2$ (the dimensionality of the parameter space), and $r = 1$ (the dimensionality of the input vector interpreted as a point). In Figure 5.5c the parameter space for circles in conformal MOG (Section 3.9.3) is presented, where $n = 2 + 2 = 4$, $p = 3$, $m = 3\,(4 - 3) = 3$, and $r = 1$. In both examples, the step for defining the linear discretization of the parameter spaces is $\pi/180$. Note how the representation of the continuous structures is compromised as $\theta^{3,2}$ in Figure 5.5a and $\theta^{4,3}$ in Figure 5.5c approach $\pi/2$. In these examples, the range of values for $\theta^{3,2}$ and $\theta^{4,3}$ have been translated from $[-\pi/2, \pi/2)$ to $[0, \pi)$ in order to improve the visualization of gaps in the rasterized structures.

A more accurate representation of curves and surfaces in the accumulator array can be obtained using a small discretization step. However, such an approach may lead to excessive use of memory to allocate the accumulator array, especially for higher dimensional parameter spaces (*i.e.,* when $m > 2$). Also, it has been shown by van Veen and Groen (1981), and Lam *et al.* (1994), that improper choice of discretization values in conventional HTs (and hence in the generalization describe in this dissertation) may lead to unsharp or multiple peaks of votes. When the parameter space is poorly discretized the bins may be too big and represent too many instances of the intended data alignment, preventing the accurate detection of the intended shapes. On the other hand, when the bins are too small the votes will fall in the neighboring bins, thus reducing the visibility of the main peaks of votes.

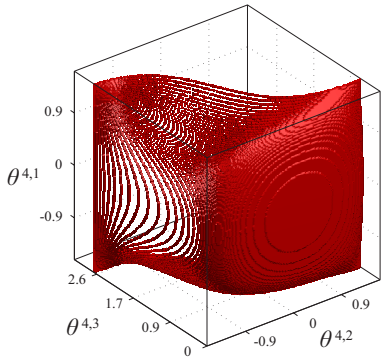
The solution proposed in this work is to use a not so small discretization step defined empirically for a given detection case, and to perform linear interpolation among samples (points) on the continuous curves/surfaces in order to avoid gaps. The results of the interpolation for the examples depicted in Figures 5.5a and 5.5c are presented in Figures 5.5b



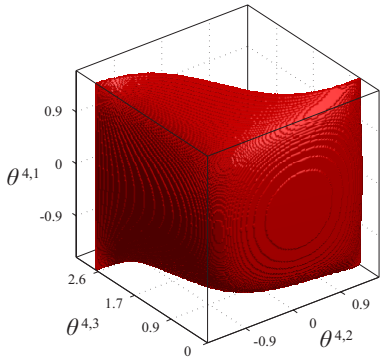
(a) Accumulator array for $\mathbb{P}^{2\,(3-2)}$



(b) Accumulator array for $\mathbb{P}^{2\,(3-2)}$



(c) Accumulator array for $\mathbb{P}^{3\,(4-3)}$



(d) Accumulator array for $\mathbb{P}^{3\,(4-3)}$

Figure 5.5: An improper choice of discretization values may lead to discontinuities in the representation of continuous curves and surfaces in the accumulator array. The accumulator array for the detection of straight lines in the 2-dimensional homogeneous MOG is presented in (a) and (b). The curve represents all the straight lines passing through a given input point. In (a) a naive voting procedure was performed. Note how the discretization step of $\pi/180$ produces a set of sparse samples on the curve. In (b) the voting is performed by interpolating pairs of neighbor samples. The problem of discontinuity can also be observed in (c), where the surface should represent all the circles passing through a given point encoded in the conformal MOG. In (d) the problem is corrected using interpolation.

and 5.5d, respectively. The interpolation-based voting procedure takes a given input blade $\mathbf{X}_{\langle r \rangle}$ and uses the mapping scheme described in Sections 5.1 and 5.2 to identify in $\mathbb{P}^m$ a set of samples (points) on the curve/surface related to $\mathbf{X}_{\langle r \rangle}$. The arbitrated parameters receive values according to the discretization of $\mathbb{P}^m$, while the other parameters receive the values computed regarding the continuous domain (*i.e.,* rounding is not performed as in the naive voting procedure). A mesh of k-simplices (*e.g.,* a 0-simplex is a point, a 1-simplex is a straight line segment, a 2-simplex is a triangle, and so on) is created by connecting neighbor samples, and the bins are incremented according to the rasterization of such a mesh. The dimensionality of simplices (k) is determined by the number of arbitrated parameters. Therefore, when no parameter is arbitrated, $k = 0$ and the voting is performed regarding a single point in $\mathbb{P}^m$. For one ($k = 1$), two ($k = 2$) and three ($k = 3$) arbitrated parameter the voting procedure rasterizes, respectively, a polygonal line, a mesh of triangles and a mesh of tetrahedra. Notice that to tessellate a mesh of simplices from the samples is straightforward because the neighborhood of a sample is known from the arbitrated parameter values. Thus, from a set of 2^k neighbor samples one can build a k-dimensional

cube and use a precomputed tessellation of the cube to define a piece of the mesh. In the examples depicted in Figures 5.5a and 5.5b, the parameter $\theta^{3,1}$ was arbitrated, while the parameter $\theta^{3,2}$ was computed from the values of $\theta^{3,1}$ and the given input blade, leading to 180 samples due to the assumed discretization step $\pi/180$. In this example, the samples on the curve define a polygonal line, whose rasterization is shown in Figure 5.5b.

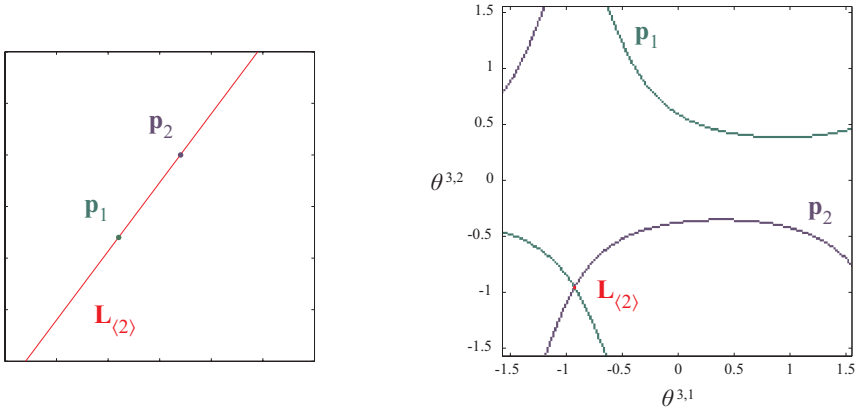
In the examples presented in this dissertation, the rasterization of a given mesh to the accumulator array has been performed using a two-stage approach. In the first stage an auxiliary buffer of Boolean values with the same size of the accumulator array (*i.e.,* the presence buffer) is initialized with *false*, indicating that they are not related to any simplex. The cells of the presence buffer are set to *true* as the simplices are rasterized to them. The rasterization is performed using the *Standard Model* (ANDRES, 2003). The Standard Model is an analytical description model for the discretization of Euclidean linear objects (*e.g.,* point, half space, k-flat and k-simplex) in dimension m. The algorithm for the rasterization of a given k-simplex is presented in Appendix D. In the second stage, the bins of the accumulator array related to the cells marked with *true* in the presence buffer are incremented by the importance ω. The presence buffer helps to avoid incrementing a single bin multiple times for the same mesh. That could happen, for instance, while rasterizing simplices sharing a face or vertex directly to the accumulator array.

5.5 Results and Discussion

The mapping and voting procedures introduced in this chapter have been implemented using C++, and the detection results (as well as all the other images presented in this dissertation) have been displayed with MATLAB® . I have chosen to use my own GA library (*i.e., Geometric Algebra Template Library*, GATL) in such proof of concept implementations. However, any other library implementing the basic products of GA could be used instead (*e.g.,* Gaigen 2 (FONTIJNE, 2006), GluCat (LEOPARDI, 2009)). The advantage of using GATL is that it is based on meta-programming with template and is designed to automatically execute low level algebraic manipulation in the implemented procedures. This way, GATL is capable to perform some basic optimizations on the program at compile time, as opposed to other solutions (FONTIJNE, 2006) that require an additional profiling step in order to produce the same kind of optimization.

Figure 5.6b illustrates the accumulator array computed for the straight-line detection from points $\mathbf{p}_1$ and $\mathbf{p}_2$ in Figure 5.6a. In this example the homogeneous MOG over a 2-dimensional base space is assumed, leading to $n = 2 + 1 = 3$, $p = 2$, and $r = 1$. The importance value (ω) of each input is set to *one* (*i.e.,* the rasterized curves increment the related bins of the accumulator array by only one vote). The two curves in Figure 5.6b are defined by mapping $\mathbf{p}_1$ and $\mathbf{p}_2$ to the 2-dimensional parameter space ($m = 2\,(3 - 2) = 2$). In such a mapping, the parameter $\theta^{3,1}$ assumes all values in $[-\pi/2, \pi/2)$, while $\theta^{3,2}$ is computed from $\theta^{3,1}$ and a given input vector. The intersection of the curves in the accumulator array (the bin having two votes) defines the parameter vector related to $\mathbf{L}_{\langle 2 \rangle}$ in Figure 5.6a. In Figure 5.6b, the accumulator array is obtained as the linear discretization of the parameter space, using $\pi/180$ as discretization step.

The accumulator array for a simple case of circle detection is presented in Figure 5.7b. For this example, data is being represented in the conformal MOG, leading to $p = 3$, $n = 2 + 2 = 4$, and $m = 3\,(4 - 3) = 3$. The points $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$ in Figure 5.7a are used as input vectors ($r = 1$). Each input vector maps to a curved surface in parameter space (see Figure 5.7b). The intersection of the three surfaces defines the parameter vector



(a) Base space $\mathbb{R}^2$ (b) Accumulator array for $\mathbb{P}^{2\,(3-2)}$

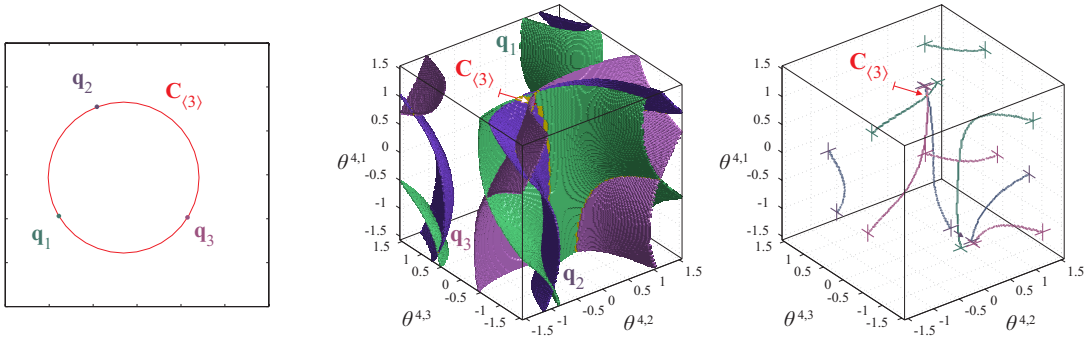
Figure 5.6: Resulting accumulator arrays for a simple detection case of line from points: (a) In homogeneous MOG, the 2-blade $\mathbf{L}_{\langle 2\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2$ is interpreted as a straight line, and vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ are interpreted as points. (b) Accumulator array resulting from the straight-line detection using points $\mathbf{p}_1$ and $\mathbf{p}_2$ from (a) as input data. The bin with two votes corresponds to the parameters of $\mathbf{L}_{\langle 2\rangle}$.

related to $\mathbf{C}_{\langle 3\rangle}$ in Figure 5.7a. In this example, $\omega = 1$ and the step for linear discretization of the parameter space is $\pi/180$.

Figure 1.1a shows the detection of subspaces geometrically interpreted as the straight lines that best fit a set of $395{,}248$ vectors interpreted as points under homogeneous MOG. Those points were generated by supersampling ($16\times$) each one of the $24{,}703$ edge pixels in Figure 1.1c, obtained from Figure 1.1a after a Canny edge detector (CANNY, 1986) plus thresholding and thinning. The supersampling is used in order to treat edge pixels as area elements rather than as point elements, leading to a smoother distribution of votes in the resulting accumulator array. In this example, the discretization step for defining the accumulator array is $\pi/900$, and the importance value of each input is the magnitude of the gradient computed by the edge detector.

Figure 1.1b illustrates the detection of circles that best fit a set of $1{,}353{,}760$ subspaces encoding tangent directions in conformal MOG. A tangent direction (Table 3.1, column 3, line 11) is a geometric primitive encoding the subspace tangent to rounds at a given location. Therefore, tangent directions have a point-like interpretation, and also direction information assigned to them. The input tangent directions (2-blades, leading to $r = 2$) were computed from $8{,}461$ edge pixels (Figure 1.1d) and their gradient vector directions (supersampled as 160 random samples per edge pixel to account for ± 0.35 radians of uncertainty on the gradient direction). As in Figure 1.1a, ω is the magnitude of gradient directions. In order to make the irregular imaged structures become more circular, the image in Figure 1.1b was convolved with a pillbox filter of radius 5 pixels before edge detection. Retrieved circles having radius larger than 50% the image width were discarded to avoid detecting the plate. In this example, the accumulator array was defined as the linear discretization of the parameter space, using $\pi/900$ as discretization step.

The use of tangent directions (2-blades) while searching for circles allows a simpler voting procedure than when using points (1-blades without tangent information), due to the constraint imposed by the directional information. Figure 5.7c illustrates the result of the mapping of directions tangent to $\mathbf{C}_{\langle 3\rangle}$ at points $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$ (Figure 5.7a) to the parameter space. By comparing the accumulator arrays in Figure 5.7b and Figure 5.7c,



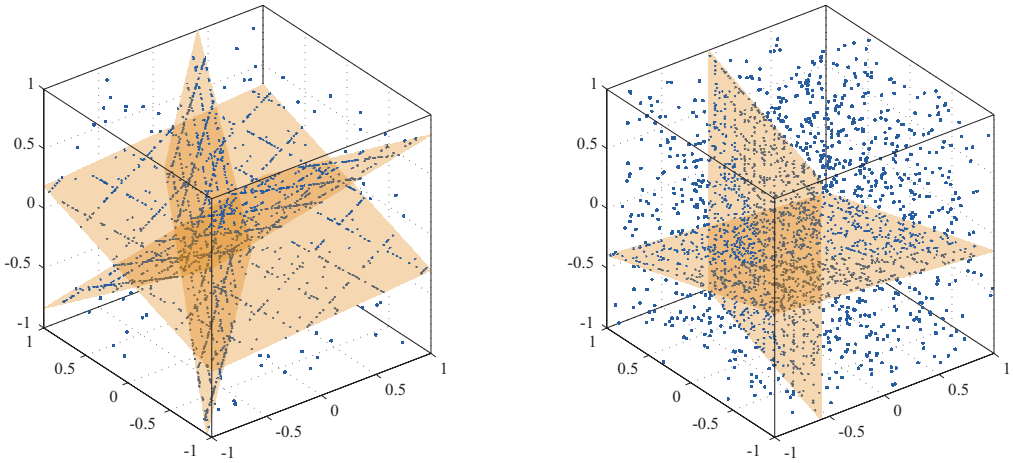
(a) Base space $\mathbb{R}^2$ (b) Accumulator array for $\mathbb{P}^{3\,(4-3)}$ (c) Accumulator array for $\mathbb{P}^{3\,(4-3)}$

Figure 5.7: Resulting accumulator arrays for a simple detection case of circle from points and tangent subspaces: (a) In conformal MOG, the 3-blade $\mathbf{C}_{\langle 3\rangle} = \mathbf{q}_1 \wedge \mathbf{q}_2 \wedge \mathbf{q}_3$ is interpreted as a circle, and vectors $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$ are interpreted as points. (b) Accumulator array resulting from circle detection using points $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$ from (a) as input. (c) Another circle detection case involving (a). However, for this example the subspaces tangent to $\mathbf{C}_{\langle 3\rangle}$ at points $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$ were used as input, leading to a simpler voting procedure. The bins with three votes in (b) and (c) correspond to the parameters of $\mathbf{C}_{\langle 3\rangle}$.

one should notice that the directional information of a given tangent direction restricts the mapping of each input blade to a curve (Figure 5.7c) on the surface related to its respective point (Figure 5.7b). Such an expected behavior is a natural outcome of the proposed general mapping procedure (Section 5.1). The subspace detection from tangent directions in the conformal MOG corresponds to the generalization of the conventional HTs presented by O'Gorman and Clowes (1973), and Kimme *et al.* (1975) (the techniques (8) and (9) in Figure 2.2).

The mapping procedure described in this dissertation allows the detection of subspaces in heterogeneous datasets. Figure 1.2a presents a synthetic dataset illustrating the use of homogeneous MOG for detection of lines ($p = 2$) that best fit a heterogeneous input set comprised by 45 points ($r = 1$) and 1 plane ($r = 3$) in a 3-dimensional base space (leading to $n = 3 + 1 = 4$, and $m = 2\,(4 - 2) = 4$). The task in this example is to detect the lines on the input plane that are also best fit for collinear input points. In order to solve this problem, the importance of the points was set to $\omega = 1$ and the importance of the plane to $\omega = 45$ (the number of points). After performing the voting procedure, the bins in the accumulator array having 47 votes or more represent the lines (on the plane) defined by at least two points. Notice in Figure 1.2a that one subset of the points clearly defines a line, but it was not retrieved because such line is not on the plane. This example shows how the proposed approach can be used to perform more complex coherence queries on data than conventional HTs. For this example, the accumulator array was defined as the linear discretization of the parameter space, using $\pi/360$ as discretization step.

Some MOGs may represent different geometric shapes with subspaces having the same dimensionality. In conformal MOG, for instance, lines and circles are 3-dimensional subspaces, and planes and spheres are 4-dimensional subspaces (Table 3.1). The proposed approach takes advantage of such a feature, allowing the concurrent detection of all shapes that have the same dimensionality on a given MOG. Figure 1.2b illustrates this situation, where 1 plane and 2 spheres ($p = 4$) are detected simultaneously. In this example the heterogeneous (synthetic) dataset is comprised by 43 points ($r = 1$), 1 straight line ($r = 3$),



(a) Base space $\mathbb{R}^3$ (b) Base space $\mathbb{R}^3$

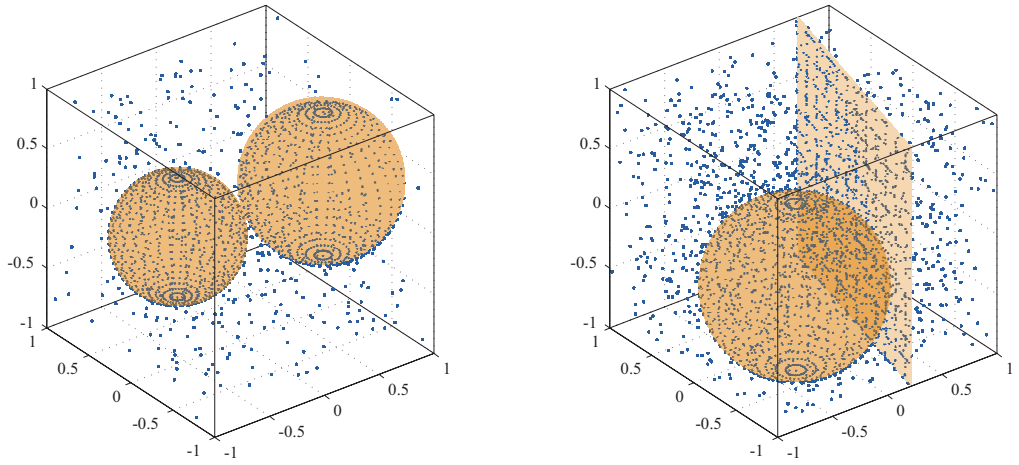
Figure 5.8: Detection of planes from points encoded into the 3-dimensional homogeneous MOG. (a) The input dataset is comprised by 2,386 vectors geometrically interpreted as points. Here, 2,186 of the input vectors define three pseudovectors geometrically interpreted as planes, while 200 input vectors characterize noise. (b) Two subspaces interpreted as planes are approximated by 1,355 vectors interpreted as points. A subset of 1,500 vectors characterizes noise in the input dataset.

and 3 circles ($r = 3$) in conformal MOG. The importance values of input blades was set to *one*, and the accumulator array was defined by using $\pi/360$ as discretization step.

Figure 5.8 shows two synthetic datasets comprised by points encoded into homogeneous MOG ($n = 3 + 1$, and $r = 1$). The points were randomly distribution on grids in order to improve the visualization of them as part of some plane to be detected ($p = 3$). In Figure 5.8a, 2,186 points are contained by three planes. The dataset also includes 200 uniformly distributed random points characterizing noise. The three detected planes are shown in orange. Notice that these planes fit the coplanar points. The dataset in Figure 5.8b is comprised by 1,355 points which approximate two planes, and 1,500 uniformly distributed random points (the non-coplanar ones), leading to 2,855 input entries altogether. The example depicted Figure 5.8b shows that the proposed approach can identify subspaces even in datasets having noise. In this example, the signal-to-noise ratio is 1.9. For the examples in Figure 5.8, the importance value of input blades was set to *one*, and the accumulator array was defined by using $\pi/450$ as discretization step (*i.e.,* 450 discrete angular values per axis of the 3-dimensional parameter space).

Figure 5.9 presents two synthetic datasets under conformal MOG ($n = 3 + 2$). In Figure 5.9a, 1,821 points ($r = 1$) define two spheres ($p = 4$), while 500 points characterize outliers in the 3-dimensional base space. Altogether, there are 2,321 input entries. Notice that both spheres where detected. However, they do not fit perfectly to the input data. The sphere at the right side of Figure 5.9a, for instance, is slightly shifted upwards. It is due to the coarse discretization step assumed for the accumulator array ($\pi/90$ radians). As a results, each bin of the accumulator array may represent a large set of p-dimensional subspaces. Figure 5.9b illustrates the detection of a plane and a sphere ($p = 4$) in an input set of 3,314 points ($r = 1$), where 1,500 of such points characterize random noise with uniform distribution (the signal-to-noise ratio is 2.21).

An approximation of the dth-order Voronoi diagram (VORONOI, 1908) of a set of points in $\mathbb{R}^d$ can be retrieved as byproduct of the detection of subspaces geometrically



(a) Base space $\mathbb{R}^3$ (b) Base space $\mathbb{R}^3$

Figure 5.9: Detection of planes and spheres from points encoded into the 3-dimensional conformal MOG. (a) The input dataset is comprised by $2{,}321$ vectors geometrically interpreted as points. Here, both spheres where detected, but the coarse discretization of the parameter space leads to a small translation on the sphere at right. (b) The concurrent detection of one plane and one sphere in a set of $3{,}314$ input points, where 45.26% of such input entries characterize noise.

interpreted as $(d-1)$-spheres (*e.g.,* a 1-sphere is a circle, a 2-sphere is an ordinary sphere, and so on) in the conformal MOG. Figure 5.10 presents an example in $\mathbb{R}^2$ (thus, $n = 2 + 2 = 4$). The points $\mathbf{p}_i$ in Figure 5.10a were encoded in conformal MOG and used as input for the detection of circles ($p = 3$). Each point maps to a surface in the 3-dimensional parameter space ($m = 3\,(4-3) = 3$). From the intersection of three or more surfaces one retrieves the circles passing thought three or more input points. The centers of the circles having the smaller radius correspond to the vertices of the Voronoi diagram (points $\mathbf{v}_j$ in Figure 5.10a). As Figure 5.10b shows, such circles reside (in parameter space) on a surface defined by the superposition of mapped input data. Thus, the vertices of the diagram can be retrieve just by looking for the bins having the largest values on that surface (*i.e.,* more than two votes). The bins having two votes correspond to the circles whose centers are at an edge of the Voronoi diagram (the gray points in Figure 5.10a). The votes accumulated by the bins below the green surface in Figure 5.10b are not shown for sake of clarity. In this example, the discretization step for defining the accumulator array is $\pi/720$, and $\omega = 1$.

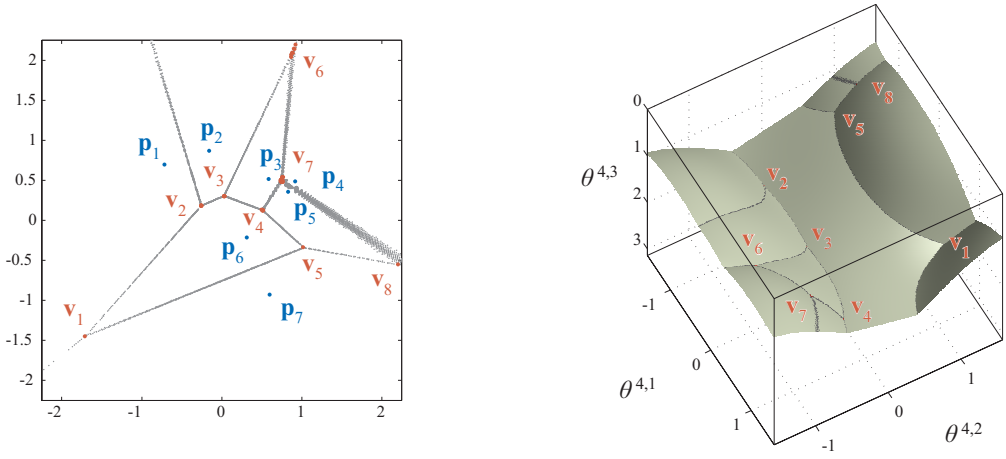
In order to have an accumulator array which approximates the Voronoi diagram, one has to assume the following order for the basis vectors of the representational space:

$$\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_d, \boldsymbol{\infty}, \mathbf{o}\}\,.$$

From this basis, the detection of pseudovectors (*i.e.,* the blades interpreted as $(d-1)$-spheres in conformal MOG) defines

$$\mathbf{E}_{\langle d+1\rangle} = \mathbf{o}^* \tag{5.23}$$

as reference blade (Equation 4.5). $\mathbf{E}_{\langle d+1\rangle}$ is geometrically interpreted as a $(d-1)$-sphere of radius *zero* at the origin of $\mathbb{R}^d$. The dual in Equation 5.23 is computed with respect to the pseudoscalar $\mathbf{I}_{\langle d+2\rangle} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d \wedge \boldsymbol{\infty} \wedge \mathbf{o}$, and it is evaluated under Euclidean, as well as all the computations performed by the proposed technique.



(a) Base space $\mathbb{R}^2$ (b) Accumulator array for $\mathbb{P}^{3\,(4-3)}$

Figure 5.10: The Voronoi diagram of a set of points can be retrieved from the accumulator array produced while performing circle detection with the conformal MOG: (a) The vertices ($\mathbf{v}_j$) and edges (approximated by gray points) of the Voronoi diagram of points $\mathbf{p}_i$ are defined by the center of circles having no points in their interior and passing through more than two, and passing through exactly two input points, respectively. (b) These circles reside on a well defined surface at the parameter space. They can be identified as the bins, in such surface, having more than two and two votes, respectively. The multiple detection of vertices $\mathbf{v}_6$, $\mathbf{v}_7$ and $\mathbf{v}_8$ in (a), and the width of the edge between $\mathbf{p}_4$ and $\mathbf{p}_5$ are related to the quasi-alignment of the surfaces related to $\mathbf{p}_4$ and $\mathbf{p}_5$ in parameter space, leading to multiple intersections while defining the surface depicted in (b).

The subspace detection scheme defines a sequence of $(d+1)$ rotation operations applied to $\mathbf{E}_{\langle d+1\rangle}$. Such rotations happen on planes:

$$\mathbf{P}^{(1)}_{\langle 2\rangle} = \mathbf{o}\wedge\boldsymbol{\infty},\quad \mathbf{P}^{(2)}_{\langle 2\rangle} = \boldsymbol{\infty}\wedge\mathbf{e}_d,\quad \mathbf{P}^{(3)}_{\langle 2\rangle} = \mathbf{e}_d\wedge\mathbf{e}_{d-1},\quad \cdots,\quad \text{and}\quad \mathbf{P}^{(d+1)}_{\langle 2\rangle} = \mathbf{e}_2\wedge\mathbf{e}_1.$$

The first rotation (on plane $\mathbf{P}^{(1)}_{\langle 2\rangle} = \mathbf{o}\wedge\boldsymbol{\infty}$) is closely related to the uniform scale operation. Actually, as the value of rotation angle θ^1 (or $\theta^{4,3}$ in double-index notation) increases from 0 to π, the radius of the $(d-1)$-sphere $\boldsymbol{R}_1\,\mathbf{E}_{\langle d+1\rangle}\,/\,\boldsymbol{R}_1$ also increases. The other rotations (on planes $\mathbf{P}^{(2)}_{\langle 2\rangle}$ to $\mathbf{P}^{(d+1)}_{\langle 2\rangle}$) are geometrically interpreted as translating $\boldsymbol{R}_1\,\mathbf{E}_{\langle d+1\rangle}\,/\,\boldsymbol{R}_1$ on the base space $\mathbb{R}^d$. By mapping input vectors interpreted as points to the parameter space, the voting procedure increments the bins of the accumulator array related to such a mapping. The surface closest to the origin of the parameter space and at the positive side of the θ^1-axis is comprised by the parameter vector of the circles with smaller radius and including at least one of the input points (*i.e.,* they receive at least one vote). The cells related to the faces of the Voronoi diagram are the ones receiving d votes, while the cells representing vertices receive more than d votes.

The idea of using a rasterizing scheme to construct 2-dimensional Voronoi diagrams of points was first suggested by Haeberli (1990), and efficiently implemented in Graphics Hardware by Hoff *et al.* (1999). The approach computes the discrete diagram by Z-buffering right circular cones onto a 2-dimensional canvas. The base of the cones is defined as being parallel to the image plane and the apex points are located at the point sites. By rendering a polygonal approximation of the cones, all distances across the polygonal mesh are represented and stored as depth in a Z-buffer. The Z-buffer depth test compares

the new depth value to the previously stored value. If the new value is less, the Z-buffer records the new distance, and the frame buffer records the site's ID as a unique color assigned to each site. In this way, each pixel in the frame buffer will have a color corresponding to the site to which it is closest, and the depth-buffer will have the distance to that site. After all the cones have being rendered, an approximation of the Voronoi diagram can be retrieved from the boundaries of the resulting image.

The Voronoi example (Figure 5.10) shows that one can restrict the proposed voting procedure to a specific region of the parameter space in order to restrict the detection process to a subset of geometric shapes (*e.g.,* circles with radius in a given range). Note that such a specialization is driven by the geometric properties of the intended shape. However, it does not affect the generality of the proposed approach because the mapping procedure (Section 5.1) is not affected. By specifying a smaller range of interest in the parameter space, the memory and computational requirements of the technique are reduced.

Due to practical issues, the examples presented in this chapter are restricted to the detection of subspaces encoded in the homogeneous and the conformal MOGs, with 2- or 3-dimensional base space. It is because the proof of concept implementations of the described algorithms (Figures 5.2 and 5.4) keep the complete accumulator array allocated in the main memory, as well as some auxiliary data structures. When the accumulator array has more than three dimensions, or when the discretization step is too small, the total memory required by the program can exceed the amount of memory that a process can allocate. In Windows® XP 32-bit, such a limit is 4 GB per program. By increasing the discretization step, a higher-dimensional accumulator array may be allocated in some acceptable amount of memory. However, the representation of the subspaces may be affected by the coarse discretization. This is the case of the detection of subspaces geometrically interpreted as 2-dimensional conic sections. The MOG used to encode such geometric shapes as single blades (Section 3.9.4) is defined oven $\mathbb{R}^6$, where the dimensionality of subspaces interpreted as conic sections is $p = 5$. As a consequence, the space of parameters has $m = 5\,(6 - 5) = 5$ dimensions. The study of solutions for the memory-budget problem is a promising direction for future exploration.

For the results presented in this dissertation, the discretization step of the accumulator arrays was defined according to the number of dimensions of the parameter space. The assumed criteria was set the discretization step as the smallest value that allows the allocation of the accumulator array, while respecting the restrictions imposed by the operating system.

6 VOTING PROCESS FOR INPUT SUBSPACES WITH UNCERTAINTY

Experimental data often contain errors due to imprecisions in the instruments used to collect them. For instance, raster images captured by digital cameras are resolution dependent, and measurements made with a ruler are based on a discrete set of marks. Thus, experimental data always contain some uncertainty, as opposed to being the true measurements.

The uncertainty intrinsic to experimental data can be taken into account while performing subspace detection by supersampling input entries according to their distribution of uncertainty and, in turn, by processing each sample with the technique described in Chapter 5. The quality of sampling-based approaches, however, depends on the number of samples (COWAN, 1998), and the computational load increases as more samples are used. This chapter presents a technique that avoids brute force sampling by predicting how the votes should be spread in the parameter space as a result of the uncertainty in the input data. The spot in Figure 6.1a illustrates $10{,}000$ samples (shown as a histogram) generated by sampling a given input straight line (a 2-blade in homogeneous MOG, thus $r = 2$) with Gaussian distributed uncertainty and mapping the samples to the parameter space used in line detection ($p = 2$, $n = 2 + 1 = 3$, and $m = 2\,(3 - 2) = 2$). The envelope, on the other hand, was computed by the approach described in this chapter. With this approach, the distribution of votes inside the envelope can be efficiently approximated.

The voting scheme presented in this chapter extends the mapping and voting procedures discussed in Chapter 5 by handling input blades with Gaussian distributed uncertainty. The extended mapping procedure is based on first-order error propagation analysis (COWAN, 1998). It transports the uncertainty of each input element throughout the computations into an auxiliary parameter space where the uncertainty is described by a multivariate Gaussian distribution (Figure 6.1b). In turn, such a distribution is mapped to the actual parameter space $\mathbb{P}^m$ by the extended voting procedure, leading to warped (non-Gaussian) distributions of votes in the accumulator array (Figure 6.1a).

A brief introduction to first-order error propagation analysis and how it can be applied to GA equations is presented in Sections 6.1 and 6.2, respectively. Section 6.3 describes the mapping procedures for input r-blades with uncertainty. Section 6.4 shows how to calculate the number of votes to be incremented to each bin in the accumulator array by projecting the bins from $\mathbb{P}^m$ (Figure 6.1a) to the auxiliary space (Figure 6.1b). The warping of the distribution of votes is a natural consequence of such a mapping.

6.1 First-Order Error Propagation

According to the error theory, the error e in some experimental value x is given by the difference of x with respect to the true value x_{true} (COWAN, 1998). Therefore, the error in x is defined as:

$$e = x - x_{\text{true}}.$$

Once x_{true} is unknown, it follows that the error e is also an unknown value. Thus, e can only be estimated with some probability. An experimental uncertain value can be expressed by a random variable $\underline{x}$, where the uncertainty is described by some probability distribution function (PDF). It is assumed in this chapter that the uncertainty is always Gaussian, so the PDF is fully determined by its expectation ($\overline{x}$) and variance (σ_x^2).

If a value y is calculated as a function of parameters x_1, x_1, $\cdots$, x_n, *i.e.,*

$$y = f(x_1, x_1, \cdots, x_n),$$

and the input values have error, then y also has error (COWAN, 1998). By expressing the uncertain data as random scalar variables, it follows that the expectation (mean) of $\underline{y}$ is given by:

$$\overline{y} = f(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n}).$$

For the case where the input variables are not correlated, the variance of $\underline{y}$ can be computed as:

$$\sigma_y^2 = \sum_{i=1}^{n} \left(\left(\frac{\partial y}{\partial x_i} \right)^2 \sigma_{x_i}^2 \right),$$

and when the input variables are correlated, the variance can be approximated using:

$$\sigma_y^2 \approx \sum_{i=1}^{n} \left(\left(\frac{\partial y}{\partial x_i} \right)^2 \sigma_{x_i}^2 \right) + 2 \sum_{i=1}^{n} \left(\frac{\partial y}{\partial x_i} \sum_{j=i+1}^{n} \left(\frac{\partial y}{\partial x_j} \sigma_{x_i,x_j} \right) \right), \tag{6.1}$$

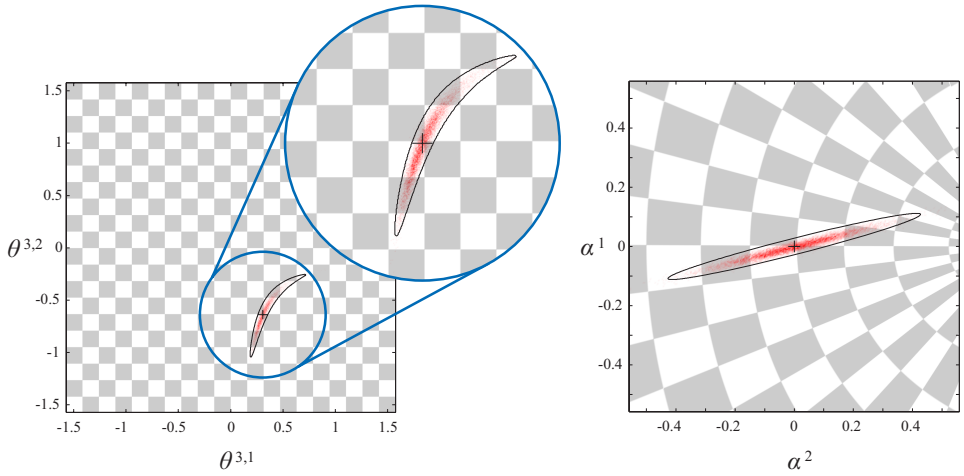
The covariance σ_{x_i,x_j} in Equation 6.1 indicates the degree of correlation of two variables.

Equation 6.1 defines the **first-order error propagation model** for scalar valued random variables with Gaussian distributed uncertainty (COWAN, 1998). The model can be extended to vector-valued random variables. In such a case, Equation 6.1 can be rewritten in the matrix form in order to simplify the notation:

$$\Sigma_{\mathrm{y}} \approx \mathrm{J}_{\mathrm{y}} \, \Sigma_{\mathrm{x}} \, \mathrm{J}_{\mathrm{y}}^{\mathrm{T}}, \tag{6.2}$$

where Σ_{x} and Σ_{y} are the covariance matrices of the vector-valued random variables $\underline{\mathrm{x}}$ and $\underline{\mathrm{y}}$, respectively, and $\square^{\mathrm{T}}$ denotes the transpose of a matrix. In Equation 6.2, J_{y} is the Jacobian matrix for the function $f(\overline{\mathrm{x}}) = \overline{\mathrm{y}}$ that computes the expectation $\overline{\mathrm{y}}$ of $\underline{\mathrm{y}}$ from the terms of the expectation $\overline{\mathrm{x}}$ of $\underline{\mathrm{x}}$. As for random scalar variables, the PDF of a vector-valued random variable with Gaussian distributed uncertainty is fully determined by its expectation and covariance matrix.

It is important to comment that a function of Gaussian distributed random variables may produce a non-Gaussian distribution. First-order error propagation provides, therefore, an approximation of the resulting uncertainty. As pointed out by Cowan (1998) and Perwass (2009), it should be checked whether the resultant distribution can be approximated well by a Gaussian distribution. What amount of error is tolerable depends on the actual application. For cases where the resulting distribution is Gaussian, a better approximation of the uncertainty (or even the correct distribution) can be achieved by extending



(a) Parameter space $\mathbb{P}^{2\,(3-2)}$ (b) Open affine covering $\mathbb{A}^2$ for $G(2,3)$

Figure 6.1: The sampling-based versus the error-propagation-based approach. (a) A visualization of the parameter space $\mathbb{P}^2$ for blades interpreted as straight lines in the 2-dimensional homogeneous MOG. (b) A visualization of the open affine covering $\mathbb{A}^2$ for the Grassmannian of the same kind of subspace. The red spots in (a) and (b) represent the uncertainty of a given straight line. The distribution is shown as a histogram computed by sampling the input line according to its Gaussian distributed uncertainty and mapping each sample to $\mathbb{P}^2$ (a) and $\mathbb{A}^2$ (b). The quality of the sampling-based approach depends on the number of samples. The envelopes in (a) and (b) were defined analytically by the error-propagation-based approach. This approach can efficiently approximate the distribution of uncertainty inside the envelopes.

the first-order error propagation model (Equation 6.2) to higher orders. Note that Equation 6.2 considers only the first-order derivatives of the function computing $\overline{\mathrm{y}}$. In order to obtain the exact result, all derivatives up to order four have to be considered:

$$\Sigma_{\mathrm{y}} = \mathrm{J}_{\mathrm{y}}\,\Sigma_{\mathrm{x}}\,\mathrm{J}_{\mathrm{y}}^{\mathrm{T}} + \mathrm{B}_{\mathrm{y}}(\Sigma_{\mathrm{x}}). \tag{6.3}$$

In Equation 6.3, B_{y} is a function that computes a bias term related to third- and fourth-order derivatives. For functions producing Gaussian distributed uncertain values, the derivatives of orders higher than four have no effect on the error propagation, since they are equal to zero. See (PERWASS, 2009) for a definition of B_{y} to bilinear products in GA.

Latter, in this chapter, it is shown that the first-order error propagation without the bias term (Equation 6.2) fits well to the proposed subspace detection framework. It is because input entries are processed individually by the mapping and voting procedures. In this case, if an input entry has Gaussian distributed uncertainty, such uncertainty will be kept Gaussian up to a well known stage of the computational flow describe in Section 6.3 (*i.e.,* the auxiliary parameter space). After that, the distribution is converted to non-Gaussian in a natural way through the correspondence between the auxiliary and the actual parameter space.

6.2 Tensor Representation of Geometric Algebra Operations

First-order error propagation (Section 6.1) can be applied to GA equations by expressing **multivectors as component vectors** and **GA operations as tensor contrac-**

tions (PERWASS, 2009). With such a representation, the Jacobian matrix in Equation 6.2 can be calculated as for conventional linear algebra equations.

In order to express multivectors as component vectors, let $\{\mathbf{E}_i\}_{i=1}^{2^n}$ be the basis of a multivector space $\bigwedge \mathbb{R}^n$. For the example in Equation 3.1:

$$\begin{aligned}
&\mathbf{E}_1 = 1, && \mathbf{E}_2 = \mathbf{e}_1, && \mathbf{E}_3 = \mathbf{e}_2,\\
&\mathbf{E}_4 = \mathbf{e}_3, && \mathbf{E}_5 = \mathbf{e}_1 \wedge \mathbf{e}_2, && \mathbf{E}_6 = \mathbf{e}_1 \wedge \mathbf{e}_3,\\
&\mathbf{E}_7 = \mathbf{e}_2 \wedge \mathbf{e}_3, && \mathbf{E}_8 = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3. &&
\end{aligned}$$

A multivector $M \in \bigwedge \mathbb{R}^n$, and hence a blade or a rotor, may then be written as:

$$M = \sum_{i=1}^{2^n} \left(\mu^i \, \mathbf{E}_i\right),$$

where μ^i is the i-th component of a vector in $\mathbb{R}^{2^n}$.

The geometric product between two multivectors A and B may be written in terms of a tensor contraction:

$$C = A\,B = \sum_{i,j,k=1}^{2^n} \left(\alpha^j \, \beta^k \, \Gamma^{i,j,k} \, \mathbf{E}_i\right), \tag{6.4}$$

where $\mathbf{E}_i$ is the i-th basis element of $\bigwedge \mathbb{R}^n$, $\{\alpha^j\}_{j=1}^{2^n}$ and $\{\beta^k\}_{k=1}^{2^n}$ are the coefficients of A and B, respectively, and Γ is a 3rd-rank tensor encoding the geometric product (Section 3.5). If $C = \sum_{i=1}^{2^n} (\gamma^i \, \mathbf{E}_i)$ then the relation among $\{\gamma^i\}_{i=1}^{2^n}$, $\{\alpha^j\}_{j=1}^{2^n}$, and $\{\beta^k\}_{k=1}^{2^n}$ is:

$$\gamma^i = \sum_{j,k=1}^{2^n} \left(\alpha^j \, \beta^k \, \Gamma^{i,j,k}\right) \quad \forall\, i \in \{1, 2, \cdots, 2^n\}.$$

The computation of the entries of Γ is performed by evaluating the product between pairs of basis elements, *i.e.*:

$$\mathbf{E}_j \, \mathbf{E}_k = \Gamma^{i,j,k} \, \mathbf{E}_i \quad \forall\, j, k \in \{1, 2, \cdots, 2^n\}. \tag{6.5}$$

Notice that entries of the tensor Γ (Equation 6.5) do not depend on the arguments A and B in Equation 6.4. Thus, Γ can be treated as a constant term while computing the derivatives in the Jacobian matrix used by the first-order error propagation model.

The same principle depicted in Equation 6.5 can be used to compute a different tensor for each bilinear product presented in Sections 3.3, 3.4 and 3.5. This is achieved just by replacing the geometric product in Equation 6.5 by the intended product. In Section 6.3, **the tensor representation of the geometric product and left contraction are denoted by Γ and Λ, respectively**.

Some unary operations may be expressed by 2nd-tensors (*i.e.,* matrices). In Section 6.3, Υ **denotes the tensor representation of the reverse operation** (Equation 3.7):

$$\widetilde{A} = \sum_{i=1}^{2^n} \left(\alpha^i \, \Upsilon^{i,i} \, \mathbf{E}_i\right).$$

Note that Υ is a diagonal matrix whose entries are either 1 or -1.

6.3 Mapping Procedure

First-order error propagation (Section 6.1) provides a good approximation for Gaussian distributed uncertainty (COWAN, 1998; PERWASS, 2009). However, Figure 6.1a clearly shows that the resulting distribution of votes in the parameter space is non-Gaussian. For instance, it is not symmetric around the mean (marked with a $+$ in Figure 6.1a), and the main axes are bent. Hence, standard first-order error propagation cannot be applied directly on the computation chain of the mapping procedure presented in Sections 5.1 and 5.2. **The technique described in current section avoids such an issue by including an alternative computation flow to propagate the uncertainty through the procedures in Figure 6.2 and Figure 6.3**. The alternative computation represents a resulting parameter vector $\Theta^{(0)}$ (Figure 6.2) mapped as the point at the origin of a m-dimensional open affine covering $\mathbb{A}^m$ for the Grassmannian $G(p,n)$ (the Grassmannian was introduced in Section 4.2). This way, the uncertainty of $\Theta^{(0)}$ is described by a multivariate Gaussian distribution at the origin of $\mathbb{A}^m$. Figure 6.1b illustrates the affine space and the probability distribution for the example depicted in Figure 6.1a.

6.3.1 Mapping Procedure for $r \geq p$

The procedure that maps input r-blades with uncertainty to the parameter space $\mathbb{P}^m$ characterizing p-dimensional subspaces is presented in Figure 6.2, for $r \geq p$. Its complementary $CalculateParameter_{\text{un.}}$ function is presented in Figure 6.3. The algorithm takes as input a random multivector variable $\mathbf{X}_{\langle r \rangle}$, whose expectation $\overline{\mathbf{X}_{\langle r \rangle}}$ is a blade and covariance matrix is $\Sigma_{\mathbf{X}_{\langle \mathrm{r} \rangle}}$. The procedure returns a set of 2-tuples comprised by a parameter vector $\Theta^{(0)} \in \mathbb{P}^m$ (Figure 6.2, line 10), and a vector-valued random variable $\underline{\mathrm{a}}$. By definition, the expectation of $\underline{\mathrm{a}}$ is at the origin of $\mathbb{A}^m$ (*i.e.,* $\overline{\mathrm{a}} = (0, 0, \cdots, 0)^{\mathrm{T}} \in \mathbb{A}^m$). The covariance matrix of $\underline{\mathrm{a}}$ is computed with the first-order error propagation model (Equation 6.2):

$$\Sigma_{\mathrm{a}} = \mathrm{J}_{\mathrm{a}} \, \Sigma_{\mathbf{X}_{\langle \mathrm{r} \rangle}} \, \mathrm{J}_{\mathrm{a}}^{\mathrm{T}}. \tag{6.6}$$

In order to evaluate Equation 6.6, one needs to compute the Jacobian matrix J_{a} for the equation that calculates the mean point $\overline{\mathrm{a}}$ in terms of the coefficients of the input mean blade $\overline{\mathbf{X}_{\langle r \rangle}}$. However, expressing the entire computation chain as a single equation, and from it computing the Jacobian matrix, turns out to be impractical. Note that intermediate variables can be combined in different ways. The combination depends on which parameters must be arbitrated and which ones must be computed while mapping the input data to the parameter space. As a result, the definition of J_{a} must handle all possible computation flows. The proposed solution for this problem is to solve the partial derivatives in matrix J_{a} using the chain rule, step-by-step, until the final result is found. In Figure 6.2 and Figure 6.3 the derivatives of intermediate computation steps are kept as the Jacobian matrices of intermediate variables. The following derivations show how these matrices are evaluated.

The extended mapping procedure starts by initializing a set $\mathcal{P}^{(m)}$ (Figure 6.2, line 1) with a 5-tuple:

$$(\mathbf{X}_{\langle r \rangle}^{(m)}, \mathrm{J}_{\mathbf{X}_{\langle r \rangle}^{(m)}}, \boldsymbol{K}^{(m)}, \mathrm{J}_{\boldsymbol{K}^{(m)}}, \Theta^{(m)}) \in \mathcal{P}^{(m)}, \tag{6.7}$$

where $\mathbf{X}_{\langle r \rangle}^{(m)} = \overline{\mathbf{X}_{\langle r \rangle}}$ is the input (mean) blade, $\mathrm{J}_{\mathbf{X}_{\langle r \rangle}^{(m)}} = \mathrm{I}$ is the Jacobian matrix of $\mathbf{X}_{\langle r \rangle}^{(m)}$ with respect to $\mathbf{X}_{\langle r \rangle}$ (*i.e.,* an identity matrix), and $\boldsymbol{K}^{(m)} = 1$ is an identity rotor denoting that no rotor $\boldsymbol{R}_t$ was computed yet. In subsequent steps of the algorithm, $\boldsymbol{K}^{(t)}$ is a ro-

Require: A random multivector variable $\underline{\mathbf{X}_{\langle r \rangle}}$ with expectation $\overline{\mathbf{X}_{\langle r \rangle}}$ and covariance matrix $\Sigma_{\mathbf{X}_{\langle r \rangle}}$

1: $\mathcal{P}^{(m)} \leftarrow \left\{(\overline{\mathbf{X}_{\langle r \rangle}}, \mathrm{I}, 1, 0, \varnothing)\right\}$
2: **for** $t = m$ **down to** 1 **do**
3: Let $\mathbf{P}^{(t)}_{\langle 2 \rangle}$ be the rotation plane of the t-th rotation applied to $\mathbf{E}_{\langle p \rangle}$ in Equation 5.3
4: $\mathcal{P}^{(t-1)} \leftarrow \varnothing$
5: **for all** $(\mathbf{X}^{(t)}_{\langle r \rangle}, \mathrm{J}_{\mathbf{X}^{(t)}_{\langle r \rangle}}, \boldsymbol{K}^{(t)}, \mathrm{J}_{\boldsymbol{K}^{(t)}}, \Theta^{(t)}) \in \mathcal{P}^{(t)}$ **do**
6: $\mathcal{T} \leftarrow CalculateParameter_{\text{un.}}(\mathbf{X}^{(t)}_{\langle r \rangle}, \mathrm{J}_{\mathbf{X}^{(t)}_{\langle r \rangle}})$
7: $\mathcal{P}^{(t-1)} \leftarrow \mathcal{P}^{(t-1)} \cup \big\{(\mathbf{X}^{(t-1)}_{\langle r \rangle}, \mathrm{J}_{\mathbf{X}^{(t-1)}_{\langle r \rangle}}, \boldsymbol{K}^{(t-1)}, \mathrm{J}_{\boldsymbol{K}^{(t-1)}}, (\theta^t, \Theta^{(t)}_1, \Theta^{(t)}_2, \cdots, \Theta^{(t)}_{m-t})) \mid$
$\mathbf{X}^{(t-1)}_{\langle r \rangle} = \widetilde{\boldsymbol{R}}_t \, \mathbf{X}^{(t)}_{\langle r \rangle} \, \boldsymbol{R}_t$, and $\boldsymbol{K}^{(t-1)} = \boldsymbol{K}^{(t)} \, \boldsymbol{R}_t$,
and $\boldsymbol{R}_t = \cos\left(\frac{\theta^t}{2}\right) - \sin\left(\frac{\theta^t}{2}\right) \mathbf{P}^{(t)}_{\langle 2 \rangle}$, and $(\theta^t, \mathrm{J}_{\theta^t}) \in \mathcal{T}\big\}$
8: **end for**
9: **end for**
10: **return** $\big\{(\Theta^{(0)}, \underline{\mathrm{a}}) \mid \overline{\mathrm{a}}$ is computed according to Equation 6.17,
$\Sigma_{\mathrm{a}} = \mathrm{J}_{\mathrm{a}} \, \Sigma_{\mathbf{X}_{\langle r \rangle}} \, \mathrm{J}_{\mathrm{a}}^{\mathrm{T}}$, and $(\mathbf{X}^{(0)}_{\langle r \rangle}, \mathrm{J}_{\mathbf{X}^{(0)}_{\langle r \rangle}}, \boldsymbol{K}^{(0)}, \mathrm{J}_{\boldsymbol{K}^{(0)}}, \Theta^{(0)}) \in \mathcal{P}^{(0)}\big\}$

Figure 6.2: The procedure that extends the algorithm presented in Figure 5.2 to blades with uncertainty. It takes as input an random r-blade $\underline{\mathbf{X}_{\langle r \rangle}}$ and returns a set of pairs comprised by a parameter vector $\Theta^{(0)} \in \mathbb{P}^m$ characterizing a p-blades that is contained by $\overline{\mathbf{X}_{\langle r \rangle}}$, and a vector-valued random variable $\underline{\mathrm{a}}$ describing the Gaussian uncertainty of the p-blade represented as the origin of the open affine covering of the Grassmannian.

tor composite by the geometric product of the last $(m - t)$ rotors $\boldsymbol{R}_t$ applied to $\mathbf{E}_{\langle p \rangle}$ in Equation 5.3, *i.e.,*

$$\boldsymbol{K}^{(t)} = \boldsymbol{K}^{(t+1)} \, \boldsymbol{R}_{t+1},$$

for $1 \leq t \leq m$, and $\boldsymbol{K}^{(m+1)} = 1$. At the end of the mapping process, $\boldsymbol{K}^{(0)} = \boldsymbol{T}$ ($\boldsymbol{T}$ is defined in Equation 4.8) is the rotor used to transform the reference blade $\mathbf{E}_{\langle p \rangle}$ into the blade characterized by the resulting parameter vector $\Theta^{(0)}$ (line 10). In Equation 6.7, $\mathrm{J}_{\boldsymbol{K}^{(m)}} = 0$ is the Jacobian matrix of $\boldsymbol{K}^{(m)}$ (*i.e.,* a zero row matrix), and $\Theta^{(m)} = \varnothing$ is an empty set denoting that no parameter was calculated yet.

At each iteration of the procedure (lines 2 to 9), the function $CalculateParameter_{\text{un.}}$ (called in line 6 of Figure 6.2 and defined in Figure 6.3) returns a set $\mathcal{T}$ of 2-tuples:

$$(\theta^t, \mathrm{J}_{\theta^t}) \in \mathcal{T},$$

where θ^t is the t-th parameter of some p-blade related to $\overline{\mathbf{X}_{\langle r \rangle}}$, and J_{θ^t} is its Jacobian matrix, whose definition is presented later in Equation 6.29. The rotation angle θ^t is used in line 7 of Figure 6.2 to compute the rotor $\boldsymbol{R}_t$ as:

$$\begin{aligned} \boldsymbol{R}_t &= \cos\left(\frac{\theta^t}{2}\right) - \sin\left(\frac{\theta^t}{2}\right) \mathbf{P}^{(t)}_{\langle 2 \rangle} \\ &= \cos\left(\frac{\theta^t}{2}\right) - \sin\left(\frac{\theta^t}{2}\right) \sum_{i=1}^{2^n} \left(\phi^i_t \, \mathbf{E}_i\right), \end{aligned} \tag{6.8}$$

where $\mathbf{P}_{\langle 2 \rangle}^{(t)}$ is a constant rotation plane with coefficients $\{\phi_t^i\}_{i=1}^{2^n}$, leading to the Jacobian matrix:

$$\mathrm{J}_{\boldsymbol{R}_t}^{i,z} = \frac{\partial \rho_t^i}{\partial \chi^z} = -\frac{1}{2}\,\mathrm{J}_{\theta^t}^{i,z} \begin{cases} \sin\left(\frac{\theta^t}{2}\right) & \text{for } i = 1 \\ \phi_t^i \cos\left(\frac{\theta^t}{2}\right) & \text{otherwise} \end{cases}. \tag{6.9}$$

Following the tensor representation introduced in Section 6.2, $\{\rho_t^i\}_{i=1}^{2^n}$ and $\{\chi^z\}_{z=1}^{2^n}$ in Equation 6.9 denote the coefficients of, respectively, $\boldsymbol{R}_t$ and $\overline{\mathbf{X}_{\langle r \rangle}}$.

The rotor $\boldsymbol{R}_t$ is used in line 7 to compute

$$\begin{aligned} \mathbf{X}_{\langle r \rangle}^{(t-1)} &= \widetilde{\boldsymbol{R}}_t\, \mathbf{X}_{\langle r \rangle}^{(t)}\, \boldsymbol{R}_t \\ &= \sum_{i,j,k,l=1}^{2^n} \left(\rho_t^j\, \lambda_t^k\, \rho_t^l\, \Psi^{i,j,k,l}\, \mathbf{E}_i\right), \end{aligned} \tag{6.10}$$

where $\{\rho_t^i\}_{i=1}^{2^n}$ and $\{\lambda_t^i\}_{i=1}^{2^n}$ denote the coefficients of $\boldsymbol{R}_t$ and $\mathbf{X}_{\langle r \rangle}^{(t)}$, respectively. The tensor

$$\Psi^{i,j,k,l} = \Upsilon^{j,j} \sum_{h=1}^{2^n} \left(\Gamma^{h,j,k}\, \Gamma^{i,h,l}\right) \tag{6.11}$$

is comprised by constant values computed from the tensors Γ and Υ encoding, respectively, the geometric product and the reverse operation. The derivatives in the Jacobian matrix of $\mathbf{X}_{\langle r \rangle}^{(t-1)}$ (Figure 6.2, line 7) are given by:

$$\mathrm{J}_{\mathbf{X}_{\langle r \rangle}^{(t-1)}}^{i,z} = \frac{\partial \lambda_{t-1}^i}{\partial \chi^z} = \sum_{j,k,l=1}^{2^n} \left(\rho_t^j\, \mathrm{J}_{\mathbf{X}_{\langle l \rangle}^{(t)}}^{k,z}\, \rho_t^l\, \Psi^{i,j,k,l}\right) + \sum_{j,k,l=1}^{2^n} \left(\lambda_t^k \left(\mathrm{J}_{\boldsymbol{R}_t}^{j,z}\, \rho_t^l + \rho_t^j\, \mathrm{J}_{\boldsymbol{R}_t}^{l,z}\right) \Psi^{i,j,k,l}\right), \tag{6.12}$$

The Jacobian matrix of $\boldsymbol{R}_t$ ($\mathrm{J}_{\boldsymbol{R}_t}$, in the summation at the right side of Equation 6.12) is defined in Equation 6.9.

The rotor $\boldsymbol{R}_t$ is also used in Figure 6.2 (line 7) to compute:

$$\begin{aligned} \boldsymbol{K}^{(t-1)} &= \boldsymbol{K}^{(t)}\, \boldsymbol{R}_t \\ &= \sum_{i,j,k=1}^{2^n} \left(\kappa_t^j\, \rho_t^k\, \Gamma^{i,j,k}\, \mathbf{E}_i\right). \end{aligned} \tag{6.13}$$

The coefficients of $\boldsymbol{K}^{(t-1)}$ are denoted by $\{\kappa_{t-1}^i\}_{i=1}^{2^n}$ and its Jacobian matrix is defined as:

$$\mathrm{J}_{\boldsymbol{K}^{(t-1)}}^{i,z} = \frac{\partial \kappa_{t-1}^i}{\partial \chi^z} = \sum_{j,k=1}^{2^n} \left(\left(\mathrm{J}_{\boldsymbol{K}^{(t)}}^{j,z}\, \rho_t^k + \kappa_t^j\, \mathrm{J}_{\boldsymbol{R}_t}^{k,z}\right) \Gamma^{i,j,k}\right). \tag{6.14}$$

After all the parameters θ^t of the p-blades related to $\overline{\mathbf{X}_{\langle r \rangle}}$ have been calculated, one also has computed their respective rotors $\boldsymbol{K}^{(0)} = \boldsymbol{T}$. Recall from Equation 5.3 that a rotor $\boldsymbol{T}$ transforms the reference blade $\mathbf{E}_{\langle p \rangle}$ into the blade $\mathbf{C}_{\langle p \rangle}$ related to a given parameter vector $\Theta^{(0)}$. The last step of the mapping procedure is to define the open affine covering $\mathbb{A}^m$ for the Grassmannian (Section 4.2) in such a way that $\mathbf{C}_{\langle p \rangle}$ (and $\Theta^{(0)}$) is represented as the point at the origin of $\mathbb{A}^m$. Such origin point is denoted by $\overline{\mathrm{a}}$. The computation of its coordinates leads to the Jacobian matrix $\mathrm{J_a}$ (Equation 6.18) used in Equation 6.6 to compute the covariance matrix of a vector-valued random variable $\underline{\mathrm{a}}$ (line 10).

As a point at the origin, the coordinates of $\overline{\mathrm{a}}$ are equal to *zero*. According to Equation 4.19, the origin of $\mathbb{A}^m$ is actually related to the blade $\mathbf{A}_{\langle p \rangle} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_p$, and not to an arbitrary blade $\mathbf{C}_{\langle p \rangle}$. Thus, a mapping from $\mathbf{C}_{\langle p \rangle}$ to $\mathbf{A}_{\langle p \rangle}$ must be defined. **The rotor**

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{T}_A \, \widetilde{\boldsymbol{K}}^{(0)} \\ &= \sum_{i,j,k=1}^{2^n} \left(\zeta^j \, \kappa_0^k \, \Upsilon^{k,k} \, \Gamma^{i,j,k} \, \mathbf{E}_i \right) \end{aligned} \tag{6.15}$$

performs a change of basis, mapping $\mathbf{C}_{\langle p \rangle}$ to $\mathbf{A}_{\langle p \rangle}$ (*i.e.,* $\boldsymbol{W} \, \mathbf{C}_{\langle p \rangle} \, \widetilde{\boldsymbol{W}} = \mathbf{A}_{\langle p \rangle}$). In Equation 6.15, $\boldsymbol{T}_A$ is the rotor that transforms $\mathbf{E}_{\langle p \rangle}$ into $\mathbf{A}_{\langle p \rangle}$. Its coefficients are denoted as $\{\zeta^j\}_{j=1}^{2^n}$. Notice that $\boldsymbol{T}_A$ may be precomputed from the parameter vector returned by the procedure in Figure 5.2 (or Figure 6.2) when $\mathbf{A}_{\langle p \rangle}$ is given as input.

The Jacobian matrix of $\boldsymbol{W}$ (Equation 6.15) is computed as:

$$\mathrm{J}_{\boldsymbol{W}}^{i,z} = \frac{\partial \omega^i}{\partial \chi^z} = \sum_{j,k=1}^{2^n} \left(\zeta^j \, \mathrm{J}_{\boldsymbol{K}^{(0)}}^{k,z} \, \Gamma^{i,j,k} \right), \tag{6.16}$$

where $\mathrm{J}_{\boldsymbol{K}^{(0)}}$ is given by Equation 6.14.

Finally, the coordinates of $\overline{\mathrm{a}}$ (denoted by $\alpha^{i,j}$ in Equation 4.19 and by α^t in current section) are computed as:

$$\alpha^t = \left(\boldsymbol{W} \, \mathbf{c}_i \, \widetilde{\boldsymbol{W}} \right) * \mathbf{e}_{p+j} = 0, \tag{6.17}$$

where $t = (i-1)\,(n-p) + j$, for $i \in \{1, 2, \cdots, p\}$ and $j \in \{1, 2, \cdots, n-p\}$. In Equation 6.17, the vector

$$\boldsymbol{W} \, \mathbf{c}_i \, \widetilde{\boldsymbol{W}} = \mathbf{e}_i + \sum_{j=1}^{n-p} \left(\alpha^{i,j} \, \mathbf{e}_{p+j} \right) = \mathbf{e}_i$$

is at the i-th row of the matrix representation of $\mathbf{A}_{\langle p \rangle}$ in row reduced echelon form (Equation 4.19), and

$$\mathbf{c}_i = \widetilde{\boldsymbol{W}} \, \mathbf{e}_i \, \boldsymbol{W}$$

is the i-th vector spanning $\mathbf{C}_{\langle p \rangle} = \mathbf{c}_1 \wedge \cdots \wedge \mathbf{c}_i \wedge \cdots \wedge \mathbf{c}_p$.

From Equation 6.17, it follows that the coordinates $\{\alpha^t\}_{t=1}^m$ of $\overline{\mathrm{a}}$ can be rewritten in tensor form as:

$$\alpha^t = \sum_{h,i=1}^{2^n} \left(\epsilon_{\ell_2}^h \, \Lambda^{1,i,h} \sum_{j,k,l=1}^{2^n} \left(\omega^j \, \gamma_{\ell_1}^k \, \omega^l \, \Xi^{i,j,k,l} \right) \right),$$

leading to the Jacobian matrix:

$$\mathrm{J}_{\mathrm{a}}^{t,z} = \frac{\partial \alpha^t}{\partial \chi^z} = \sum_{h,i=1}^{2^n} \left(\epsilon_{\ell_2}^h \, \Lambda^{1,i,h} \sum_{j,k,l=1}^{2^n} \left(\gamma_{\ell_1}^k \, \left(\mathrm{J}_{\boldsymbol{W}}^{j,z} \, \omega^l + \omega^j \, \mathrm{J}_{\boldsymbol{W}}^{l,z} \right) \, \Xi^{i,j,k,l} \right) \right), \tag{6.18}$$

where $\ell_1 = \lceil \frac{t}{n-p} \rceil$ and $\ell_2 = t + n - \lceil \frac{t}{n-p} \rceil \, (n-p)$ are indices computed from t, and $\lceil \square \rceil$ denotes the ceiling function. The Jacobian matrix $\mathrm{J}_{\boldsymbol{W}}$ used in Equation 6.18 is defined in Equation 6.16. The constant values $\{\gamma_{\ell_1}^k\}_{k=1}^{2^n}$ and $\{\epsilon_{\ell_2}^h\}_{h=1}^{2^n}$ are the coefficients of $\mathbf{c}_{\ell_1}$ and $\mathbf{e}_{\ell_2}$, respectively. The constant Ξ is given by

$$\Xi^{i,j,k,l} = \Upsilon^{l,l} \sum_{h=1}^{2^n} \left(\Gamma^{h,j,k} \, \Gamma^{i,h,l} \right).$$

Require: $\mathbf{Y}^{(t)}$, the current input blade
Require: $\mathrm{J}_{\mathbf{Y}^{(t)}}$, the Jacobian matrix of $\mathbf{Y}^{(t)}$ with respect to $\underline{\mathbf{X}_{\langle r \rangle}}$

1: Let $\mathbf{P}^{(t)}_{\langle 2 \rangle}$ be the rotation plane of the t-th rotation applied to $\mathbf{E}_{\langle p \rangle}$ in Equation 5.3
2: Let $\mathbf{F}^{(t)}_{l}$ be a space of possibilities as defined in Equation 5.5
3: Let $\mathbf{r}^{(t)}_{l} \leftarrow \mathbf{F}^{(t-1)}_{l} \rfloor \mathbf{F}^{(t)}_{l}$, *i.e.,* the vector factor in $\mathbf{F}^{(t)}_{l}$ that is not in $\mathbf{F}^{(t-1)}_{l}$
4: **loop**
5: $\quad \mathcal{M} \leftarrow \left\{(\mathbf{M}^{(t)}_{l}, \mathrm{J}_{\mathbf{M}^{(t)}_{l}}) \mid \mathbf{M}^{(t)}_{l} = \mathbf{Y}^{(t)} \cap \mathbf{F}^{(t)}_{l},\ l \in \mathbb{Z}, \text{ and } 1 \leq l \leq |\mathcal{E}|\right\}$
6: $\quad \mathcal{N} \leftarrow \left\{(\mathbf{M}^{(t)}_{l}, \mathrm{J}_{\mathbf{M}^{(t)}_{l}}) \mid (\mathbf{M}^{(t)}_{l}, \mathrm{J}_{\mathbf{M}^{(t)}_{l}}) \in \mathcal{M}, \text{ and } \mathrm{grade}\,(\mathbf{M}^{(t)}_{l}) = |\mathcal{S}|\,,\right.$
$\qquad\qquad \left. \text{where } \mathcal{S} \leftarrow \left\{\mathbf{M}^{(t)}_{h} \mid (\mathbf{M}^{(t)}_{h}, \mathrm{J}_{\mathbf{M}^{(t)}_{h}}) \in \mathcal{M}, \text{ and } \mathbf{M}^{(t)}_{l} * \mathbf{M}^{(t)}_{h} \neq 0\right\}\right\}$
7: $\quad$ **if** $\mathcal{N} = \varnothing$ **then**
8: $\quad\quad$ **return** $\left\{(\theta^t, 0) \mid \theta^t \in [-\pi/2, \pi/2)\right\}$
9: $\quad$ **end if**
10: $\quad \mathcal{O} \leftarrow \left\{(\mathbf{M}^{(t)}_{l}, \mathrm{J}_{\mathbf{M}^{(t)}_{l}}) \mid (\mathbf{M}^{(t)}_{l}, \mathrm{J}_{\mathbf{M}^{(t)}_{l}}) \in \mathcal{N}, \text{ and } \mathrm{grade}\,(\mathbf{M}^{(t)}_{l}) = \mathrm{grade}\,(\mathbf{r}^{(t)}_{l}) = 1\right\}$
11: $\quad \mathcal{Q} \leftarrow \left\{(\mathbf{q}^{(t)}_{l}, \mathrm{J}_{\mathbf{q}^{(t)}_{l}}) \mid \mathbf{q}^{(t)}_{l} = (\mathbf{m}^{(t)}_{l} \rfloor \mathbf{P}^{(t)}_{\langle 2 \rangle}), \text{ and } (\mathbf{m}^{(t)}_{l}, \mathrm{J}_{\mathbf{m}^{(t)}_{l}}) \in \mathcal{O}, \text{ and } \mathbf{q}^{(t)}_{l} \neq 0\right\}$
12: $\quad$ **if** $\mathcal{Q} \neq \varnothing$ **then**
13: $\quad\quad$ **return** $\left\{(\theta^t, \mathrm{J}_{\theta^t}) \mid \theta^t = \tan^{-1}\left(((\mathbf{q}^{(t)}_{l} \wedge \mathbf{r}^{(t)}_{l}) * \mathbf{P}^{(t)}_{\langle 2 \rangle}) \,/\, (\mathbf{q}^{(t)}_{l} * \mathbf{r}^{(t)}_{l})\right),\right.$
$\qquad\qquad\qquad \left. \text{where } (\mathbf{q}^{(t)}_{l}, \mathrm{J}_{\mathbf{q}^{(t)}_{l}}) \text{ is one tuple in } \mathcal{Q}\right\}$
14: $\quad$ **end if**
15: $\quad \mathbf{Y}^{(t)} \leftarrow (\mathbf{M}^{(t)}_{l})^{-1} \rfloor \mathbf{Y}^{(t)}$, where $(\mathbf{M}^{(t)}_{l}, \mathrm{J}_{\mathbf{M}^{(t)}_{l}}) \in \mathcal{N}$ and $\mathbf{M}^{(t)}_{l}$ has the highest dimensionality in the set $\mathcal{N}$
16: $\quad \mathrm{J}_{\mathbf{Y}^{(t)}} \leftarrow$ the Jacobian matrix of $\mathbf{Y}^{(t)}$ (line 15) with respect of $\underline{\mathbf{X}_{\langle r \rangle}}$
17: **end loop**

Figure 6.3: Function $CalculateParameter_{\text{un.}}$. It complements the mapping procedure in Figure 6.2. The algorithm extends the procedure presented in Figure 5.4 by computing the Jacobian matrix of the intermediate variables with respect to the coefficients of the input variable $\underline{\mathbf{X}_{\langle r \rangle}}$ in Figure 6.2.

The scalar values $\{\alpha^t\}_{t=1}^{m}$ are computed using a scalar product (see the $*$ operator in Equation 6.17). However, the tensor representation of the left contraction (*i.e.,* Λ) has been used while computing their derivatives in Equation 6.18. It is because the left contraction (Section 3.4.2) reduces to the scalar product (Section 3.4.1) when applied to blades with the same dimensionality. In such a case, only the first coefficients of the resulting multivector need to be evaluated. Note in Equation 6.18 that the very first index of Λ has been set to 1.

6.3.2 The $CalculateParameter_{\text{un.}}$ Function

Figure 6.3 complements the procedure presented in Figure 6.2. It takes as input the blade $\mathbf{Y}^{(t)}$ and the Jacobian matrix $\mathrm{J}_{\mathbf{Y}^{(t)}}$, computed in Figure 6.2 as $\mathbf{X}^{(t)}_{\langle r \rangle}$ and $\mathrm{J}_{\mathbf{X}^{(t)}_{\langle r \rangle}}$, respectively (see Figure 6.2, line 6).

The meet operation in Figure 6.3 (line 5) must be evaluated in terms of the pseu-

doscalar $\mathbf{I}_l^{(t)}$ of the space where the intersection problem resides (*i.e.,* $\mathbf{I}_l^{(t)} = \mathbf{Y}^{(t)} \cup \mathbf{F}_l^{(t)}$). By doing so, the meet reduces to the application of two left contraction:

$$\begin{aligned}\mathbf{M}_l^{(t)} = \mathbf{Y}^{(t)} \cap \mathbf{F}_l^{(t)} = \left(\mathbf{F}_l^{(t)} \lrcorner (\mathbf{I}_l^{(t)})^{-1}\right) \lrcorner \mathbf{Y}^{(t)} = \mathbf{D}_l^{(t)} \lrcorner \mathbf{Y}^{(t)} \\ = \sum_{i,j,k=1}^{2^n} \left(\delta_{t,l}^j \, \eta_t^k \, \Lambda^{i,j,k} \, \mathbf{E}_i\right).\end{aligned} \tag{6.19}$$

In Equation 6.19, $\mathbf{D}_l^{(t)} = \mathbf{F}_l^{(t)} \lrcorner (\mathbf{I}_l^{(t)})^{-1}$ is a constant blade. The derivatives in the Jacobian of $\mathbf{M}_l^{(t)}$ are given by:

$$\mathrm{J}_{\mathbf{M}_l^{(t)}}^{i,z} = \frac{\partial \mu_{t,l}^i}{\partial \chi^z} = \sum_{j,k=1}^{2^n} \left(\delta_{t,l}^j \, \mathrm{J}_{\mathbf{Y}^{(t)}}^{k,z} \, \Lambda^{i,j,k}\right). \tag{6.20}$$

In Equations 6.19 and 6.20, the coefficients of $\mathbf{M}_l^{(t)}$, $\mathbf{D}_l^{(t)}$ and $\mathbf{Y}^{(t)}$ are denoted by $\{\mu_{t,l}^i\}_{i=1}^{2^n}$, $\{\delta_{t,l}^j\}_{j=1}^{2^n}$ and $\{\eta_t^k\}_{k=1}^{2^n}$, respectively.

Recall from Section 5.1 that $\mathcal{N}$ (Figure 5.4, line 6) is a subset of $\mathcal{M}$ (Figure 5.4, line 5). When $\mathcal{N}$ is empty (Figure 5.4, line 7), θ^t can assume any value in the $[-\pi/2,\pi/2)$ range. Hence, in Figure 6.3 line 8, the Jacobian matrix of θ^t (*i.e.,* the second element in resulting tuples) is a zero row matrix because the θ^t values do not depend on the input blade.

When $\mathcal{N}$ is not empty, $\mathcal{O}$ is computed as a subset of $\mathcal{N}$ (Figure 6.3, line 10). In turn, $\mathcal{Q}$ (line 11) is defined as the set of 2-tuples $(\mathbf{q}_l^{(t)}, \mathrm{J}_{\mathbf{q}_l^{(t)}})$, where $\mathbf{q}_l^{(t)}$ is a nonzero vector resulting from the contraction of vectors $\mathbf{m}_l^{(t)} \in \mathcal{O}$ onto the rotation plane $\mathbf{P}_{\langle 2\rangle}^{(t)}$:

$$\begin{aligned}\mathbf{q}_l^{(t)} = \mathbf{m}_l^{(t)} \lrcorner \mathbf{P}_{\langle 2\rangle}^{(t)} \\ = \sum_{i,j,k=1}^{2^n} \left(\mu_{t,l}^j \, \phi_t^k \, \Lambda^{i,j,k} \, \mathbf{E}_i\right).\end{aligned} \tag{6.21}$$

The Jacobian matrix $\mathrm{J}_{\mathbf{q}_l^{(t)}}$ of $\mathbf{q}_l^{(t)}$ is computed from $\mathrm{J}_{\mathbf{m}_l^{(t)}}$ (Equation 6.20), the coefficients of $\mathbf{P}_{\langle 2\rangle}^{(t)}$ (denoted by $\{\phi_t^k\}_{k=1}^{2^n}$), and the tensor Λ:

$$\mathrm{J}_{\mathbf{q}_l^{(t)}}^{i,z} = \frac{\partial \beta_{t,l}^i}{\partial \chi^z} = \sum_{j,k=1}^{2^n} \left(\mathrm{J}_{\mathbf{m}_l^{(t)}}^{j,z} \, \phi_t^k \, \Lambda^{i,j,k}\right). \tag{6.22}$$

Recall from Section 5.1 that, when $\mathcal{Q}$ is not empty, one of the tuples in $\mathcal{Q}$ is used to compute the parameter θ^t (Figure 13 line 13). The same is true for in the procedure in Figure 6.3. However, in order to simplify the definition of the Jacobian of θ^t, the equation

$$\theta^t = \tan^{-1}\left(\frac{\left(\mathbf{q}_l^{(t)} \wedge \mathbf{r}_l^{(t)}\right) * \mathbf{P}_{\langle 2\rangle}^{(t)}}{\mathbf{q}_l^{(t)} * \mathbf{r}_l^{(t)}}\right)$$

is rewritten as:

$$\theta^t = \tan^{-1}\left(\frac{\tau_{t,l}}{\nu_{t,l}}\right), \tag{6.23}$$

where

$$\begin{aligned}\tau_{t,l} = \left(\mathbf{q}_l^{(t)} \wedge \mathbf{r}_l^{(t)}\right) * \mathbf{P}_{\langle 2\rangle}^{(t)} &= \mathbf{q}_l^{(t)} * \left(\mathbf{r}_l^{(t)} \rfloor \mathbf{P}_{\langle 2\rangle}^{(t)}\right) \\ &= \sum_{j,k=1}^{2^n} \left(\beta_{t,l}^j \, \Omega^{j,k}\right),\end{aligned} \tag{6.24}$$

and

$$\begin{aligned}\nu_{t,l} &= \mathbf{q}_l^{(t)} * \mathbf{r}_l^{(t)} \\ &= \sum_{j,k=1}^{2^n} \left(\beta_{t,l}^j \, \psi_{t,l}^k \, \Lambda^{1,j,k}\right),\end{aligned} \tag{6.25}$$

with Ω (in Equation 6.24) being a constant defined as:

$$\Omega^{j,k} = \Lambda^{1,j,k} \sum_{h,l=1}^{2^n} \left(\psi_{t,l}^h \, \phi_t^l \, \Lambda^{k,h,l}\right). \tag{6.26}$$

In Equations 6.24, 6.25 and 6.26, $\{\beta_{t,l}^j\}_{j=1}^{2^n}$, $\{\psi_{t,l}^k\}_{k=1}^{2^n}$ and $\{\phi_t^l\}_{l=1}^{2^n}$ denote the coefficients of, respectively, $\mathbf{q}_l^{(t)}$ (Equation 6.21), $\mathbf{r}_l^{(t)}$ (defined in Equation 5.7) and $\mathbf{P}_{\langle 2\rangle}^{(t)}$.

The derivatives of $\tau_{t,l}$ (Equation 6.24) and $\nu_{t,l}$ (Equation 6.25) are, respectively:

$$\mathrm{J}_{\tau_{t,l}}^{1,z} = \frac{\partial \tau_{t,l}}{\partial \chi^z} = \sum_{j,k=1}^{2^n} \left(\mathrm{J}_{\mathbf{q}_l^{(t)}}^{j,z} \, \Omega^{j,k}\right) \tag{6.27}$$

and

$$\mathrm{J}_{\nu_{t,l}}^{1,z} = \frac{\partial \nu_{t,l}}{\partial \chi^z} = \sum_{j,k=1}^{2^n} \left(\mathrm{J}_{\mathbf{q}_l^{(t)}}^{j,z} \, \psi_{t,l}^k \, \Lambda^{1,j,k}\right). \tag{6.28}$$

Once one has $\tau_{t,l}$ (Equation 6.24), $\nu_{t,l}$ (Equation 6.25), $\mathrm{J}_{\tau_{t,l}}$ (Equation 6.27), and $\mathrm{J}_{\nu_{t,l}}$ (Equation 6.28), one can compute the Jacobian of θ^t (Equation 6.23) as:

$$\mathrm{J}_{\theta^t}^{1,z} = \frac{\partial \theta^t}{\partial \chi^z} = \frac{1}{(\tau_{t,l})^2 + (\nu_{t,l})^2} \left(\mathrm{J}_{\tau_{t,l}}^{1,z} \, \nu_{t,l} - \tau_{t,l} \, \mathrm{J}_{\nu_{t,l}}^{1,z}\right) \tag{6.29}$$

At each iteration of the loop in Figure 6.3, the blade $\mathbf{Y}^{(t)}$ is updated (see line 15) by removing from it the blade $\mathbf{M}_l^{(t)} \in \mathcal{N}$ having the highest dimensionality. The new $\mathbf{Y}^{(t)}$ is given by:

$$\begin{aligned}\mathbf{Y}^{(t)} = (\mathbf{M}_l^{(t)})^{-1} \rfloor \mathbf{Y}_{\text{old}}^{(t)} &= \mathbf{N}_l^{(t)} \rfloor \mathbf{Y}_{\text{old}}^{(t)} \\ &= \sum_{i,j,k=1}^{2^n} \left(\xi_{t,l}^j \, \eta_{t,old}^k \, \Lambda^{i,j,k} \, \mathbf{E}_i\right),\end{aligned} \tag{6.30}$$

where

$$\begin{aligned}\mathbf{N}_l^{(t)} = (\mathbf{M}_l^{(t)})^{-1} &= \frac{\widetilde{\mathbf{M}}_l^{(t)}}{\mathbf{M}_l^{(t)} * \widetilde{\mathbf{M}}_l^{(t)}} \\ &= \frac{1}{\sum_{h,k=1}^{2^n} \left(\mu_{t,l}^h \, \mu_{t,l}^k \, \Upsilon^{k,k} \, \Lambda^{1,h,k}\right)} \sum_{j=1}^{2^n} \left(\mu_{t,l}^j \, \Upsilon^{j,j} \, \mathbf{E}_j\right)\end{aligned} \tag{6.31}$$

is an intermediate multivector variable with coefficients $\{\xi_{t,l}^{j}\}_{j=1}^{2^n}$. The derivative of $\{\xi_{t,l}^{j}\}_{j=1}^{2^n}$ is:

$$\mathrm{J}_{\mathbf{N}_l^{(t)}}^{j,z} = \frac{\partial \xi_{t,l}^{j}}{\partial \chi^z} = \sum_{h,k=1}^{2^n} \left(\left(\mathrm{J}_{\mathbf{M}_l^{(t)}}^{j,z} \, \mu_{t,l}^{h} \, \mu_{t,l}^{k} - \mu_{t,l}^{j} \, \mathrm{J}_{\mathbf{M}_l^{(t)}}^{h,z} \, \mu_{t,l}^{k} \right. \right. \\ \left. \left. - \mu_{t,l}^{j} \, \mu_{t,l}^{h} \, \mathrm{J}_{\mathbf{M}_l^{(t)}}^{k,z} \right) \Upsilon^{j,j} \, \Upsilon^{k,k} \, \Lambda^{1,h,k} \right), \quad (6.32)$$

and the Jacobian of the new $\mathbf{Y}^{(t)}$ (Equation 6.30) is:

$$\mathrm{J}_{\mathbf{Y}^{(t)}}^{i,z} = \frac{\partial \eta_t^i}{\partial \chi^z} = \sum_{j,k=1}^{2^n} \left(\left(\mathrm{J}_{\mathbf{N}_l^{(t)}}^{j,z} \, \eta_{t,old}^{k} + \xi_{t,l}^{j} \, \mathrm{J}_{\mathbf{Y}_{\text{old}}^{(t)}}^{k,z} \right) \Lambda^{i,j,k} \right). \quad (6.33)$$

The coefficients of $\mathbf{Y}_{\text{old}}^{(t)}$ and $\mathbf{M}_l^{(t)}$ are denoted in Equations 6.30, 6.31, 6.32 and 6.33 by $\{\eta_{t,old}^{k}\}_{k=1}^{2^n}$ and $\{\mu_{t,l}^{i}\}_{i=1}^{2^n}$, respectively. The reverse operation is encoded by Υ, and the left contraction and the scalar product are encoded by Λ.

It is important to notice that tensors encoding GA operations are sparse structures. Also, multivectors encoding k-blades have at most $\binom{n}{k}$ nonzero coefficients (*i.e.,* the ones related to $\bigwedge^k \mathbb{R}^n$), and rotors use only the coefficients related to even-dimensional basis blades of $\bigwedge \mathbb{R}^n$. It follows that the summations in all preceding derivations only need to be evaluated for possible nonzero multivector's coefficients and tensor's entries. Such a feature may be used to reduce the computational load of computing the Jacobian matrices.

6.3.3 Mapping Procedure for $r \leq p$

As discussed in Section 5.2, when the dimensionality of an input blade is less or equal than the dimensionality of the intended type of subspace one can take the dual of input ($\mathbf{X}_{\langle r \rangle}^{*}$) and reference ($\mathbf{E}_{\langle p \rangle}^{*}$) blades in order to reduce the mapping procedure to the case described in Section 5.1 (*i.e.,* where $r \geq p$). Thus, for input subspaces with uncertainty, the dual of a random multivector variable $\underline{\mathbf{X}_{\langle r \rangle}}$ must be considered.

In this case, the set $\mathcal{P}^{(m)}$ in Figure 6.2 (line 1) is initialized with a 5-tuple (see Equation 6.7) having:

$$\left(\overline{\mathbf{X}_{\langle r \rangle}}\right)^{*} = \mathbf{X}_{\langle r \rangle} \rfloor \mathbf{I}_{\langle n \rangle}^{-1} = \mathbf{X}_{\langle r \rangle} \rfloor \widetilde{\mathbf{I}_{\langle n \rangle}} \\ = \sum_{i,j,k=1}^{2^n} \left(\chi^j \, \iota^k \, \Upsilon^{k,k} \, \Lambda^{i,j,k} \, \mathbf{E}_i \right) \quad (6.34)$$

as its first entry. The second entry of the 5-tuple is the Jacobian matrix of $\left(\overline{\mathbf{X}_{\langle r \rangle}}\right)^{*}$, whose derivatives are computed as:

$$\frac{\partial \lambda_m^i}{\partial \chi^z} = \sum_{k=1}^{2^n} \left(\iota^k \, \Upsilon^{k,k} \, \Lambda^{i,i,k} \right). \quad (6.35)$$

The other three entries of the 5-tuple are, respectively, $\boldsymbol{K}^{(m)} = 1$, $\mathrm{J}_{\boldsymbol{K}^{(m)}} = 0$ and $\varnothing$. In Equations 6.34 and 6.35, $\{\lambda^i\}_{i=1}^{2^n}$ denotes the coefficients of $\left(\overline{\mathbf{X}_{\langle r \rangle}}\right)^{*}$, and $\{\iota^k\}_{k=1}^{2^n}$ are the coefficients for the (constant) pseudoscalar $\mathbf{I}_{\langle n \rangle}$. It is important to notice that, when $r \leq p$, the derivatives in the Jacobian matrices computed by the mapping procedure are related to the coefficients $\{\chi^z\}_{z=1}^{2^n}$ of the direct representation of the mean input blade $\overline{\mathbf{X}_{\langle r \rangle}}$.

6.4 Voting Procedure

The subspace detection framework presented in this thesis identifies the most likely p-blades in a given dataset by performing a voting procedure using an accumulator array as the discrete representation of $\mathbb{P}^m$ (Equation 4.10). The mapping procedure described in Section 6.3 is key for such a voting. It takes an uncertain r-blade ($\underline{\mathbf{X}_{\langle r \rangle}}$) and decompose it as parameter vectors $\Theta^{(0)} \in \mathbb{P}^m$ and vector-valued random variables $\underline{\mathrm{a}}$. The resulting parameter vectors are computed from the expectation of $\mathbf{X}_{\langle r \rangle}$. Thus, they characterize the most probable p-blades related to the input entry. The p-blades related to the uncertainty around the expectation of $\mathbf{X}_{\langle r \rangle}$ are represented in an auxiliary space $\mathbb{A}^m$ by $\underline{\mathrm{a}}$.

For a given pair $(\Theta^{(0)}, \underline{\mathrm{a}})$ (Figure 6.2, line 10), the number of votes to be incremented to each accumulator's bin can be computed by: (i) mapping the bin's region from $\mathbb{P}^m$ to $\mathbb{A}^m$; and, in turn, (ii) weighting the importance value ω of the input $\underline{\mathbf{X}_{\langle r \rangle}}$ by the probability of a related p-blade be in the mapped region. Ideally, such a probability should be computed in $\mathbb{A}^m$ as the hypervolume under the portion of the (normalized) multivariate "bell" curve contained by the mapped region. However, as depicted in Figure 6.4, rectangular regions in the actual parameter space (Figure 6.4a) map to warped regions in the auxiliary parameter space (Figure 6.4b). It is a challenging and computationally intensive task to evaluate the probability in such warped regions (COWAN, 1998). **The proposed solution for this issue is to define, for each bin's region at $\mathbb{A}^m$, a representative box aligned to the eigenvectors of the covariance matrix Σ_{a} of $\underline{\mathrm{a}}$.** As depicted in Figure 6.4c, in the space defined by the orthogonal eigenvectors of Σ_{a}, the eigenvalues represent the variances of an axis-aligned Gaussian distribution (in Figure 6.4c, the unit eigenvectors are scaled by the eigenvalues), and the covariances are equal to zero. Hence, **the resulting probability can be efficiently computed as the product of the probabilities of the intervals defined by the representative box**.

The representative box of a bin having coordinates Θ_{bin} is built from points $\{\Theta^i_{\text{face}}\}_{i=1}^{2\,m}$ placed at the center of bin's faces in the parameter space (Figure 6.4a). By using Δ_{θ^t} radians as step in the linear discretization of the t-th dimension of the parameter space, the center of the faces are computed as:

$$\Theta^i_{\text{face}} = \Theta_{\text{bin}} + \Theta_{\text{offset}_i} \quad \forall\, i \in \{1, 2, \cdots, 2\,m\},$$

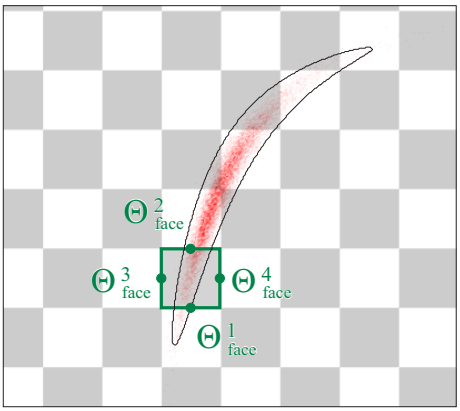
where

$$\Delta_{\text{offset}_i} = \begin{cases} (0, \cdots, -\Delta_{\theta^{\lfloor \frac{i+1}{2} \rfloor}}, \cdots, 0) & \text{for odd } i \\ (0, \cdots, +\Delta_{\theta^{\lfloor \frac{i+1}{2} \rfloor}}, \cdots, 0) & \text{for even } i \end{cases}$$

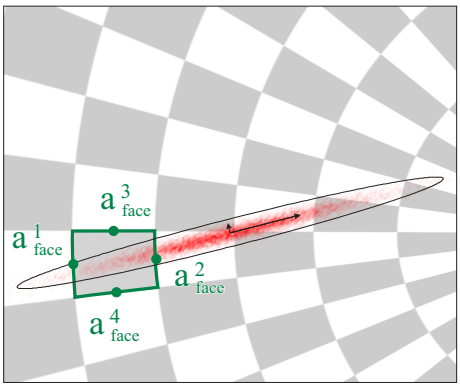
is the translation vector from the center of a bin to the center of a bin's face. $\lfloor \square \rfloor$ denotes the floor function.

Each point Θ^i_{face} is mapped from $\mathbb{P}^m$ to $\mathbb{A}^m$ (Figure 6.4b) by using the following procedure:

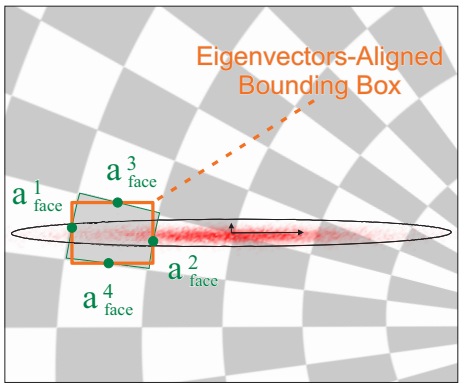
1. The vectors spanning the reference blade $\mathbf{E}_{\langle p \rangle}$ (Equation 5.4) are transformed by the rotor $\boldsymbol{W}\,\boldsymbol{T}^i_{\text{face}}$, where $\boldsymbol{T}^i_{\text{face}}$ is computed according to Equation 4.8 using the coordinates (rotation angles) of Θ^i_{face}. The rotor $\boldsymbol{W}$ is given by Equation 6.15 for current pair $(\Theta^{(0)}, \underline{\mathrm{a}})$;
2. The vectors resulting from step 1 are used to defined a $p \times n$ matrix representation the subspace related to Θ^i_{face};
3. The location of Θ^i_{face} in $\mathbb{A}^m$ (denoted by points $\mathrm{a}^i_{\text{face}}$ in Figure 6.4b) is retrieved from the row reduced echelon form of the matrix build in step 2.



(a) Parameter space $\mathbb{P}^{2\,(3-2)}$



(b) Open affine covering $\mathbb{A}^2$ for $G(2,3)$



(c) Orthonormal basis of eigenvectors in $\mathbb{A}^2$

Figure 6.4: Computing the number of votes to be incremented to a given bin of the accumulator array: The points Θ^i_{face} at the center of the bin's face (a) are mapped to the open affine covering for the Grassmannian as points $\mathrm{a}^i_{\text{face}}$ (b). In turn, the points $\mathrm{a}^i_{\text{face}}$ are mapped to the basis defined by the orthonormal eigenvectors of the distribution of probability (c). An axis-aligned bounding box is computed for $\mathrm{a}^i_{\text{face}}$ in such a basis. The number of votes is proportional to the weight of the input blade and the probabilities of an intended p-blade be inside of the box.

Once the points $\{\mathrm{a}^i_{\text{face}}\}_{i=1}^{2^m}$ are known, each $\mathrm{a}^i_{\text{face}}$ is transformed to the basis defined by the (orthonormal) eigenvectors of Σ_{a} (Figure 6.4c). The transformation is achieved by pre-multiplicating $\mathrm{a}^i_{\text{face}}$ by the transpose of a matrix having such eigenvectors as its columns. The eigenvectors-aligned bounding box including $\{\mathrm{a}^i_{\text{face}}\}_{i=1}^{2^m}$ is the representative box of the bin's region in $\mathbb{A}^m$. Each dimension of such a box defines an interval $[\min_t, \max_t]$ in one of the axis related to an eigenvector (see Figure 6.4c). The number of votes to be incremented in the bin is computed as:

$$votes = \omega \prod_{t=1}^{m} \left(\Phi\left(\frac{\max_t}{\sigma_t}\right) - \Phi\left(\frac{\min_t}{\sigma_t}\right) \right), \tag{6.36}$$

where σ_t is the square root of the eigenvalue related to the t-th eigenvector of Σ_{a}, ω is the importance of the input entry $\mathbf{X}_{\langle r \rangle}$, and Φ is the cumulative distribution function of the standard normal distribution (COWAN, 1998).

As depicted in Figure 6.4, one needs to compute votes only for the bins intersecting the footprint of the Gaussian distribution in $\mathbb{A}^m$. It is because the number of votes for the bins beyond three standard deviations (the ellipses in Figures 6.4b and 6.4c) are negligible.

Intuitively, Equation 6.36 can be evaluated starting at the bin containing $\Theta^{(0)}$ (*i.e.,* the parameter vector related to the expectation $\overline{\mathrm{a}}$) and moving outwards (in parameter space, Figure 6.4a) in a flood fill fashion, until the returned values are lower than a threshold ϵ_{votes}. For the examples presented in this chapter ϵ_{votes} had been empirically set to 10^{-6}.

It is important to notice that the vector-valued random variable $\underline{\mathrm{a}}$ defines a k-dimensional multivariate Gaussian distribution in $\mathbb{A}^m$, for $0 \leq k \leq m$. It means that its distribution of uncertainty may have arbitrary dimensionality in the open affine covering, and hence also in $\mathbb{P}^m$. The dimensionality of the Gaussian distribution (k) will be equal to m if and only if the following conditions are satisfied:

1. A given input entry $\underline{\mathbf{X}_{\langle r \rangle}}$ maps to only one pair $(\Theta^{(0)}, \underline{\mathrm{a}})$ (*i.e.,* all parameter values were computed and returned by Figure 6.3, line 13); and
2. The uncertainty in $\underline{\mathbf{X}_{\langle r \rangle}}$ does not restrict the distribution of $\underline{\mathrm{a}}$ to a linear subspace in $\mathbb{A}^m$ having a dimensionality smaller than m.

When condition (1) fails then the distribution of $\underline{\mathrm{a}}$ will have at most $k = (m - s)$ dimensions, where s is the number of arbitrated parameters (Figure 6.3, line 8). In such a case, the distributions of votes centered at the coordinates of each parameter vector $\Theta^{(0)} \in \mathbb{P}m$ define (warped) parallel profiles in $\mathbb{P}^m$.

The distribution of $\underline{\mathrm{a}}$ also loose dimensions when condition (2) fails. It happens when there is no uncertainty in some of the degrees of freedom of the input entry $\underline{\mathbf{X}_{\langle r \rangle}}$. In an extreme case, the input blade does not have uncertainty at all. In this situation, the extended voting scheme presented in this chapter reduces to the approach presented in Chapter 5, spreading votes in the accumulator array using the simpler voting procedure described in Section 5.4 (*i.e.,* the one without interpolation).

From our experience, the interpolation scheme described in Section 5.4 is not necessary for uncertain data. This is because the flood-fill voting is capable to fill the gaps in the structures rasterized in the accumulator array.

6.5 Results and Discussion

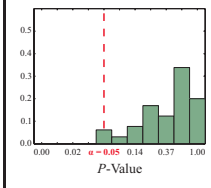
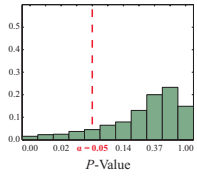
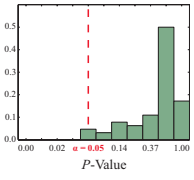
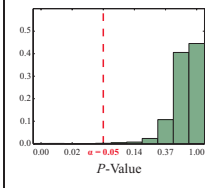
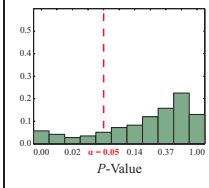
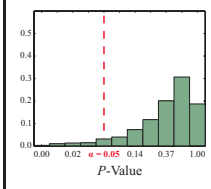
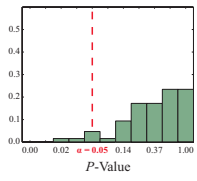
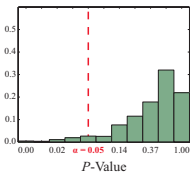
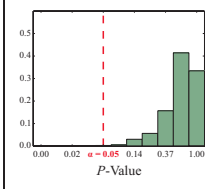
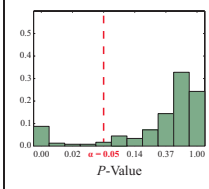
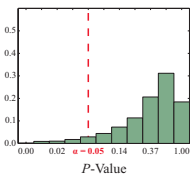
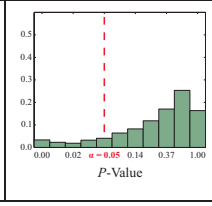
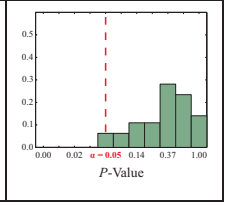
A mapping and a voting procedure for input subspaces with Gaussian distributed uncertainty were presented in this chapter. The mapping scheme uses first-order error propagation to transport the uncertainty from input data to an auxiliary space defined as the open affine covering $\mathbb{A}^m$ for the Grassmannian $G(p,n)$. The propagated uncertainty defines a Gaussian profile in $\mathbb{A}^m$, which is mapped by the voting procedure to the actual parameter space $\mathbb{P}^m$ as a non-Gaussian distribution of votes.

The prerequisite for using first-order error propagation is to check whether the resulting uncertainty can be well approximated by a Gaussian distribution (COWAN, 1998; PERWASS, 2009). This condition was asserted for the proposed mapping procedure by carrying out two sets of sampling-based statistical experiments. In the first of the two sets, each experiment assumes a detection case (*i.e.,* r-dimensional subspaces in a n-dimensional representational space), and one input uncertain r-blade having a given covariance matrix. The covariance matrices were defined by rotating the principal axes of a reference axis-aligned Gaussian distribution according to its degrees of freedom (*i.e.,* the $k\,(k-1)/2$ generalized Euler angles – Appendix C – characterizing the orientation of a k-dimensional Gaussian distribution). In this way, the experiments cover a wide range of settings for Gaussian distributions. For each experiment, a set of $1{,}000$ random samples of the input uncertain blade was generated according to the assumed Gaussian distribution.

The uncertain blade was then mapped to $\mathbb{A}^m$ with the procedure presented in Section 6.3.1 (or Section 6.3.3). In turn, the samples related to the uncertain blade where mapped to the same auxiliary space $\mathbb{A}^m$. Recall from Section 6.3.1 that $\mathbb{A}^m$ is defined in such a way that the expectation of the uncertain blade is at its origin. Thus, the samples are mapped to points around the origin of $\mathbb{A}^m$. This distribution of points must be Gaussian (Normal) in order to validate the assumption that first-order error propagation can be used to predict a distribution of uncertainty in $\mathbb{A}^m$. The normality of the distribution of points was verified with the statistical hypothesis test proposed by Mardia (1970). Mardia's procedure (MARDIA, 1970) computes a P-value for the skewness, and another one for the kurtosis of a given set of points. If these P-values are greater or equal to the significance level $\alpha = 0.05$, then the results are statistically not-significant (*i.e.,* the distribution of points is probably Gaussian because there is only a 5% chance of the expected measures of skewness and kurtosis have happened by coincidence). Otherwise, for P-values lower that α, there is a reasonable chance of the distribution being non-Gaussian.

Tables 6.1 and 6.2 summarize the P-values computed, respectively, for skewness and kurtosis in all the experiments of the first of two sets. These experiments ($1{,}530$ altogether) are grouped by table entries according to a detection case (see the n and p values at column headers), and a dimensionality of input uncertain blades (see the r value for each row). Each table entry shows a relative frequency histogram of P-values (the abscissa, in logarithmic scaling) for the respective detection/input case. Notice that almost all computed P-values had fallen in the bins at right side of (and including) the signif-

Table 6.1: Relative frequency histograms of P-values computed for the skewness of distributions of points in the auxiliary space $\mathbb{A}^m$. The histograms at table entries are related to experiments carried out regarding the combination of a given detection case (columns) with input uncertain blades having a given dimensionality r (rows). Notice that the most frequent P-values are greater or equal to the significance level $\alpha = 0.05$ (the dashed lines). These results show that distributions of points related to almost all the experiments have the skewness of a Gaussian distribution.

	$n = 3, p = 1$ $(m = 2)$	$n = 3, p = 2$ $(m = 2)$	$n = 4, p = 1$ $(m = 3)$	$n = 4, p = 2$ $(m = 4)$	$n = 4, p = 3$ $(m = 3)$
$r = 1$					
$r = 2$					
$r = 3$	–	–			

icance level $\alpha = 0.05$ (denoted by the dashed lines). These results suggest that samples mapped to $\mathbb{A}^m$ define Gaussian distributions for different values of variance and covariance in the input uncertain blades. However, some of the P-values presented in Tables 6.1 and 6.2 are smaller than α. Thus, it is necessary to verify whether these P-values are related with some sensibility of Mardia's procedure, or with some bounding limit for input uncertainty.

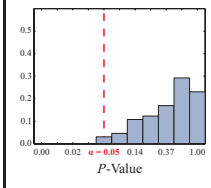
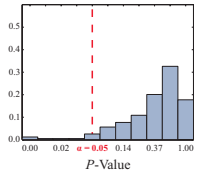
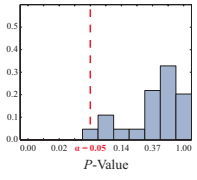
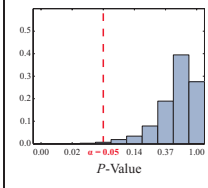
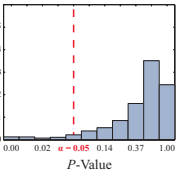
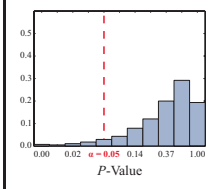
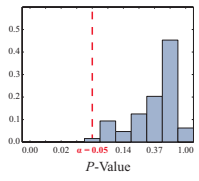
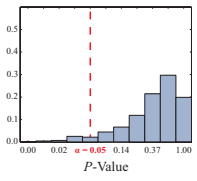
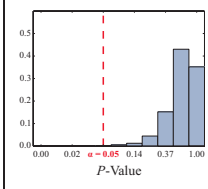
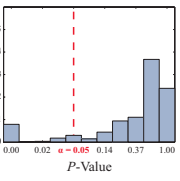
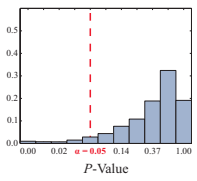
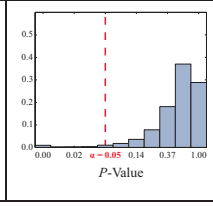
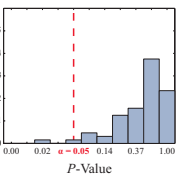
The results in Tables 6.1 and 6.2 motivated the development of another set of sampling-based statistical experiments. With this second set of experiments we want to observe how the "quality" of skewness and kurtosis change as the uncertainty of input blades changes. For these experiments, it is assumed the detection of straight lines in the 2-dimensional homogeneous MOG. Thus, $p = 2$ and $n = 2 + 1$, leading to $m = 2\,(3 - 2) = 2$.

Initially, a set of $1{,}225$ blades was created by choosing 35×35 parameter vectors linearly distributed over the parameter space $\mathbb{P}^2$ for 2-blades in $\bigwedge \mathbb{R}^{2+1}$. Through Equation 4.9, each one of the parameter vectors is related to a blade, which is regarded here as the expectation ($\overline{\mathbf{X}_{\langle 2 \rangle}}$) of a random multivector variable $\underline{\mathbf{X}_{\langle 2 \rangle}}$. By converting $\overline{\mathbf{X}_{\langle 2 \rangle}}$ to the parameterization defined by the normal equation of the line:

$$x \cos\phi + y \sin\phi - \rho = 0, \tag{6.37}$$

and by assigning a standard deviation to ρ (denoted by σ_ρ) and another one to ϕ (denoted by σ_ϕ), one can define the covariance matrix of $\underline{\mathbf{X}_{\langle 2 \rangle}}$ from σ_ρ and σ_ϕ. This way, it is possible to verify changes on the skewness and kurtosis of distributions of samples mapped to $\mathbb{A}^2$ as an effect of changing the uncertainty in parameters having a clear geometrical interpretation: ρ defines the distance from the origin point of the base space to the line, and

Table 6.2: Relative frequency histograms of P-values computed for the kurtosis of distributions of points in the auxiliary space $\mathbb{A}^m$. This table complements the results presented in Table 6.1. It shows that distributions of points related to almost all the experiments have the kurtosis of a Gaussian distribution.

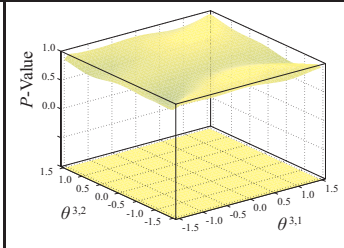
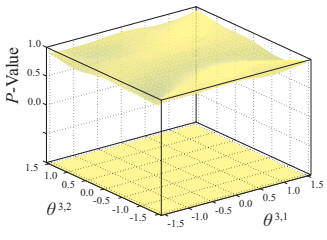
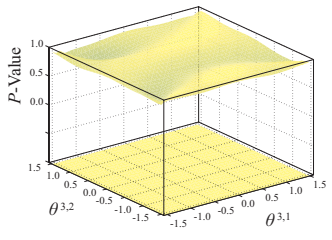
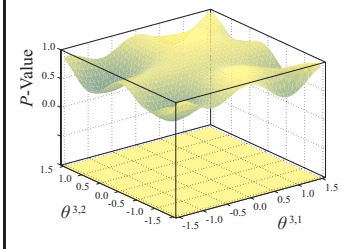
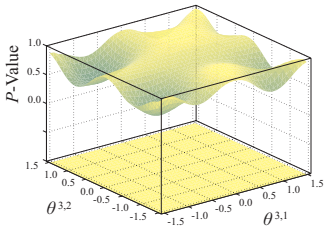
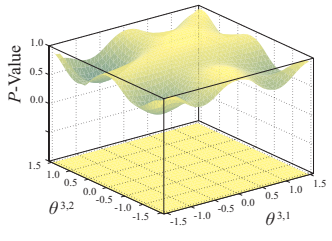
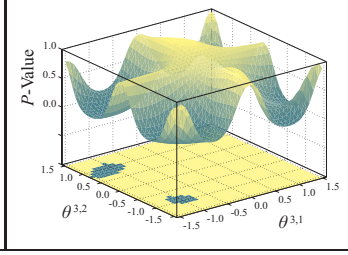
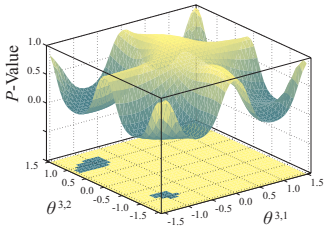
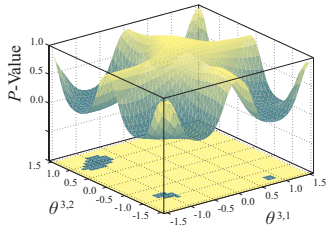
	$n = 3, p = 1$ $(m = 2)$	$n = 3, p = 2$ $(m = 2)$	$n = 4, p = 1$ $(m = 3)$	$n = 4, p = 2$ $(m = 4)$	$n = 4, p = 3$ $(m = 3)$
$r = 1$					
$r = 2$					
$r = 3$	–	–			

ϕ is the angle between the x-axis and the normal to the line. The mean normal parameters of the line are computed from $\overline{\mathbf{X}_{\langle 2\rangle}}$ as:

$$\overline{\rho} = \|\mathbf{s}\| \quad \text{and} \quad \overline{\phi} = \tan^{-1}\left(\frac{\mathbf{n} * \mathbf{e}_2}{\mathbf{n} * \mathbf{e}_1}\right), \tag{6.38}$$

where $\mathbf{s} = \left(\mathbf{e}_0^{-1} \rfloor \left(\mathbf{e}_0 \wedge \overline{\mathbf{X}_{\langle 2\rangle}}\right)\right) / \mathbf{d}$ is the support vector of the line, $\mathbf{d} = \mathbf{e}_0^{-1} \rfloor \overline{\mathbf{X}_{\langle 2\rangle}}$ is its direction, and $\mathbf{n} = \mathbf{e}_0^{-1} \rfloor \left(\mathbf{e}_0 \wedge \left(\overline{\mathbf{X}_{\langle 2\rangle}}\right)^*\right)$ is the normal to the line, with the condition that $\mathbf{s} * \mathbf{n} \geq 0$. Recall from Section 3.9.2 that the basis vector $\mathbf{e}_0$ is interpreted in the homogeneous MOG as the point at the origin. Vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ define the coordinate system of the base space. They are related to, respectively, x and y in Equation 6.37. It is important to comment that, in an implementation, one should evaluate the arctangent in Equation 6.38 with the function ATAN2 in order to provide numerical stability and additionally make $\overline{\phi} \in [-\pi, \pi)$. In such a case, one should assume $\overline{\rho} \in [0, \infty)$. The ATAN2 function is available in many programming languages. The expectation of $\underline{\mathbf{X}_{\langle 2\rangle}}$ can be computed from $\overline{\rho}$

Table 6.3: P-values computed for the skewness of distributions of points in the auxiliary spaces $\mathbb{A}^2$ related to uncertain 2-blades. These blades were chosen as the parameter vectors defining a 35×35 grid in parameters space $\mathbb{P}^2$. The 3-dimensional visualization of $\mathbb{P}^2$ at table entries are related to experiments carried out regarding the input uncertain blades and given values for standard deviations σ_ρ (rows) and σ_ϕ (columns). The heights of the surfaces represent the P-values, while the dark spots on the plane at the bottom of the charts denote those P-values which are lower than the significance level $\alpha = 0.05$. Notice that the skewness of some distributions differ from the one expected for a Gaussian distribution only at $\sigma_\rho = 0.08$.

	$\sigma_\phi = 0.09$	$\sigma_\phi = 0.13$	$\sigma_\phi = 0.17$
$\sigma_\rho = 0.02$			
$\sigma_\rho = 0.04$			
$\sigma_\rho = 0.08$			

and $\overline{\phi}$ (Equation 6.38) using:

$$\overline{\mathbf{X}_{\langle 2\rangle}} = \left(\overline{\rho}\,\left(\mathbf{e}_0^{-1}\right) - \cos\left(\overline{\phi}\right)\mathbf{e}_1 - \sin\left(\overline{\phi}\right)\mathbf{e}_2\right)^{-*}. \tag{6.39}$$

It follows that the covariance matrix of $\underline{\mathbf{X}_{\langle 2\rangle}}$ is defined as:

$$\Sigma_{\mathbf{X}_{\langle r\rangle}} = \mathrm{J}_{\mathbf{X}_{\langle r\rangle}} \begin{pmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix} \mathrm{J}^{\mathrm{T}}_{\mathbf{X}_{\langle r\rangle}}, \tag{6.40}$$

where the derivatives in the Jacobian matrix $\mathrm{J}_{\mathbf{X}_{\langle r\rangle}}$ are computed as:

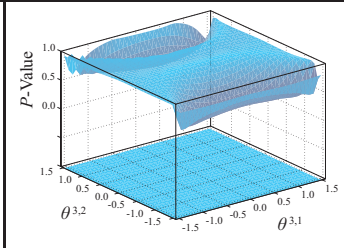
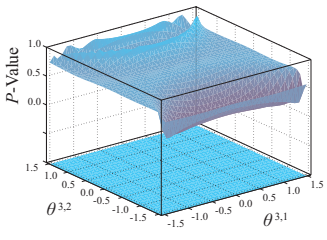
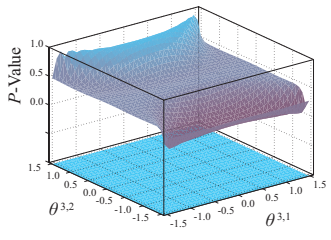
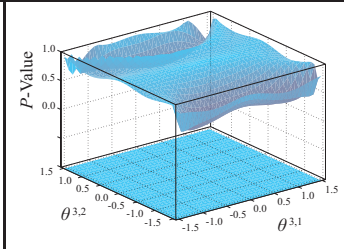
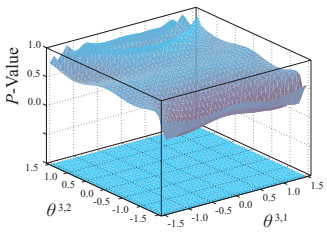
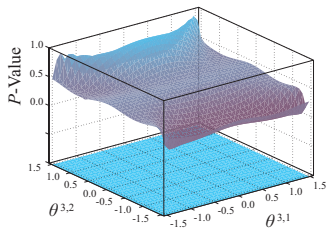
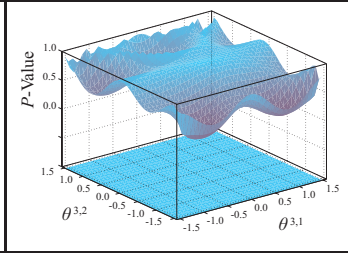
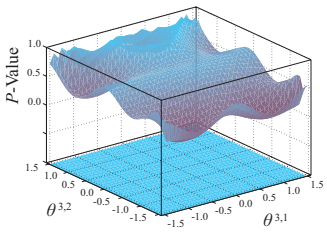
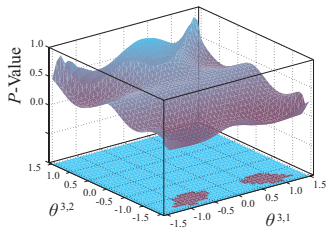
$$\mathrm{J}^{i,1}_{\mathbf{X}_{\langle r\rangle}} = \frac{\partial \chi^i}{\partial \rho} = \sum_{k=1}^{2^n} \left(\delta_0^k\, \Lambda^{i,1,k}\right), \tag{6.41}$$

and

$$\mathrm{J}^{i,2}_{\mathbf{X}_{\langle r\rangle}} = \frac{\partial \chi^i}{\partial \phi} = \sin\left(\overline{\phi}\right) \sum_{k=1}^{2^n} \left(\delta_1^k\, \Lambda^{i,1,k}\right) - \cos\left(\overline{\phi}\right) \sum_{k=1}^{2^n} \left(\delta_2^k\, \Lambda^{i,1,k}\right). \tag{6.42}$$

In Equations 6.41 and 6.42, the coefficients of the constant subspaces $\left(\mathbf{e}_0^{-1}\right)^{-*}$, $\mathbf{e}_1^{-*}$, and $\mathbf{e}_2^{-*}$ are denoted by $\{\delta_0^k\}_{k=1}^{2^n}$, $\{\delta_1^k\}_{k=1}^{2^n}$, and $\{\delta_2^k\}_{k=1}^{2^n}$, respectively.

Table 6.4: P-values computed for the kurtosis of distributions of points in the auxiliary spaces $\mathbb{A}^2$ related to uncertain 2-blades. This table complements the results presented in Table 6.3. Notice that the kurtosis is less sensitive to variations on the input uncertain than the skewness. The first distributions having kurtosis different from the one expected for a Gaussian distribution had appeared at $\sigma_\rho = 0.08$ and $\sigma_\phi = 0.17$.

	$\sigma_\phi = 0.09$	$\sigma_\phi = 0.13$	$\sigma_\phi = 0.17$
$\sigma_\rho = 0.02$			
$\sigma_\rho = 0.04$			
$\sigma_\rho = 0.08$			

After the $1{,}225$ mean blades $\overline{\mathbf{X}_{\langle 2\rangle}}$ had been written as a function of their mean normal parameters $\overline{\rho}$ and $\overline{\phi}$ (Equation 6.39), a value for σ_ρ and another one for σ_ϕ was chosen, and the covariance matrices $\Sigma_{\mathbf{X}_{\langle 2\rangle}}$ were computed using Equation 6.40. Then, two canonical sets with 500 random real values each were generated following a standard Normal distribution. A copy of these canonical samples was assigned to each input uncertain blade $\mathbf{X}_{\langle 2\rangle}$, and converted to the Gaussian distribution of its respective $\underline{\rho}$ and $\underline{\phi}$ variables. The use of canonical samples is important because they make possible the comparison of skewness and kurtosis of distributions related to different input uncertain subspaces $\underline{\mathbf{X}_{\langle 2\rangle}}$. Finally, as well as in the first set of experiments presented in this section, each uncertain blade was mapped to $\mathbb{A}^2$. In turn, its respective samples where mapped to the same auxiliary space, and the normality of the distribution of points in $\mathbb{A}^2$ was verified with Mardia's procedure (MARDIA, 1970). Tables 6.3 and 6.4 present the P-values produced for, respectively, the skewness and kurtosis of the distributions of points in $\mathbb{A}^n$. The table entries group the experiments according to the values assumed for σ_ρ (rows) and σ_ϕ (columns). The P-values are shown as heights of a surface on a 3-dimensional visualization of the parameter space $\mathbb{P}^2$ (*i.e.,* the height at the parameter vector related to a $\overline{\mathbf{X}_{\langle 2\rangle}}$ is the P-value estimated from the samples of the respective $\underline{\rho}$ and $\underline{\phi}$ variables). The P-values lower than $\alpha = 0.05$ are distinguished as darker colors on the plane at the bottom of the charts. Notice that the samples mapped to $\mathbb{A}^2$ define Gaussian distributions even for larger values of σ_ρ and σ_ϕ (*i.e.,* there are a few small dark spots in Tables 6.3 and 6.4). The first asymmetric distributions had appeared only for $\sigma_\rho = 0.08$ (see the dark spots at third row of Table 6.3), while the tails of a few distributions differ from a Gaussian distribution only for $\sigma_\rho = 0.08$ and $\sigma_\phi = 0.17$ (see the two dark spots at the last entry of Table 6.4). By assuming an image with coordinates in $[-1, +1] \times [-1, +1]$ range and the homogeneous MOG, it follows that the higher uncertainty values ($\sigma_\rho = 0.08$ and $\sigma_\phi = 0.17$) define a confidence interval of ± 0.24 units for the distance from the center of the image to a given line (*i.e.,* almost $1/4$ of image's size), and a confidence interval of ± 0.51 radians for the direction of the line (*i.e.,* almost $\pi/3$ radians). These results show that, as far the uncertainty are kept low, input random multivector variables define Gaussian distributions in $\mathbb{A}^2$. Therefore, P-values smaller than $\alpha = 0.05$ observed in Tables 6.1 and 6.2 may be related to high uncertainty in the input blades.

From Tables 6.1, 6.2, 6.3, and 6.4 it is possible to conclude that error propagation can be used to predict the distribution of uncertainty of blades mapped to $\mathbb{A}^m$. But it is also important to verify if first-order analysis is sufficient to approximate the expected Gaussian distributions. Such an analysis was performed by comparing the covariance matrix computed for random samples mapped to $\mathbb{A}^m$ with the covariance matrix estimated by processing the respective uncertain blade $\underline{\mathbf{X}_{\langle r\rangle}}$ with the proposed mapping procedure (Sections 6.3.1 and 6.3.3). These two covariance matrices were compared with the distance function described by Bogner (1981). Such a function receives as input a true (or estimated) distribution (*i.e.,* in this case, the covariance matrix computed with first-order error propagation) and a set of observations (*i.e,* the points resulting from mapping random samples to $\mathbb{A}^m$). Bogner's function (BOGNER, 1981) returns a real value D as a measure of the distance between the two distributions and the expected theoretical variance for the distance, computed from the size of the matrix. This variance can be used to interpret the resulting values as a similarity measurement.

Table 6.5 shows the distances computed for the examples depicted in Tables 6.3 and 6.4. Here, the heights of the surfaces represent the distance D computed by Bogner's

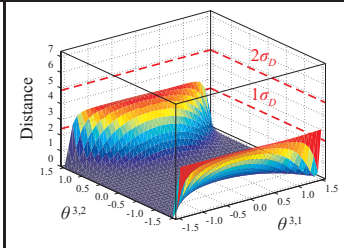
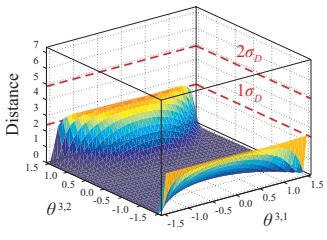
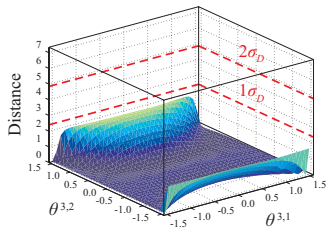
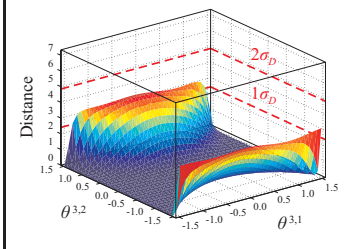
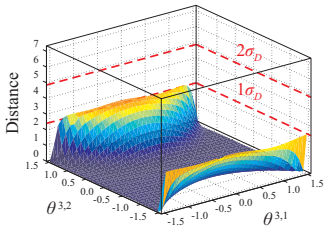
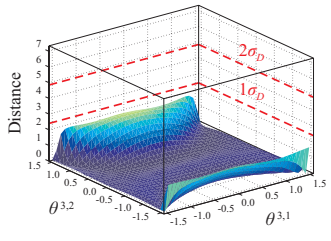
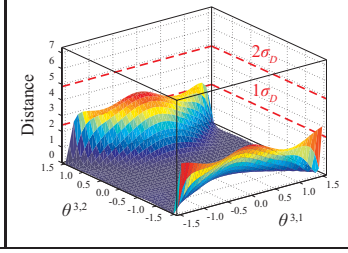
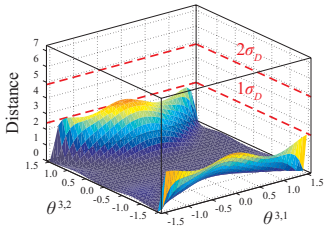
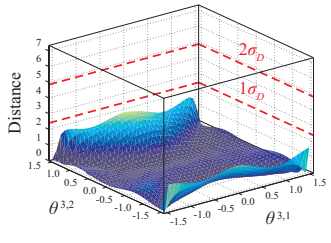
procedure (BOGNER, 1981). For these examples, the theoretical variance is equal to:

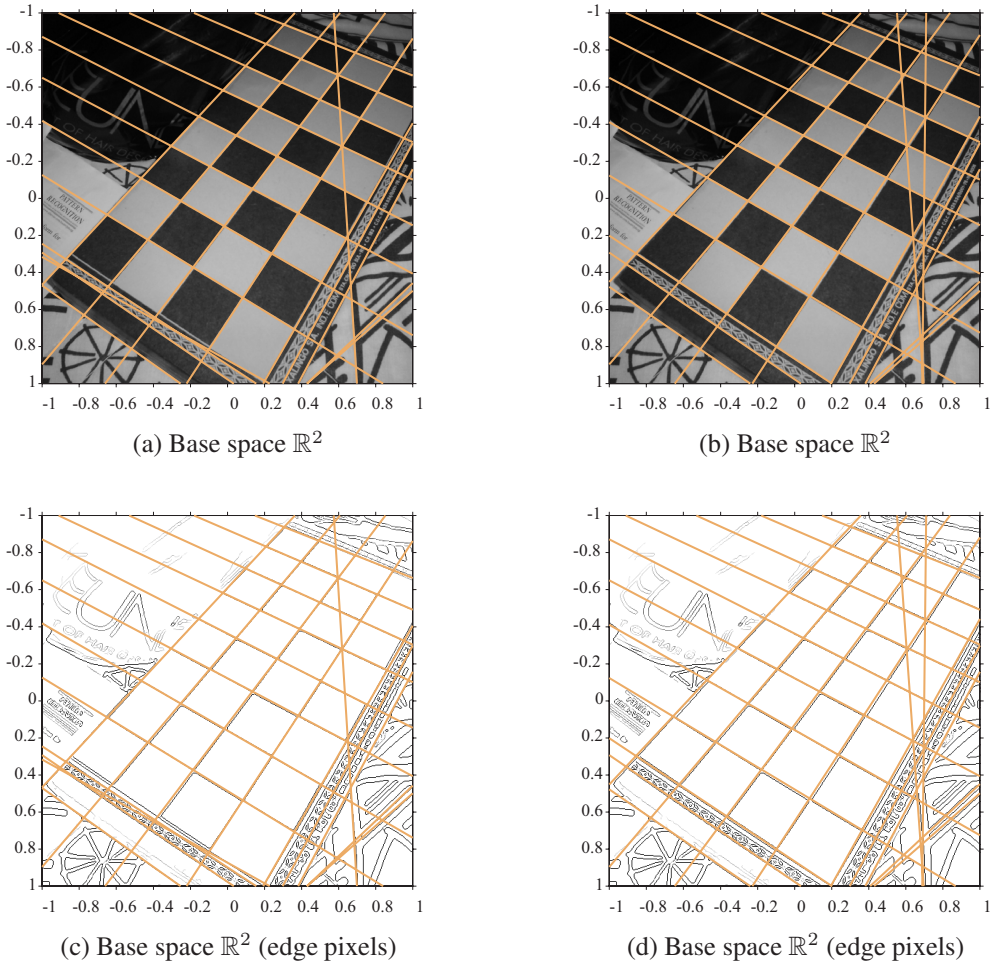
$$\sigma_D^2 = m\,(m+1) = 2\,(2+1) = 6,$$

where m is the dimensionality of the distribution (in this case, the same as the dimensionality of $\mathbb{A}^2$). The charts in Table 6.5 denote one and two standard deviations of D by dashed red lines. Notice that all the distances are below $1.5\,\sigma_D$. These results show that the Gaussian distribution estimated with first-order error propagation is consistent with the Gaussian distribution of samples in $\mathbb{A}^2$. A behavior that can be further investigated is why the propagated distribution fits better the samples for larger values of σ_ρ and σ_ϕ.

A clear advantage of the voting procedure based on first-order error propagation over a sampling-based approach is the reduced computational load of the former. It is because, with first-order analysis, only one uncertain blade needs to be processed per entry of the input dataset. With a sampling-based voting procedure, on the other hand, hundreds of samples must be generated and processed in order to properly represent the uncertainty on a given input entry. Another advantage is the possibility of spreading smoother distributions of values over the bins of the accumulator array. Such a feature improves the

Table 6.5: Measure of the similarity between Gaussian distributions approximated with first-order error propagation analysis and a respective distribution of points obtained through a sampling-based approach. This table displays results for the same experiments carried out for Tables 6.3 and 6.4. The heights of the surfaces represent the distance between two sets of covariance matrices. The first one was estimated with first-order error propagation while mapping uncertain blades $\underline{\mathbf{X}_{\langle 2\rangle}}$ to $\mathbb{A}^2$. The second set of matrices was computed from points resulting from mapping random samples of $\underline{\mathbf{X}_{\langle 2\rangle}}$ to $\mathbb{A}^2$. These results show that first-order analysis provides a good approximation for the distribution of samples. Notice that the computed distances are lower than 1.5 standard deviations (σ_D).

	$\sigma_\phi = 0.09$	$\sigma_\phi = 0.13$	$\sigma_\phi = 0.17$
$\sigma_\rho = 0.02$			
$\sigma_\rho = 0.04$			
$\sigma_\rho = 0.08$			



(a) Base space $\mathbb{R}^2$

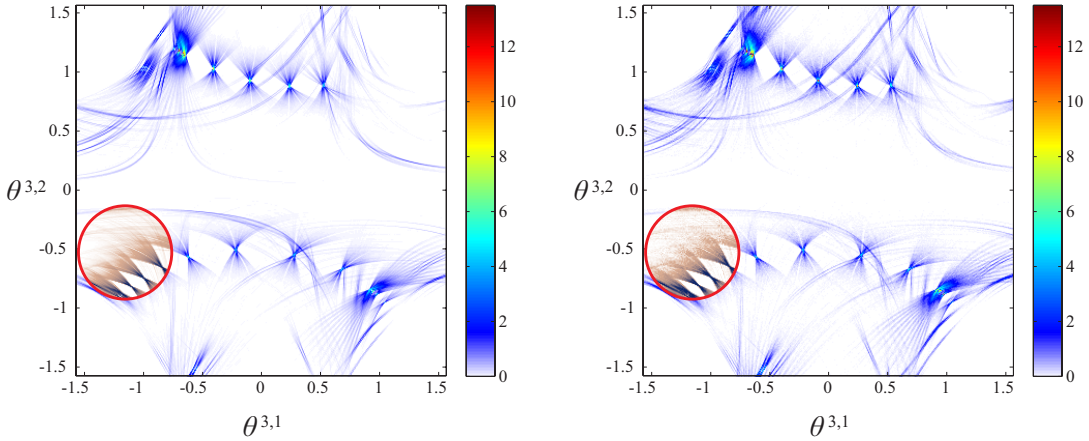
(b) Base space $\mathbb{R}^2$

(c) Base space $\mathbb{R}^2$ (edge pixels)

(d) Base space $\mathbb{R}^2$ (edge pixels)

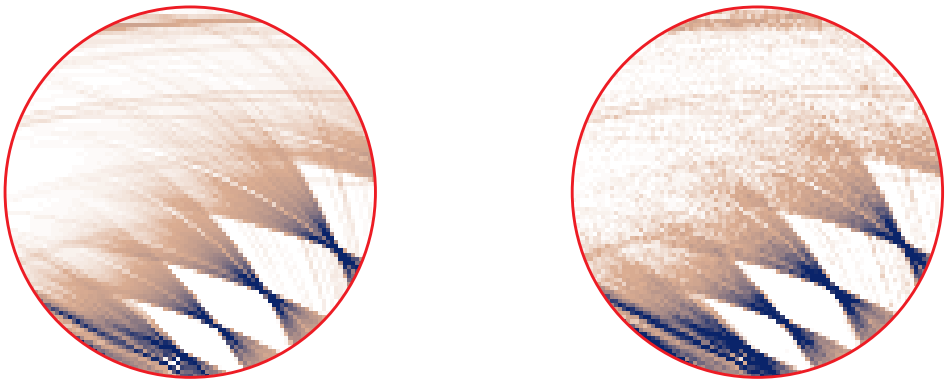
Figure 6.5: The 22 most relevant detect lines obtained using the proposed first-order error-propagation-based voting scheme (a) and a sampling-based approach (b). These results where obtained from the edge information shown in (c) and (d), respectively.

identification of local maxima in the resulting map of votes by reducing the occurrence of spurious peaks of votes. Figure 6.6 presents a comparison between the accumulator array produced in straight-line detection in Figure 6.5 with the technique described in the current chapter (Figures 6.6a) and the sampling-based voting using the technique described in Chapter 5 (Figure 6.6b). Figures 6.6c and 6.6d show a detailed view of the highlighted portions in Figures 6.6a and 6.6b, respectively. Notice the smoother transitions of votes produced by the error-propagation-based technique. In these example, the input dataset is comprised by $15{,}605$ uncertain subspaces encoding straight lines in the 2-dimensional homogeneous MOG. The input 2-blades were computed from the edge pixels of the image (Figures 6.5c and 6.5d) and their gradient vector directions. The standard deviation for coordinates of a given edge pixel is $2/(512\sqrt{12})$, where 2 is the size of the image after normalizing its coordinates to the $[-1, +1]$ range, 512 is the number of pixels in each dimension of the image, and 12 come from the second central moment of a pixel with unit side length. The standard deviation assumed for gradient directions was 0.13, leading to ± 0.39 radians of uncertainty on the direction normal to a given input line. The accumulator arrays where obtained as the linear discretization of the parameter space $\mathbb{P}^2$, using $\pi/360$ as discretization step. The importance value ω of each input is the magnitude



(a) Accumulator array for $\mathbb{P}^{2\,(3-2)}$

(b) Accumulator array for $\mathbb{P}^{2\,(3-2)}$



(c) Highlighted portion of (a)

(d) Highlighted portion of (b)

Figure 6.6: The accumulator arrays produced for Figures 6.5a and 6.5b are shown in (a) and (b), respectively. The highlighted portions are presented in detail by (c) and (d). The colors where changed in order to improve the contrast of the accumulated values. Notice that first-order error propagation (c) produces smoother distributions of votes than the sampling-based approach (d). As a result, the former is less prone to the detection of spurious subspaces.

of the gradient computed by the edge detector. For the sampling-based voting procedure, each one of the $15{,}605$ uncertain blades was represented by 160 random samples.

By processing various datasets it was observed that the approximation assumed for computing the number of votes to be incremented to a given bin of the accumulator array (Figure 6.4c) affects the resulting distributions of votes by a small scaling factor. When the accumulator arrays are shown in grayscale, the voting maps produced with the sampling-based approach seem a little more brighter than the ones produced with first-order error propagation. Also, it was observed a small shift (up to one bin) on the location of some peaks. Such a displacement is not uniform in the whole parameter space. It is important to comment that while a displacement occurs it seems not to have affected the quality of detections.

7 PEAK DETECTION

The last step of the subspace detection framework is performed after the voting procedure has been applied to all input data entries. It consists in identifying the bins that correspond to local maxima in the accumulator array. This chapter describes the sweep-hyperplane-based peak detection scheme developed for accumulator arrays having arbitrary dimensionality. The proposed approach is an extension of the peak detection technique described in (FERNANDES; OLIVEIRA, 2008) for 2-dimensional accumulator arrays. The technique returns a list with all detected vote peaks, sorted according to their importance (*i.e.,* number of votes).

Given an m-dimensional accumulator array, the first step is to create a list pointing to all bins that received at least a given minimum number of votes (τ_{min}). Then, this list is sorted in descending order according to the result of the convolution of the bins with a discrete hypercubic Gaussian kernel G_{filter} of side S_{filter} and variance σ^2_{filter}. For the examples presented in this dissertation, $S_{\text{filter}} = 5$, $\sigma^2_{\text{filter}} = 0.25$, and the threshold τ_{min} was set as the lowest importance value (ω) among the input blades, times the number of input blades required to define an intended p-dimensional subspace. However, it is important to comment that fine-tunned values for S_{filter} and σ^2_{filter} may improve the identification of peaks of votes.

The entries of the discrete Gaussian kernel are computed using:

$$G^i_{\text{filter}} = \frac{1}{A_{\text{filter}}} \exp\left(-\frac{1}{2\,\sigma^2_{\text{filter}}} \sum_{t=1}^{m} \left(i^t - \frac{S_{\text{filter}}+1}{2}\right)^2\right),$$

where A_{filter} is a normalization factor computed as:

$$A_{filter} = \sum_{i \in \mathcal{I}} \left(\exp\left(-\frac{1}{2\,\sigma^2_{\text{filter}}} \sum_{t=1}^{m} \left(i^t - \frac{S_{\text{filter}}+1}{2}\right)^2\right)\right),$$

and $i \in \mathcal{I}$ denotes a multi-index addressing a cell in G_{filter}, with

$$\mathcal{I} = \left\{i = (i^1, i^2, \cdots, i^m) \mid i^t \in \mathbb{Z}, \text{ and } 1 \leq i^t \leq S_{\text{filter}}\right\}$$

being the set of all multi-indices i.

The filtering operation smoothes the accumulator array, helping to consolidate adjacent peaks as single ones which characterize an intended subspace. Figure 7.1 presents the detection of the 39 most relevant lines in a given image. Figures 7.1a and 7.1c show the obtained result with the use of convolution, whereas for Figures 7.1b and 7.1d no convolution has been applied. Note the recovered lines labeled with (A), (B) and (C) on Figures 7.1a and 7.1c. These lines fit to edge pixels which, when mapped to parameter

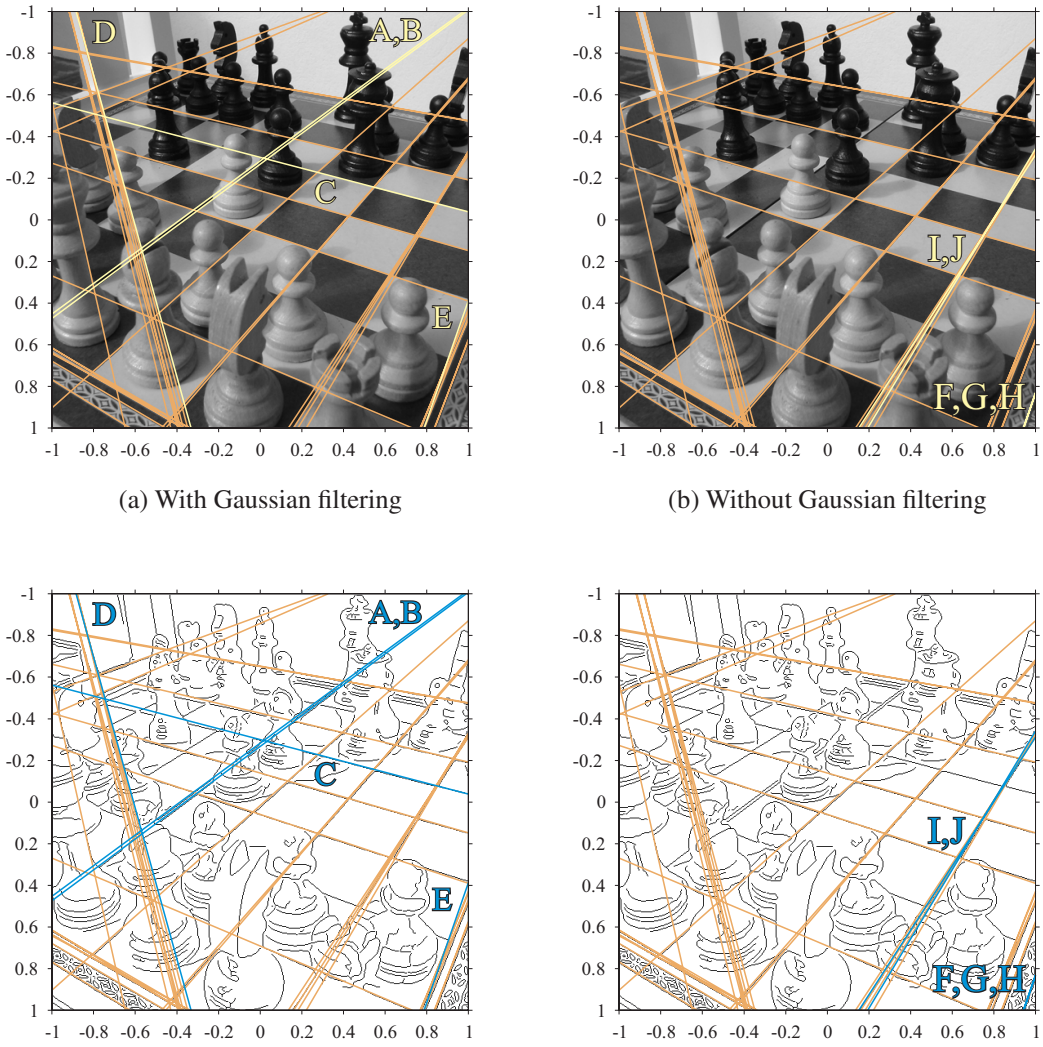
space, define unsharp peaks of votes at neighbor bins of the accumulator array. The convolution combines the peaks into larger ones, resulting in three significant lines. For the result shown in Figures 7.1b and 7.1d, since no convolution has been applied, the three lines were missed and replaced by some other less significant features. For instance, the peak detection applied to Figure 7.1b retrieved three concurrent lines very close to the corner of the image, labeled by (F), (G), and (H), which are less relevant. The lines denoted by (D) and (E) in Figure 7.1a, and (I) and (J) in Figure 7.1b, correspond to other lines detected in one image but not in another. The voting procedure presented in Chapter 6 was used in the examples depicted in Figure 7.1. The discretization step for defining the accumulator array is $\pi/1800$, and the importance value of each input is $\omega = 1$.

After the sorting step, the elements of the list are visited from the first to the last one. By treating the accumulator array as a height map (*i.e.,* the accumulated values define heights of a hypersurface in a $(m+1)$-dimensional space), visiting the elements of the list is analogous to gradually moving a sweeping hyperplane from each peak toward the τ_{min} height. For each visited element, it is checked if any of its vertex-connected neighbors (*i.e.* the $(3^m - 1)$ bins in the accumulator array sharing at least one vertex with current bin) has already been visited. If so, the current bin should be in the decay of some local maxima (*i.e.,* it is at a lower portion of the hypersurface, next to a taller one). In such a case, the bin is marked as visited and the procedure moves the sweeping hyperplane to the next element in the list. Otherwise, the coordinates of the current bin is added to the list of detected peaks, the bin is marked as visited, and then, the sweeping hyperplane is moved to the next element in the list. The procedure stops after all elements have been visited by the sweeping hyperplane, or after a given number of bins have been added to the list of detected peaks. The resulting group of detected peaks contains the parameter vectors of the most significant p-dimensional subspaces identified in the input dataset, already sorted by number of votes.

7.1 Discussion

The proposed peak detection scheme was used in all examples of subspace detection presented in this work. The worst case scenario for the approach is achieved by requiring the detection of all peaks of votes (even the negligible ones) and by setting $\tau_{\text{min}} = 0$. In this case, the convolution is applied to every bin in the accumulator array. Also, all bins are visited by the sweeping hyperplane, leading to redundant verifications of visiting status of neighbor bins. However, in practice, the value of τ_{min} can be defined in such a way that bins having negligible values are not included in the visitation list used by the sweeping hyperplane. This is achieved, for instance, by computing τ_{min} as a function of the smaller importance value ω in the input dataset, and the number of input entries used to define an intended subspace. Also, a typical situation is to know in advance how many resulting subspaces one wants to detect (or at least be capable to deduce when to stop reporting detected peaks). This way, the redundant work of checking the visiting status of neighbors is performed only for a small subset of bins close to the main peaks of votes.

While implementing the described algorithm, one may include an extra bin at the beginning and another one at the end of each dimension of the accumulator array. Such extra bins never receive votes. However, they allow one to treat all actual bins in the same way. For instance, the use of extra bins avoids testing whether a given actual bin is on the edge of the accumulator array while checking the visiting status of its neighbors. Also, by assuming that the accumulator array is allocated in a continuous block of memory,



(a) With Gaussian filtering

(b) Without Gaussian filtering

(c) With Gaussian filtering (edge pixels)

(d) Without Gaussian filtering (edge pixels)

Figure 7.1: The 39 most relevant detected lines obtained using the voting scheme presented in Chapter 6 and the proposed peak detection procedure on the edge pixels (shown in (c) and (d)) of a given image (shown in (a) and (b)). In (a) and (c), a 5×5 Gaussian kernel was convolved with the accumulator array before performing peak detection. On (b) and (d), no convolution was applied prior the peak detection. Note that some important subspaces (denoted by (A), (B) and (C) in (a)) were not detected in (b).

addressing neighbors of any (actual) bin is achieved by adding some offset to the memory address of the current bin.

8 CONCLUSIONS AND FUTURE WORK

The analysis of large volumes of unordered multidimensional data is a problem confronted by scientists and data analysts every day. Often, it involves searching for data alignments that emerge as well-defined structures or geometric patterns in datasets. The development of automatic detectors has been explored and extended in many ways in order to produce techniques specialized in specific types of alignments of a given type of input. This research has presented a more general framework for detecting data alignments. The proposed approach is independent of the geometric properties of the alignments to be detected, as well as independent of the type of input data. This closing chapter presents a summary of the major points discussed in this dissertation. It also provides additional discussions on the generality claim and on the limitations of the approach. The chapter ends with a list of suggested areas for future exploration.

8.1 Synopsis

This thesis has presented a framework for detecting emerging data alignments in unordered noisy multidimensional data. This approach results from the observation that a wide class of alignments can be represented as linear subspaces in some multidimensional space. Thus, instead of defining a different detector for each specific case and input data type, it is possible to design a unifying framework to detect the occurrences of emerging subspaces in multidimensional datasets. The proposed subspace detector is based on a voting strategy, and it is formulated with GA. By doing so, the technique takes advantage of the conceptual simplicity of the voting paradigm for pattern recognition, while exploring the superior modeling capability of computational primitives and operations in GA.

The concepts of GA used in this dissertation have been discussed. The geometric intuition of its operations have been illustrated with examples, and the interpretation of its primitive structures have been presented in different MOGs.

I have described a parameterization for p-dimensional subspaces embedded in some n-dimensional vector space, for $0 \leq p \leq n$. In such a parameterization, the attitude of a subspace is characterized by $p\,(n-p)$ angular values describing rotation operations in well defined planes. The stance of a subspace is retrieved from its parameters by applying rotation operations on a canonical subspace with dimensionality p. By definition, the attitude is a nonmetric property of subspaces. Thus, the proposed parameterization is independent of the actual geometric interpretation of a given subspace. The number of parameters used by the proposed approach is in accordance with the smallest set of parameters required to characterize a p-dimensional subspace. As a consequence, the data alignments encoded by such subspaces are also represented in the most compact way.

I have presented two algorithms for mapping subspaces of arbitrary dimensionality r

(*i.e.,* $0 \leq r \leq n$) to the parameter space defined for p-dimensional subspaces. The mapping procedure identifies in the parameter space all p-blades that contain the given input entry (for $r \leq p$), or all p-blades contained in the input (for $r \geq p$). The first mapping algorithm assumes that input entries are exact. The second algorithm extends the former by taking into account the uncertainty intrinsic to experimental data while performing the mapping procedure. In such a case, the input data entries are modeled with random variables, where the uncertainty is described by some Gaussian probability distribution function. The mapping of uncertain data is achieved by using the affine cover of the Grassmannian as an alternative space where the uncertainty of arbitrary geometric primitives can be handled in an unified fashion.

The subspace detection framework uses a discrete representation of the parameter space in order to verify which are the most likely p-dimensional subspaces in input data. The mapping procedure is used to spread votes in such a discrete accumulator array. After all input subspaces have been mapped to parameter space and the bins related to such a mapping have been incremented, the peaks of votes in the accumulator array identify which are the subspaces that best fit the dataset. I have presented an algorithm for detecting local maxima in the accumulator array. The technique retrieves the group of detected peaks, already sorted by number of votes.

The techniques proposed in this dissertation have been demonstrated by implementations of the algorithms described. The implementations have been validated by applying the subspace detection framework in real and synthetic datasets.

8.2 Discussion

8.2.1 Generality of the Proposed Framework

As one would expect, the proposed framework is limited to the detection of elements that can be represented by linear subspaces in some multidimensional space. However, it is sufficient for detecting a wide class of data alignments. Examples of alignments include, but are not limited to:

- Euclidean subspaces in Euclidean MOG;
- Flats and directions encoded as blades in the homogeneous MOG;
- Flats, rounds, directions and tangents represented in the conformal MOG; and
- General conic sections in the conic MOG.

It is important to notice that, since the approach is independent of the metric space, it can be used without any change even in some MOG that may be defined in the future. Perwass (2009) provides a discussion on how general the geometries modeled in GA can be. As pointed out by Perwass (2009), a multivector space may be defined over a space other than a real vector space. The particular examples considered in his book are the Hilbert space of random variables and the Hilbert space of the basis functions of a finite Fourier series. In all cases, the concepts of blades and combinations of subspaces are still valid even though they may not have the same geometric meaning than in conventional MOGs.

Regarding the extension presented in Chapter 6, it is limited to input subspaces with Gaussian distributed uncertainty. It is because first-order error propagation analysis had been used to propagate the uncertainty of each input element throughout the computations. The treatment of non-Gaussian distribution would require a more cumbersome error propagation scheme. For instance, some approach based on Monte Carlo.

8.2.2 Usability

The application of the generalized subspace detection scheme is straightforward. It requires no initial guess or thresholding. The definition of the parameter space requires only the dimensionality p of the subspace interpreted as the intended type of data alignment, and the dimensionality n of the representational space. Thus, a MOG must be assumed. The input data must be encoded in such a MOG, and the detected alignments (*i.e.,* the p-dimensional subspaces) will be represented in the same MOG. As pointed out in Appendix B, the conversion between GA primitives and the representations typically used with linear algebra is simple and, in many situations, no extra computation is required. Such a conversion will not be necessary in cases where GA is adopted as the mathematical system of the whole solution.

The construction of the accumulator array requires some discretization criterion. For the examples presented in this dissertation the accumulator array was defined as the linear discretization of the parameter space. The discretization step was chosen regarding the amount of memory available in the computational system.

8.2.3 Memory Requirements and Computational Load

A naive implementation of the proposed approach suffers from the same drawbacks as conventional HTs: large memory requirement and computational cost. However, as any HT, the proposed approach is robust to the presence of outliers and is suitable for implementation on massively parallel architectures. Moreover, the generality of the technique guarantees that any optimization immediately benefits the processing of all detectable data alignments. The attempts to minimize drawbacks in conventional HTs, on the other hand, are targeted at particular versions of HTs due to specificities in their formulations (ILLINGWORTH; KITTLER, 1988; LEAVERS, 1993).

8.3 Future Work

Some possible areas for further exploration include the design of generally applicable optimizations, the study of the proposed parameter space in order to perform the detection of manifolds with boundary, and the detection of arbitrary shapes.

8.3.1 Generally Applicable Optimizations

Mapping existing HT optimizations for the proposed framework constitutes some promising direction for future exploration. For instance, I believe that the *Statistical Hough Transform* described by Dahyot (2009) for straight-line detection could be generalized to the subspace detector. This would overcome limitations due to the use of a discrete accumulator array. Such limitations are related to the trade-off in between the number of bins in the accumulator, the number of available input entries and the number of alignments observed in data. Too many bins for too few observations would lead to a sparse representation of the importance of the alignments. Too few bins would also reduce the resolution in the parameter space, and therefore, limit the precision of the estimates. Dahyot (2009) states that limitations related to the discrete accumulator array can be avoided by replacing it by a continuous representation of the parameter space.

The generalization of the real-time line detection scheme presented in (FERNANDES; OLIVEIRA, 2008) operates on clusters of approximately collinear pixels and seems to be a promising workaround for reducing the computation load of the proposed subspace

detector. For each cluster, votes are cast in the accumulator array by using an oriented elliptical-Gaussian kernel that models the uncertainty associated with the best-fitting line with respect to the corresponding cluster. Such an approach not only significantly improves the performance of the voting procedure for line detection, but also produces a much cleaner accumulator array and makes the HT more robust to the detection of spurious lines. The generalization of such optimization consists of defining a generalized clustering algorithm for input subspaces with arbitrary dimensionality. The voting procedure for the uncertainty associated to the best-fitting p-dimensional subspace is precisely the extension presented in Chapter 6.

The more we delve into the computer vision and image processing literature, the more optimizations for conventional HTs we find (ILLINGWORTH; KITTLER, 1988; LEAVERS, 1993). Some of these techniques may not have contributed significantly to advancing the state of the art of HT simply because they have not been applied in the detection of the same type of data alignment. However, by combining them into a generalized detection framework, such optimizations may be key for allowing the definition of an approach having low computational burden and memory requirements.

8.3.2 Detection of Manifolds with Boundary

The conventional HT specialized in detecting straight lines in images provides the parameters of the lines but not the length or the end points of line segments. Atiquzzaman and Akhtar (1995) had observed that line segments can be determined by analyzing the distribution of votes around the peaks in an accumulator array for line detection. The novelty of their algorithm with respect to other solutions (RICHARDS; CASASENT, 1991; ATIQUZZAMAN; AKHTAR, 1994) is its independence of the accuracy with which the peaks can be detected. Kierkegaard (1992), and Pei and Horng (1995) had used a similar analysis to determine the parameters of circular arcs from the accumulator array of conventional HT variations for circle detection. Such an analysis can be extended to accumulator arrays computed with the subspace detection framework described in this dissertation. By doing so, it will be possible to perform the detection of a broad class of manifolds with boundaries. Such boundaries could include line segments, circular and elliptical arcs, spherical and ellipsoidal caps, among others.

8.3.3 Detection of Arbitrary Shapes

Finally, I believe that the detection of arbitrary shapes with the proposed framework can be achieved by defining a MOG where arbitrary structures can be represented (or at least approximated) by pseudovectors in some high-dimensional space. A promising direction for future investigation is to explore radial basis functions (RBFs) in the representation of smooth manifolds and to define a multivector space over such a functional basis. RBFs have been successfully applied in the reconstruction of 3-dimensional surfaces from point-cloud data obtained from range scanners (CARR et al., 2001). In such a case, RBFs approximate the function that describes the surface that one wants to reconstruct. Such a problem is analogous to the representation of arbitrary shapes. The drawback of this approach is the dimensionality of the functional basis, which leads to a high-dimensional accumulator array.

REFERENCES

ABBOTT, B. et al. All-sky search for periodic gravitational waves in LIGO S4 data. **Phys. Rev. D**, College Park, MD, v.77, n.2, p.022001, Jan. 2008.

ACHTERT, E. et al. Robust clustering in arbitrarily oriented subspaces. In: SIAM INT. CONF. DATA MIN., 8., Atlanta, GA. **Proceedings...** SIAM, 2008. p.763–774.

ANDRES, E. Discrete linear objects in dimension n: the standard model. **Graph. Models**, San Diego, CA, v.65, n.1-3, p.92–111, May 2003.

ATIQUZZAMAN, M.; AKHTAR, M. W. Complete line segment description using the Hough transform. **Image Vis. Comput.**, Amsterdam, Netherlands, v.12, n.5, p.267–273, June 1994.

ATIQUZZAMAN, M.; AKHTAR, M. W. A robust Hough transform technique for complete line segment description. **Real-Time Imaging**, London, UK, v.1, n.6, p.419–426, Dec. 1995.

BALLARD, D. H. Generalizing the Hough transform to detect arbitrary shapes. **Pattern Recognit.**, Oxford, England, v.13, n.2, p.111–122, April 1981.

BENNETT, N.; BURRIDGE, R.; SAITO, N. A method to detect and characterize ellipses using the Hough transform. **IEEE Trans. Pattern Anal. Mach. Intell.**, Los Alamitos, CA, v.21, n.7, p.652–657, July 1999.

BEWES, J. M.; SUCHOWERSKA, N.; MCKENZIE, D. R. Automated cell colony counting and analysis using the circular Hough image transform algorithm (CHiTA). **Phys. Med. Biol.**, [S.l.], v.53, n.21, p.5991–6008, Nov. 2008.

BOGNER, R. E. Pattern recognition via observation correlations. **IEEE Trans. Pattern Anal. Mach. Intell.**, Los Alamitos, CA, v.PAMI-3, n.2, p.128–133, March 1981.

BöHM, C. et al. Computing clusters of correlation connected objects. In: ACM INT. CONF. ON MANAG. OF DATA, Paris, France. **Proceedings...** ACM, 2004. p.455–466.

BROWN, C. M. Computer vision and natural constraints. **Science**, New York, NY, v.224, n.4655, p.1299–1305, June 1984.

BROWNE, J. **Grassmann algebra**: exploring applications of extended vector algebra with Mathematica. Melbourne, Australia: Swinburne University of Technology, 2001. Incomplete draft. Available at: <http://sites.google.com/site/grassmannalgebra>. Access in: fev. 2002.

CANNY, J. A computational approach to edge detection. **IEEE Trans. Pattern Anal. Mach. Intell.**, Los Alamitos, CA, v.8, n.6, p.679–698, Nov. 1986.

CARR, J. C. et al. Reconstruction and representation of 3D objects with radial basis functions. In: ACM SIGGRAPH, 28., Los Angeles, CA. **Proceedings...** ACM Press, 2001. p.67–76.

COWAN, G. **Statistical data analysis**. Oxford: Oxford University Press, 1998. 197p.

DAHYOT, R. Statistical Hough transform. **IEEE Trans. Pattern Anal. Mach. Intell.**, Los Alamitos, CA, v.31, n.8, p.1502–1509, Aug. 2009.

DING, M.; FENSTER, A. A real-time biopsy needle segmentation technique using Hough transform. **Med. Phys.**, Melville, NY, v.30, n.8, p.2222–2233, Aug. 2003.

DORST, L.; FONTIJINE, D.; MANN, S. **Geometric algebra for computer science**: an object oriented approach to geometry. Amsterdam: Morgan Kaufmann Publishers, 2007. 664p. (The Morgan Kaufmann Series in Computer Graphics).

DUDA, R. O.; HART, P. E. Use of the Hough transformation to detect lines and curves in pictures. **Commun. ACM**, New York, NY, v.15, n.1, p.11–15, Jan. 1972.

EULER, L. Du mouvement de rotation des corps solides autour d'un axe variable. **Mémoires de l'académie des sciences de Berlin**, Berlin, Germany, v.14, p.154–193, 1765.

FECHNER, M. et al. Kinematic reconstruction of atmospheric neutrino events in a large water Cherenkov detector with proton identification. **Phys. Rev. D**, College Park, MD, v.79, n.11, p.112010, June 2009.

FERNANDES, L. A. F.; OLIVEIRA, M. M. Real-time line detection through an improved Hough transform voting scheme. **Pattern Recognit.**, New York, NY, v.41, n.1, p.299–314, Jan. 2008.

FERNANDES, L. A. F.; OLIVEIRA, M. M. Geometric algebra: a powerful tool for solving geometric problems in visual computing. In: TUTORIALS OF SIBGRAPI 2009 (XXII BRAZILIAN SYMP. COMPUT. GRAPH. IMAGE PROCESS.), Rio de Janeiro, RJ, Brazil. **Proceedings...** IEEE, 2009. p.17–30.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Commun. ACM**, New York, NY, v.24, n.6, p.381–395, June 1981.

FONTIJNE, D. Gaigen 2: a geometric algebra implementation generator. In: INT. CONF. GENERATIVE PROGR. COMPONENT ENG., 5., Portland, OR. **Proceedings...** ACM Press, 2006. p.141–150.

FONTIJNE, D. Efficient algorithms for factorization and join of blades. In: INT. CONFER. APPL. GEOM. ALGEBR. COMPUT. SCI. ENG. (AGACSE), 3., Grimma, Germany. **Proceedings...** [S.l.: s.n.], 2008.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. 3rd.ed. Upper Saddle River, NJ: Prentice Hall, 2008. 954p.

GUY, G.; MEDIONI, G. Inference of surfaces, 3-D curves, and junctions from sparse 3-D data. In: IEEE INT. SYMP. COMPUT. VIS., Coral Gables, FL. **Proceedings...** IEEE Computer Society, 1995. p.599–604.

HAEBERLI, P. Paint by numbers: abstract image representations. In: ACM SIGGRAPH, 17., Dallas, TX. **Proceedings...** ACM Press, 1990. p.207–214.

HARRIS, J. **Algebraic geometry**: a first course. New York, NY: Springer-Verlag, 1992. 328p.

HARTLEY, R. I.; ZISSERMAN, A. **Multiple view geometry in computer vision**. 2nd.ed. Cambridge, UK: Cambridge University Press, 2004. 655p.

HESTENES, D. **New foundations for classical mechanics**. Dordrecht, Netherlands: Reidel Publishing Company, 1987.

HESTENES, D. Old wine in new bottles: a new algebraic framework for computational geometry. In: BAYRO-CORROCHANO, E.; SOBCZYK, G. (Ed.) **Geometric algebra with applications in science and engineering**. Birkhäuser: Boston, 2001. p. 3-17.

HESTENES, D.; ZIEGLER, R. Projective geometry with Clifford algebra. **Acta Appl. Math.**, Dordrecht, Netherlands, v.23, n.1, p.25–63, April 1991.

HOFF, K. E. et al. Fast computation of generalized Voronoi diagrams using graphics hardware. In: ACM SIGGRAPH, 26., Los Angeles, CA. **Proceedings...** ACM Press, 1999. p.277–286.

HOFFMAN, D. K.; RAFFENETTI, R. C.; RUEDENBERG, K. Generalization of Euler angles to n-dimensional orthogonal matrices. **J. Math. Phys.**, Melville, NY, v.13, n.4, p.528–533, April 1972.

HOUGH, P. V. C. Machine analysis of bubble chamber pictures. In: INT. CONF. ON HIGH ENERGY ACCELERATORS AND INSTRUM., Geneva. **Proceedings...** CERN, 1959.

HOUGH, P. V. C. **Methods and means for recognizing complex patterns**. U.S. Patent 3.069.654, 1962.

ILLINGWORTH, J.; KITTLER, J. A survey of the Hough transform. **Comput. Vis. Graph. Image Process.**, New York, NY, v.44, n.1, p.87–116, Oct. 1988.

KIERKEGAARD, P. A method for detection of circular arcs based on the Hough transform. **Mach. Vis. Appl.**, New York, NY, v.5, n.4, p.249–263, Sept. 1992.

KIMME, C.; BALLARD, D.; SKLANSKY, J. Finding circles by an array of accumulators. **Commun. ACM**, New York, NY, v.18, n.2, p.120–122, Feb. 1975.

KOBATAKE, H.; YOSHINAGA, Y. Detection of spicules on mammogram based on skeleton analysis. **IEEE Trans. Med. Imaging**, Piscataway, NJ, v.15, n.3, p.235–245, Jun. 1996.

KRISHNAN, B. et al. Hough transform search for continuous gravitational waves. **Phys. Rev. D**, College Park, MD, v.70, n.8, p.082001, Oct. 2004.

KüRNER, J.; FRANGAKIS, A. S.; BAUMEISTER, W. Cryo-electron tomography reveals the cytoskeletal structure of Spiroplasma Melliferum. **Science**, New York, NY, v.307, n.5708, p.436–438, Jan. 2005.

LAM, W. C. Y. et al. An analysis on quantizing the Hough space. **Pattern Recognit. Lett.**, Amsterdam, Netherlands, v.15, n.11, p.1127–1135, Nov. 1994.

LEAVERS, V. F. Which Hough transform? **CVGIP: Image Underst.**, Orlando, FL, v.58, n.2, p.250–264, Sept. 1993.

LEOPARDI, P. C. **GluCat**: generic library of universal Clifford algebra templates. Available at: <http://glucat.sourceforge.net>. Access in: out. 2009.

LI, H. Hyperbolic geometry with Clifford algebra. **Acta Appl. Math.**, Dordrecht, Netherlands, v.48, n.3, p.317–358, Sept. 1997.

LIU, T.; RAABE, D.; ZAEFFERER, S. A 3D tomographic EBSD analysis of a CVD diamond thin film. **Sci. Technol. Adv. Mater.**, Bristol, England, v.9, n.3, p.035013, Sept. 2008.

LOUNESTO, P. **Clifford algebras and spinors**. 2nd.ed. Cambridge, UK: Cambridge University Press, 2001. 338p. (London Mathematical Society lecture note series, v.286).

MANKEL, R. Pattern recognition and event reconstruction in particle physics experiments. **Rep. Prog. Phys.**, Bristol, England, v.67, n.4, p.553–622, April 2004.

MARDIA, K. V. Measures of multivariate skewness and kurtosis with applications. **Biometrika**, Oxford, England, v.57, n.3, p.519–530, 1970.

MEDIONI, G.; LEE, M.-S.; TANG, C.-K. **A computational framework for segmentation and grouping**. Amsterdam: Elsevier, 2000. 260p.

NAUMOVIć, D. et al. Formation of a stable decagonal quasicrystalline Al-Pd-Mn surface layer. **Phys. Rev. Lett.**, College Park, MD, v.87, n.19, p.195506, Oct. 2001.

O'GORMAN, F.; CLOWES, M. B. Finding picture edges through collinearity of feature points. In: JOINT CONF. ARTIF. INTELL., 3., Stanford, CA. **Proceedings...** Morgan Kaufmann Publishers Inc., 1973. p.543–555.

PEI, S.-C.; HORNG, J.-H. Circular arc detection based on Hough transform. **Pattern Recognit. Lett.**, Amsterdam, Netherlands, v.16, n.6, p.615–625, June 1995.

PERWASS, C. **Geometric algebra with applications in engineering**. New York: Springer Publishing Company, 2009. 386p.

PERWASS, C.; FÖRSTNER, W. Uncertain geometry with circles, spheres and conics. In: KLETTE, R. et al. (Ed.) **Geometric properties for incomplete data**. Springer-Verlag: New York, NY, 2006. p. 23-41.

RAGURAM, R.; FRAHM, J.-M.; POLLEFEYS, M. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: EUR. CONF. COMPUT. VIS. (ECCV), 10., Marseille, France. **Proceedings...** Springer-Verlag, 2008. p.500–513.

RICHARDS, J.; CASASENT, D. P. Extracting input-line position from Hough transform data. **Appl. Optics**, Washington, DC, v.30, n.20, p.2899–2905, July 1991.

SCHNABEL, R.; WAHL, R.; KLEIN, R. Efficient RANSAC for point-cloud shape detection. **Comput. Graph. Forum**, Malden, MA, v.26, n.2, p.214–226, May 2007.

SKLANSKY, J. On the Hough technique for curve detection. **IEEE Trans. Comput.**, Los Alamitos, CA, v.C-27, n.10, p.923–926, Oct. 1978.

TANG, C.-K.; MEDIONI, G.; LEE, M.-S. N-dimensional tensor voting and application to epipolar geometry estimation. **IEEE Trans. Pattern Anal. Mach. Intell.**, Los Alamitos, CA, v.23, n.8, p.829–844, Aug. 2001.

VEEN, T. M. van; GROEN, F. C. A. Discretization errors in the Hough transform. **Pattern Recognit.**, Oxford, England, v.14, n.1-6, p.137–145, Jan. 1981.

VORONOI, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. **J. Reine Angew. Math. (Crelle's J.)**, Berlin, Germany, v.1908, n.134, p.198–287, Jan. 1908.

WANG, H. L.; REEVES, A. P. Three-dimensional generalized Hough transform for object identification. In: SPIE, INTELL. ROBOTS AND COMP. VIS. VIII, Philadelphia. **Proceedings...** SPIE, 1990. v.1192, p.363–374.

APPENDIX A DEFINITION OF THE GEOMETRIC PRODUCT

This appendix presents the algebraic definition of the **geometric product** (or **Clifford product**) of two general blades (see Section 3.5). By exploring the distributivity over the sum, one can use the formulas presented in this appendix to implement the geometric product involving pairs of basis blades in a multivector.

Before one can define the geometric product, it is necessary to introduce the **co-basis elements** (Section A.1), the **complement**, and the **regressive product** (Section A.2). These operations are used in the definition of the **interior product** and the **generalized Grassmann product** (Section A.3), which are key for defining the geometric product of blades (Section A.4). The following equations are based on the formulas described by Browne in (BROWNE, 2001).

A.1 Co-Basis

The basis $\{\mathbf{e}_i\}_{i=1}^{n}$ of a vector space $\mathbb{R}^n$ induces the definition of a co-basis. The co-basis element related to a basis vector $\mathbf{e}_i$ is denoted by $\underline{\mathbf{e}_i}$ (do not confuse the co-basis notation used in this appendix with the random variable notation used in Chapter 6). It is defined as the outer product of the remaining basis elements. As a result, the outer product of a given basis element with its related co-basis element is equal to the unit positive pseudoscalar of the n-dimensional space:

$$\mathbf{e}_i \wedge \underline{\mathbf{e}_i} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_n = \mathbf{I}_{\langle n \rangle}.$$

This way, the co-basis of $\mathbf{e}_i$ is:

$$\underline{\mathbf{e}_i} = (-1)^{i-1}\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \breve{\mathbf{e}}_i \wedge \cdots \wedge \mathbf{e}_n, \tag{A.1}$$

where $\breve{\mathbf{e}}_i$ indicates that $\mathbf{e}_i$ is not present in this sequence of outer products. The definition of co-basis is valid for n-dimensional spaces with any metric.

A.2 Complement and Regressive Product

The complement of a basis vector $\mathbf{e}_i$ is computed as:

$$\overline{\mathbf{e}_i} = \delta \sum_{j=1}^{n} \mu^{i,j}\, \underline{\mathbf{e}_j}, \tag{A.2}$$

where $\underline{\mathbf{e}_j}$ is the co-basis element (Equation A.1) of $\mathbf{e}_j$ and δ is a constant scalar value, computed from the metric matrix $\mathrm{M} = (\mu^{i,j})_{n\times n}$ as:

$$\delta = \frac{1}{\sqrt{\det(\mathrm{M})}}.$$

The notion of complement can be extended from basis vectors to basis blades because the complement of a basis blade $\mathbf{E}_{\langle m\rangle} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_m$ is defined in terms of the basis vectors spanning it:

$$\overline{\mathbf{E}}_{\langle m\rangle} = \overline{\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_m} = \overline{\mathbf{e}_1} \vee \overline{\mathbf{e}_2} \vee \cdots \vee \overline{\mathbf{e}_m}. \tag{A.3}$$

Again, it is important not to confuse the complement notation used in Equations A.2 and A.3 for the complement, with the notation used for the expectation of a random variable in Chapter 6.

In Equation A.3, $\vee$ denotes the regressive product (Section 3.3.2). The regressive product of two arbitrary blades $\mathbf{A}_{\langle r\rangle}$ and $\mathbf{B}_{\langle s\rangle}$ is compute as:

$$\mathbf{A}_{\langle r\rangle} \vee \mathbf{B}_{\langle s\rangle} = \begin{cases} \mathbf{C}_{\langle t\rangle} & \text{for } (r+s+t) = n \\ 0 & \text{otherwise} \end{cases}, \tag{A.4}$$

where n is the number of dimensions of $\mathbb{R}^n$, and

$$\mathbf{C}_{\langle t\rangle} \equiv \left(\mathbf{A'}_{\langle r-t\rangle} \wedge \mathbf{B'}_{\langle s-t\rangle} \wedge \mathbf{C}_{\langle t\rangle}\right) \vee \mathbf{C}_{\langle t\rangle},$$

for $\mathbf{A}_{\langle r\rangle} = \mathbf{A'}_{\langle r-t\rangle} \wedge \mathbf{C}_{\langle t\rangle}$, and $\mathbf{B}_{\langle s\rangle} = \mathbf{B'}_{\langle s-t\rangle} \wedge \mathbf{C}_{\langle t\rangle}$. Note that the regressive product is computed by finding the subspace $\mathbf{C}_{\langle t\rangle}$ shared by $\mathbf{A}_{\langle r\rangle}$ and $\mathbf{B}_{\langle s\rangle}$. For basis blades, $\mathbf{C}_{\langle t\rangle}$ is found simply by checking which basis vectors are used to span the input blades.

A.3 Generalized Grassmann Product and Interior Product

The equations presented so far are used in the definition of the generalized Grassmann product of order k:

$$\mathbf{A}_{\langle r\rangle} \underset{k}{\triangle} \mathbf{B}_{\langle s\rangle} = \sum_{j=1}^{\nu} \left(\mathbf{A}_{\langle r\rangle} \ominus \mathbf{B}^{(j)}_{\langle k\rangle}\right) \wedge \mathbf{B}^{(j)}_{\langle s-k\rangle}, \tag{A.5}$$

where $\nu = \binom{s}{k}$, $\ominus$ denotes the interior product (Equation A.7), and $\wedge$ is the outer product. In the generalized Grassmann product, the blade $\mathbf{B}_{\langle s\rangle}$ is represented using all the $j = \{1,2,\cdots,\nu\}$ combinations of its factors, separated in a blade of grade k and another one of grade $(s-k)$:

$$\begin{aligned} \mathbf{B}_{\langle s\rangle} &= \mathbf{B}^{(j)}_{\langle k\rangle} \wedge \mathbf{B}^{(j)}_{\langle s-k\rangle} \\ &= \mathbf{B}^{(1)}_{\langle k\rangle} \wedge \mathbf{B}^{(1)}_{\langle s-k\rangle} = \cdots = \mathbf{B}^{(\nu)}_{\langle k\rangle} \wedge \mathbf{B}^{(\nu)}_{\langle s-k\rangle}. \end{aligned} \tag{A.6}$$

The factors of $\mathbf{B}_{\langle s\rangle}$ are the s vectors used in its construction. For instance, by assuming $\mathbf{B}_{\langle 3\rangle} = \mathbf{b}_1 \wedge \mathbf{b}_2 \wedge \mathbf{b}_3$ and $k = 2$, Equation A.6 results in:

$$\begin{aligned} \mathbf{B}_{\langle 3\rangle} &= \mathbf{B}^{(1)}_{\langle 2\rangle} \wedge \mathbf{B}^{(1)}_{\langle 1\rangle} = \quad (\mathbf{b}_1 \wedge \mathbf{b}_2) \wedge (\mathbf{b}_3) \\ &= \mathbf{B}^{(2)}_{\langle 2\rangle} \wedge \mathbf{B}^{(2)}_{\langle 1\rangle} = -(\mathbf{b}_1 \wedge \mathbf{b}_3) \wedge (\mathbf{b}_2) \\ &= \mathbf{B}^{(3)}_{\langle 2\rangle} \wedge \mathbf{B}^{(3)}_{\langle 1\rangle} = \quad (\mathbf{b}_2 \wedge \mathbf{b}_3) \wedge (\mathbf{b}_1)\,. \end{aligned}$$

The generalized Grassmann product (Equation A.5) is composed by a part which does not depend on the metric of the space (*i.e.,* the outer product at the right side of the equation) and a part that depends on the metric (*i.e.,* the interior product inside the parenthesis). The interior product of blades $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{F}_{\langle k \rangle}$ (where $\mathbf{F}_{\langle k \rangle} \equiv \mathbf{B}^{(j)}_{\langle k \rangle}$ is not necessarily equal to $\mathbf{B}_{\langle s \rangle}$) is computed as:

$$\mathbf{A}_{\langle r \rangle} \ominus \mathbf{F}_{\langle k \rangle} = \begin{cases} \mathbf{A}_{\langle r \rangle} \vee \overline{\mathbf{F}}_{\langle k \rangle} & \text{for } r \geq k \\ 0 & \text{otherwise} \end{cases}, \tag{A.7}$$

where $\vee$ is the regressive product (Equation A.4) and $\overline{\mathbf{F}}_{\langle k \rangle}$ is the complement of $\mathbf{F}_{\langle k \rangle}$, as presented in Equation A.3 for basis blades.

A.4 Geometric Product

Finally, the geometric product of blades $\mathbf{A}_{\langle r \rangle}$ and $\mathbf{B}_{\langle s \rangle}$ is expressed as:

$$\mathbf{A}_{\langle r \rangle} \mathbf{B}_{\langle s \rangle} = \sum_{k=0}^{\min(r,s)} (-1)^{k(r-\frac{k+1}{2})} \left(\mathbf{A}_{\langle r \rangle} \underset{k}{\triangle} \mathbf{B}_{\langle s \rangle} \right),$$

where $\underset{k}{\triangle}$ is the generalized Grassmann product of order k, defined in Equation A.5.

APPENDIX B GEOMETRIC PRIMITIVES AS BLADES

This appendix presents a collection of equations related to geometric algebra. It aims at being a quick reference guide for defining geometric primitives as blades in some models of geometry (MOGs), from parameters that are typically used with linear algebra. For an in-depth discussions on the subject, see (DORST; FONTIJINE; MANN, 2007; PERWASS, 2009).

Homogeneous Model (d-dimensional base space)

Construction of Oriented Flats	
direction vector	$\mathbf{v} = \alpha^1 \mathbf{e}_1 + \alpha^2 \mathbf{e}_2 + \cdots + \alpha^d \mathbf{e}_d$
k-dimensional direction	$\mathbf{D}_{\langle k \rangle} = \mathbf{v}_1 \wedge \mathbf{v}_2 \wedge \cdots \wedge \mathbf{v}_k$
finite point (0-flat)	$\mathbf{p} = \gamma \left(\mathbf{e}_0 + \alpha^1 \mathbf{e}_1 + \alpha^2 \mathbf{e}_2 + \cdots + \alpha^d \mathbf{e}_d \right)$
straight line (1-flat)	$\mathbf{L}_{\langle 2 \rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2$
plane (2-flat)	$\mathbf{P}_{\langle 3 \rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \mathbf{p}_2$
finite k-flat	$\mathbf{F}_{\langle k+1 \rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \cdots \wedge \mathbf{p}_{k+1}$
	$\mathbf{F}_{\langle k+1 \rangle} = \mathbf{p} \wedge \mathbf{A}_{\langle k \rangle}$
	where $\mathbf{p}$ is its support point, and
	$\mathbf{A}_{\langle k \rangle} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ is its direction.
hyperplane ($(d-1)$-flat)	$\mathbf{H}_{\langle d \rangle} = (-\mathbf{n} + \delta \, \mathbf{e}_0^{-1})^{-*}$
	where $\mathbf{n} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ is its unit normal
	vector, and δ is its distance from the origin.

Basis Vectors
$\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_d\}$
$\mathbf{e}_0$: point at origin

Inner Product Table

$\cdot$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\cdots$	$\mathbf{e}_d$
$\mathbf{e}_0$	± 1	0	0	$\cdots$	0
$\mathbf{e}_1$	0	1	0	$\cdots$	0
$\mathbf{e}_2$	0	0	1	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{e}_d$	0	0	0	$\cdots$	1

Geometric Interpretation and Parameters of $\mathbf{X}_{\langle k \rangle}$

	finite flat	*direction*
condition	$\mathbf{e}_0 \rfloor \mathbf{X}_{\langle k \rangle} \neq 0$	$\mathbf{e}_0 \rfloor \mathbf{X}_{\langle k \rangle} = 0$
direction ($\mathbf{A}_{\langle t \rangle}$)	$\mathbf{e}_0^{-1} \rfloor \mathbf{X}_{\langle k \rangle}$	$\mathbf{X}_{\langle k \rangle}$
moment ($\mathbf{M}_{\langle t+1 \rangle}$)	$\mathbf{e}_0^{-1} \rfloor (\mathbf{e}_0 \wedge \mathbf{X}_{\langle k \rangle})$	–
support vector ($\mathbf{d}$)	$\mathbf{M}_{\langle k+1 \rangle} \,/\, \mathbf{A}_{\langle k \rangle}$	–
unit support point ($\mathbf{e}_0 + \mathbf{d}$)	$\mathbf{X}_{\langle k \rangle} \,/\, \mathbf{A}_{\langle k \rangle}$	–

Construction of Oriented Tangent Subspaces

tangent scalar (finite point) $\mathbf{p} = \gamma\,(\mathbf{o} + \mathbf{l} + \frac{1}{2}\mathbf{l}^2\,\boldsymbol{\infty})$

where $\mathbf{l} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ defines the location of the point.

tangent at $\mathbf{p}$ $\mathbf{T}_{\langle k+1\rangle} = \mathbf{p} \wedge (-\mathbf{p} \rfloor (\widehat{\mathbf{A}}_{\langle k\rangle}\,\boldsymbol{\infty}))$

where $\mathbf{A}_{\langle k\rangle} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ defines its direction, and $\mathbf{p}$ is a finite point.

tangent to $\mathbf{X}_{\langle k\rangle}$ *at* $\mathbf{p}$ $\mathbf{T}_{\langle k-1\rangle} = \mathbf{p} \rfloor \widehat{\mathbf{X}}_{\langle k\rangle} = \mathbf{p}\,\widehat{\mathbf{X}}_{\langle k\rangle}$

where $\mathbf{X}_{\langle k\rangle}$ is a round or a flat, and $\mathbf{p} \subset \mathbf{X}_{\langle k\rangle}$.

Construction of Oriented Rounds

point pair (0-sphere) $\mathbf{K}_{\langle 2\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2$

circle (1-sphere) $\mathbf{C}_{\langle 3\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \mathbf{p}_3$

ordinary sphere (2-sphere) $\mathbf{O}_{\langle 4\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \mathbf{p}_3 \wedge \mathbf{p}_4$

k-sphere $\mathbf{S}_{\langle k+2\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \cdots \wedge \mathbf{p}_{k+2}$

$\mathbf{S}_{\langle k+2\rangle} = (\mathbf{c} + \frac{1}{2}\,(\rho)^2\,\boldsymbol{\infty}) \wedge (-\mathbf{c} \rfloor (\widehat{\mathbf{A}}_{\langle k\rangle}\,\boldsymbol{\infty}))$

where $\mathbf{c}$ is its center, ρ is its radius, and $\mathbf{A}_{\langle k\rangle} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ defines the direction of the carrier flat.

$(d-1)$-sphere $\mathbf{S}_{\langle d+1\rangle} = \mathbf{p} \wedge (\mathbf{c} \wedge \boldsymbol{\infty})^{-*}$

gives the sphere around $\mathbf{c}$ through $\mathbf{p}$.

Basis Vectors

$\{\mathbf{o}, \mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_d, \boldsymbol{\infty}\}$

$\mathbf{o}$: null point at origin (“no”)

$\boldsymbol{\infty}$: null point at infinity (“ni”)

Inner Product Table

$\cdot$	$\mathbf{o}$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\cdots$	$\mathbf{e}_d$	$\boldsymbol{\infty}$
$\mathbf{o}$	0	0	0	$\cdots$	0	-1
$\mathbf{e}_1$	0	1	0	$\cdots$	0	0
$\mathbf{e}_2$	0	0	1	$\cdots$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{e}_d$	0	0	0	$\cdots$	1	0
$\boldsymbol{\infty}$	-1	0	0	$\cdots$	0	0

Point Pair Decomposition

$$\mathbf{p}_\pm = \frac{\mathbf{K}_{\langle 2\rangle} \mp \sqrt{\mathbf{K}_{\langle 2\rangle}^2}}{-\boldsymbol{\infty} \rfloor \mathbf{K}_{\langle 2\rangle}}$$

where $\mathbf{K}_{\langle 2\rangle} = \mathbf{p}_- \wedge \mathbf{p}_+$

Conformal Model (d-dimensional base space) – Part 2/3

Construction of Oriented Finite Flats		Geometric Interpretation of $\mathbf{X}_{\langle k\rangle}$	
finite flat point (0-flat)	$\mathbf{Q}_{\langle 2\rangle} = \mathbf{p} \wedge \boldsymbol{\infty}$	*direct/dual tangent*	$\boldsymbol{\infty} \wedge \mathbf{X}_{\langle k\rangle} \neq 0$
straight line (1-flat)	$\mathbf{L}_{\langle 3\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \boldsymbol{\infty}$		$\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k\rangle} \neq 0$
plane (2-flat)	$\mathbf{P}_{\langle 4\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \mathbf{p}_3 \wedge \boldsymbol{\infty}$		$\mathbf{X}^2_{\langle k\rangle} = 0$
finite k-flat	$\mathbf{F}_{\langle k+2\rangle} = \mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \cdots \wedge \mathbf{p}_{k+1} \wedge \boldsymbol{\infty}$		
	$\mathbf{F}_{\langle k+2\rangle} = \mathbf{p} \wedge \mathbf{A}_{\langle k\rangle} \wedge \boldsymbol{\infty}$	*direct/dual round*	$\boldsymbol{\infty} \wedge \mathbf{X}_{\langle k\rangle} \neq 0$
	where $\mathbf{p}$ is its support point, and		$\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k\rangle} \neq 0$
	$\mathbf{A}_{\langle k\rangle} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ defines its direction.		$\mathbf{X}^2_{\langle k\rangle} \neq 0$
hyperplane ($(d-1)$-flat)	$\mathbf{H}_{\langle d+1\rangle} = (\mathbf{n} + \delta\,\boldsymbol{\infty})^{-*}$		
	where $\mathbf{n} \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$ is its unit normal	*direct finite flat*	$\boldsymbol{\infty} \wedge \mathbf{X}_{\langle k\rangle} = 0$
	vector, and δ is its distance from the origin.		$\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k\rangle} \neq 0$
	$\mathbf{H}_{\langle d+1\rangle} = \mathbf{p} \wedge (\mathbf{n} \wedge \boldsymbol{\infty})^{-*}$		
	gives the hyperplane with normal $\mathbf{n}$, through $\mathbf{p}$.	*dual finite flat*	$\boldsymbol{\infty} \wedge \mathbf{X}_{\langle k\rangle} \neq 0$
	$\mathbf{H}_{\langle d+1\rangle} = (\mathbf{p} + \mathbf{q})^{-*}$		$\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k\rangle} = 0$
	gives the mid-hyperplane between unit $\mathbf{p}$ and $\mathbf{q}$.		
Construction of Oriented Directions		*direct/dual direction*	$\boldsymbol{\infty} \wedge \mathbf{X}_{\langle k\rangle} = 0$
k-dimensional direction	$\mathbf{D}_{\langle k\rangle} = \mathbf{v}_1 \wedge \mathbf{v}_2 \wedge \cdots \wedge \mathbf{v}_k \wedge \boldsymbol{\infty}$		$\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k\rangle} = 0$
	where $\mathbf{v}_i \subset (\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_d)$.		

Conformal Model (d-dimensional base space) – Part 3/3

Parameters of $\mathbf{X}_{\langle k \rangle}$			
	direction ($\mathbf{A}_{\langle t \rangle} \boldsymbol{\infty}$)	*location (finite point* $\mathbf{p}$*)*	*squared radius* (ρ^2)
direct tangent	$(-\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle}) \wedge \boldsymbol{\infty}$	$\dfrac{\mathbf{X}_{\langle k \rangle}}{-\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle}}$	0
dual tangent	$(-\boldsymbol{\infty} \rfloor \mathbf{X}^{-*}_{\langle k \rangle}) \wedge \boldsymbol{\infty}$	$\dfrac{\mathbf{X}_{\langle k \rangle}}{-\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle}}$	0
direct round	$(-\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle}) \wedge \boldsymbol{\infty}$	$-\frac{1}{2} \dfrac{\mathbf{X}_{\langle k \rangle} \boldsymbol{\infty} \mathbf{X}_{\langle k \rangle}}{(\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle})^2}$	$\dfrac{\mathbf{X}_{\langle k \rangle} \widehat{\mathbf{X}}_{\langle k \rangle}}{(\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle})^2}$
dual round	$(-\boldsymbol{\infty} \rfloor \mathbf{X}^{-*}_{\langle k \rangle}) \wedge \boldsymbol{\infty}$	$-\frac{1}{2} \dfrac{\mathbf{X}_{\langle k \rangle} \boldsymbol{\infty} \mathbf{X}_{\langle k \rangle}}{(\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle})^2}$	$-\dfrac{\mathbf{X}_{\langle k \rangle} \widehat{\mathbf{X}}_{\langle k \rangle}}{(\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle})^2}$
direct finite flat	$-\boldsymbol{\infty} \rfloor \mathbf{X}_{\langle k \rangle}$	$(\mathbf{o} \rfloor \mathbf{X}_{\langle k \rangle}) \mathbf{X}^{-1}_{\langle k \rangle}$	–
dual finite flat	$-\boldsymbol{\infty} \rfloor \mathbf{X}^{-*}_{\langle k \rangle}$	$(\mathbf{o} \wedge \mathbf{X}_{\langle k \rangle}) \mathbf{X}^{-1}_{\langle k \rangle}$	–
direct direction	$\mathbf{X}_{\langle k \rangle}$	–	–
dual direction	$\mathbf{X}^{-*}_{\langle k \rangle}$	–	–

APPENDIX C PARAMETERIZATION OF ROTORS

The Euler angles characterize the orientation of a rigid body in 3-dimensional Euclidean space (EULER, 1765). Hoffman *et al.* (1972) presented a generalization of the Euler angles for a rigid body in n-dimensional Euclidean space. As described by Hoffman *et al.* (1972), the generalized Euler angles may be used to characterize an arbitrary rotation matrix in terms of $n\,(n-1)/2$ independent parameters. This appendix presents the derivations proposed by Hoffman *et al.* (1972), originally with linear algebra, which I adapted to geometric algebra. In such a context, the generalized Euler angles are used in the parameterization of rotors.

C.1 Generalized Euler Angles

Let $\mathbf{a}$ be an arbitrary vector in Euclidean vector space $\mathbb{R}^n$ with orthonormal basis $\{\mathbf{e}_i\}_{i=1}^{n}$. The direction of $\mathbf{a}$ (*i.e.,* the combination of attitude and orientation, Section 3.1) may be expressed by $(n-1)$ rotation operations applied to a reference vector. Without loss of generality, one may choose $\mathbf{e}_n$ as the canonical vector and $\{\mathbf{e}_n \wedge \mathbf{e}_j\}_{j=1}^{n-1}$ as the rotation planes. By assuming the domain of the rotation angles as:

$$-\frac{\pi}{2} \leq \phi^1, \phi^2, \cdots, \phi^{n-2} \leq \frac{\pi}{2}$$
$$-\pi \leq \phi^{n-1} < \pi,$$

all possible directions of $\mathbf{a}$ are characterized by a set of parameters $\{\phi^j\}_{j=1}^{n-1}$, where ϕ^j is the angle (in radians) related to the rotation on plane $\mathbf{e}_n \wedge \mathbf{e}_j$.

Now, let $\{\mathbf{a}_i\}_{i=1}^{n}$ be another orthonormal basis in $\mathbb{R}^n$. By making $\mathbf{a}_n$ be the vector $\mathbf{a}$ in previous example, we know that $(n-1)$ rotations are necessary to map $\mathbf{e}_n$ to $\mathbf{a}_n$. In the notational convention of this appendix, the angles of such rotations are denoted by $\{\phi^{n,j}\}_{j=1}^{n-1}$. The operations describing the direction of $\mathbf{a}_n$ also affect vectors $\mathbf{a}_1$, $\mathbf{a}_2$, $\cdots$, $\mathbf{a}_{n-1}$. Thus, there is an $(n-1)$-dimensional space subjacent to $\mathbf{a}_n$ where $(n-2)$ rotation operations may be used to describe the direction of $\mathbf{a}_{n-1}$ with respect to $\mathbf{e}_{n-1}$. The rotation angles used in such a subjacent space are depicted as $\{\phi^{n-1,j}\}_{j=1}^{n-2}$. This recursive definition shows that

$$\sum_{u=2}^{n} (u-1) = \frac{n\,(n-1)}{2}$$

rotations map a basis $\{\mathbf{e}_i\}_{i=1}^{n}$ to $\{\mathbf{a}_i\}_{i=1}^{n}$ and vice versa. The domain of the parameter $\phi^{u,j}$,

for $u \in \{2, 3, \cdots, n\}$ and $j \in \{1, 2, \cdots, u-1\}$, is given by:

$$-\frac{\pi}{2} \leq \phi^{u,1}, \phi^{u,2}, \cdots, \phi^{u,u-2} \leq \frac{\pi}{2} \qquad (C.1)$$
$$-\pi \leq \phi^{u,u-1} < \pi.$$

C.2 Computing a Rotor from its Generalized Euler Angles

A rotor $\boldsymbol{W}$ encoding an arbitrary rotation in Euclidean n-dimensional space is defined from its generalized Euler angles $\{\phi^{u,j}\}$ by the composition of rotations on planes $\mathbf{e}_u \wedge \mathbf{e}_j$:

$$\boldsymbol{W} = \boldsymbol{V}_n \cdots \boldsymbol{V}_3 \boldsymbol{V}_2, \qquad (C.2)$$

where

$$\boldsymbol{V}_u = \boldsymbol{U}_{u,u-1} \cdots \boldsymbol{U}_{u,2} \boldsymbol{U}_{u,1} \qquad (C.3)$$

is the sequence of $(u-1)$ rotation operations used to redirect the vector $\mathbf{e}_u$ in an u-dimensional space. A rotor $\boldsymbol{U}_{u,j}$ in Equation C.3 encodes a rotation of $\phi^{u,j}$ radians on plane $\mathbf{e}_u \wedge \mathbf{e}_j$. It is computed as:

$$\boldsymbol{U}_{u,j} = \cos\left(\frac{\phi^{u,j}}{2}\right) - \sin\left(\frac{\phi^{u,j}}{2}\right)(\mathbf{e}_u \wedge \mathbf{e}_j).$$

Recall from Section 3.8 that rotors are applied to subspaces by using a sandwiching construction, where the rotation operations are evaluated from inside to outside of it. Therefore, by reading Equation C.2 from right to left, the geometric interpretation of the application of a rotor $\boldsymbol{W}$ is the successive transformation of vectors in a basis $\{\mathbf{e}_i^{(u)}\}_{i=1}^{u}$ to a basis $\{\mathbf{e}_i^{(u+1)}\}_{i=1}^{u+1}$, for $u \in \{2, 3, \cdots, n\}$.

C.3 Computing the Generalized Euler Angles of a Rotor

The procedure for computing the generalized Euler angles of an arbitrary rotor $\boldsymbol{W}$ is given by the recursive application of Equations C.4 to C.7, starting with $u = n$ and $\boldsymbol{W}^{(n)} = \boldsymbol{W}$. In such a procedure,

$$\left(\boldsymbol{W}^{(u)}\,\mathbf{e}_u \,/\, \boldsymbol{W}^{(u)}\right) / \mathbf{e}_u = \beta^{u,0} + \beta^{u,1}\,\mathbf{e}_1 \wedge \mathbf{e}_u + \beta^{u,2}\,\mathbf{e}_2 \wedge \mathbf{e}_u + \cdots + \beta^{u,u-1}\,\mathbf{e}_{u-1} \wedge \mathbf{e}_u \qquad (C.4)$$

is the square of the simplest rotor that transforms the vector $\mathbf{e}_u$ into the vector resulting from $\boldsymbol{W}^{(u)}\,\mathbf{e}_u \,/\, \boldsymbol{W}^{(u)}$. The right side of Equation C.4 can be used to compute the Euler angles related to $\boldsymbol{V}_u$ (Equation C.3):

$$\phi^{u,k} = \sin^{-1}\left(\frac{\beta^{u,k}}{\cos\phi^{u,1}\,\cos\phi^{u,2}\,\cdots\,\cos\phi^{u,k-1}}\right), \qquad (C.5)$$

for $k \in \{1, 2, \cdots, (u-2)\}$, and

$$\phi^{u,u-1} = \tan^{-1}\left(\frac{\beta^{u,u-1}}{\beta^{u,0}}\right). \qquad (C.6)$$

The angles computed with Equations C.5 and C.6 are used to retrieve $\boldsymbol{W}^{(u-1)}$ as:

$$\boldsymbol{W}^{(u-1)} = \boldsymbol{V}_u^{-1}\,\boldsymbol{W}^{(u)}, \qquad (C.7)$$

where $\boldsymbol{V}_u$ is defined in Equation C.3. The geometric product of $\boldsymbol{V}_u^{-1}$ and $\boldsymbol{W}^{(u)}$ means that all the transformations that affect the vector $\mathbf{e}_u$ are being removed from $\boldsymbol{W}^{(u)}$. The remaining transformations affect only the $(u-1)$-dimensional space subjacent to $\mathbf{e}_u$.

It is important to comment that the arctangent in Equation C.6 must be evaluated with the function ATAN2, available in many programming languages. By doing so, one guarantees that the limits defined in Equation C.1 will be respected while computing the parameters of a rotor.

APPENDIX D RASTERIZATION OF SIMPLICES

This appendix presents an algorithm that takes a k-simplex in continuous m-dimensional space and converts it to a raster representation in some discrete m-dimensional domain. The algorithm receives as input a set of $(k+1)$ points defining the vertices of the simplex in $\mathbb{P}^m$ (*i.e.,* the continuous parameter space for p-blades, Equation 4.10). In the proposed rasterization scheme, the discretization of the simplex is defined analytically by inequalities according to the *Standard Model* (ANDRES, 2003). Such inequalities impose a set of restrictions that are used to identify which hyxels in $\mathbb{D}^m$ (*i.e.,* the discrete domain) are related to the given simplex. A hyxel is a hyper-volume picture element, as the pixel in raster images and the voxel in raster volumes. In practice, each inequality defines a discrete half-space (*i.e.,* that portion of $\mathbb{D}^m$ obtained by removing from it that part lying on one side of a hyperplane). Thus, the discrete simplex is given by hyxels at the intersection of discrete half-spaces.

The discrete space $\mathbb{D}^m$ is defined as

$$\mathbb{D}^m = \left\{ \left(\delta^1, \delta^2, \cdots, \delta^m\right) \mid \delta^t \in \mathbb{N}_0 \right\},$$

where $\mathbb{N}_0$ denotes the set of all natural numbers including zero. A point in $\mathbb{D}^m$ can be regarded as the address of a bin in the m-dimensional accumulator array used by the voting procedure presented in Section 5.4.

In its initialization step, the rasterization algorithm defines a set $\mathcal{J}$ of multi-indices. The multi-indices are all the strictly growing sequences of d-combinations of the values in the set $\{1, 2, \cdots, m\}$, for $d \in \mathbb{Z}$ varying from 1 to $\min(k+1, m)$:

$$\mathcal{J} = \{ j \in \mathbb{N}^d \mid 1 \leq j^1 < j^2 < \cdots < j^d \leq m, 1 \leq d \leq \min(k+1, m) \}. \tag{D.1}$$

The multi-indices in $\mathcal{J}$ are used by the Standard Model (ANDRES, 2003) while computing the set of inequalities describing the discrete simplex. In the proposed rasterization scheme, the multi-indices have a slightly different definition (and usage) from the ones originally presented by Andres (2003). However, the resulting inequalities are equivalent.

After the initialization, the application of the algorithm consists of three steps:

1. Map vertices from $\mathbb{P}^m$ to the real vector space $\mathbb{R}^m$ containing $\mathbb{D}^m$;
2. Use the Standard Model to create a set of discrete half-spaces whose intersection define the intended discrete k-simplex; and
3. Identify the hyxels included in all half-spaces, and hence in the simplex.

Step (1) of the algorithm uses

$$\left(\alpha^1, \alpha^2, \cdots, \alpha^m\right) = \frac{1}{\Delta_\theta} \left(\theta^1 + \frac{\pi}{2}, \theta^2 + \frac{\pi}{2}, \cdots, \theta^m + \frac{\pi}{2}\right), \tag{D.2}$$

to map the vertices of the input simplex from $\mathbb{P}^m$ to the real vector space $\mathbb{R}^m$ containing $\mathbb{D}^m$. In Equation D.2, $(\alpha^1, \alpha^2, \cdots, \alpha^m)$ are the coordinates of a point in $\mathbb{R}^m$, Δ_θ is the linear step assumed for the discretization of $\mathbb{P}^m$, and $(\theta^1, \theta^2, \cdots, \theta^m) \in \mathbb{P}^m$ are the coordinates of one of the original vertices.

Step (2) generates the inequalities (*i.e.,* half-spaces) which will define the analytical discrete representation of the simplex. For each multi-index $j \in \mathcal{J}$ the rasterization algorithm builds a matrix with the form:

$$\mathrm{S} = \begin{pmatrix} \alpha^{1,j^1} & \alpha^{1,j^2} & \cdots & \alpha^{1,j^d} & 1 \\ \alpha^{2,j^1} & \alpha^{2,j^2} & \cdots & \alpha^{2,j^d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{k+1,j^1} & \alpha^{k+1,j^2} & \cdots & \alpha^{k+1,j^d} & 1 \end{pmatrix}. \tag{D.3}$$

The rows of S store the coordinates of the $(k+1)$ vertices represented in space $\mathbb{R}^m$ (Equation D.2) and, in turn, orthogonally projected onto the subspace spanned by the d basis vectors of $\mathbb{R}^m$ indexed by j. Notice that the projection is achieved just by discarding the coordinates which are not indexed by j.

When the rank of the matrix S (Equation D.3) is less than the number of basis vectors indexed by j (*i.e.,* $\mathrm{rank}\,(\mathrm{S}) < d$), it means that the projected simplex induces a flat subspaces with low dimensionality in the d-dimensional subspace. In such a case, no inequality needs to be generated because there is some other multi-index that will handle the projection of the simplex onto a lower dimensional subspace. The cases that generate inequalities are: $\mathrm{rank}\,(\mathrm{S}) = d$, and $\mathrm{rank}\,(\mathrm{S}) = d+1$.

The projected simplex induces a hyperplane in the d-dimensional space defined by current multi-index j when the rank of S is equal to d. In such a case, two inequalities are generated:

$$\begin{cases} -\gamma \leq \left(\beta^{j^1}\,\delta^{j^1} + \beta^{j^2}\,\delta^{j^2} + \cdots + \beta^{j^d}\,\delta^{j^d} + \beta^{m+1}\right) < \gamma & \text{if it has standard orientation} \\ -\gamma \leq -\left(\beta^{j^1}\,\delta^{j^1} + \beta^{j^2}\,\delta^{j^2} + \cdots + \beta^{j^d}\,\delta^{j^d} + \beta^{m+1}\right) < \gamma & \text{otherwise} \end{cases}. \tag{D.4}$$

In Inequation D.4, $(\delta^1, \delta^2, \cdots, \delta^m)$ are the coordinates of a hyxel in $\mathbb{D}^m$ to be tested later against the inequalities. The coefficients $(\beta^1, \beta^2, \cdots, \beta^{m+1})$ of the inequalities are computed as the entries at the last row of the matrix in D.5, while the threshold γ is computed using Equation D.6. Andres (2003) state that a half-space has a *standard orientation* if $\beta^1 = \beta^2 = \cdots = \beta^i = 0$ and $\beta^i > 0$, for any $1 \leq i \leq m$.

The coefficients of a half-space in some d-dimensional subspace induced by j are computed as the cofactors for the entries at the last row of the matrix:

$$\begin{pmatrix} \alpha^{l^1,j^1} & \alpha^{l^1,j^2} & \cdots & \alpha^{l^1,j^d} & 1 \\ \alpha^{l^2,j^1} & \alpha^{l^2,j^2} & \cdots & \alpha^{l^2,j^d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{l^d,j^1} & \alpha^{l^d,j^2} & \cdots & \alpha^{l^d,j^d} & 1 \\ \beta^{j^1} & \beta^{j^2} & \cdots & \beta^{j^d} & \beta^{m+1} \end{pmatrix}, \tag{D.5}$$

i.e.,

$$\beta^{j^t} = (-1)^{(d+1)\,t}\,\det \begin{pmatrix} \alpha^{l^1,j^1} & \alpha^{l^1,j^2} & \cdots & \alpha^{l^1,j^{t-1}} & \alpha^{l^1,j^{t+1}} & \cdots & \alpha^{l^1,j^d} & 1 \\ \alpha^{l^2,j^1} & \alpha^{l^2,j^2} & \cdots & \alpha^{l^2,j^{t-1}} & \alpha^{l^2,j^{t+1}} & \cdots & \alpha^{l^2,j^d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \\ \alpha^{l^d,j^1} & \alpha^{l^d,j^2} & \cdots & \alpha^{l^d,j^{t-1}} & \alpha^{l^d,j^{t+1}} & \cdots & \alpha^{l^d,j^d} & 1 \end{pmatrix},$$

where $1 \leq t \leq (d+1)$. The $(m-d)$ coefficients β^i not indexed by j in matrix D.5 are equal to *zero*. In this matrix, l denotes a set of indices indicating d linearly independent rows in S (Equation D.3). Once one has the coefficients $(\beta^1, \beta^2, \cdots, \beta^{m+1})$, the threshold used in Equations D.4 and D.7 is computed as:

$$\gamma = \frac{1}{2} \sum_{i=1}^{m} \mathrm{abs}\left(\beta^i\right). \tag{D.6}$$

Such a threshold is geometrically interpreted as an offset added to the hyperplane at the edge of the half-space, leading to the inclusion of hyxels crossed by the hyperplane while defining the discrete half-space.

When the rank of S (Equation D.3) is equal to $(d+1)$, the projected simplex defines a convex polytope (not necessarily a simplex) which induces the whole d-dimensional space. In this case, an inequality is generated for each facet of the polytope. From the convex hull of the projected vertices one can identify which points (rows) in matrix S lie on the vertices of the polytope, and also the facets defined by them. Thus, each facet has a multi-index l pointing to rows in S. The inequality for a given facet is written as:

$$\begin{cases} \left(\beta^{j^1}\,\delta^{j^1} + \beta^{j^2}\,\delta^{j^2} + \cdots + \beta^{j^d}\,\delta^{j^d} + \beta^{m+1}\right) < \gamma & \text{if it has standard orientation} \\ \left(\beta^{j^1}\,\delta^{j^1} + \beta^{j^2}\,\delta^{j^2} + \cdots + \beta^{j^d}\,\delta^{j^d} + \beta^{m+1}\right) \leq \gamma & \text{otherwise} \end{cases}, \tag{D.7}$$

where the coefficients $(\beta^1, \beta^2, \cdots, \beta^{m+1})$ and the threshold γ are computed using, respectively, Equations D.5 and D.6 with the multi-index l of the given facet. In Equation D.7, $(\delta^1, \delta^2, \cdots, \delta^m)$ are the coordinates of a hyxel in $\mathbb{D}^m$ to be tested later against the inequalities. It is important to notice that a half-space defined by Equation D.7 must include all the vertices of the polytope.

After all the inequalities have being computed, the step (3) of the rasterization algorithm identifies the hyxels related to the input k-simplex by testing, against the inequalities, all the hyxels inside the axis-aligned bounding box of the given input points in $\mathbb{D}^m$.

APPENDIX E RESUMO EXPANDIDO (EXTENDED ABSTRACT IN PORTUGUESE)

A análise de dados é um elemento fundamental na descoberta científica e na mineração de dados. Em diversos campos da ciência, a inspeção visual de dados experimentais é realizada com o objetivo de identificar coerência local acentuada. Essa coerência resulta do alinhamento de dados (em algum espaço multidimensional) e, usualmente, emerge como formas geométricas e padrões. Por exemplo, linhas retas e círculos aparecem como estruturas bem definidas na análise de imagens produzidas por difração de elétrons retroespelhados (Figura E.1a) e ensaios clonogênicos (Figura E.1b), respectivamente. Entretanto, quando um grande volume de dados precisa ser analisado, a inspeção visual torna-se impraticável. Por este motivo, detectores automáticos para tipos específicos de alinhamentos de dados têm sido amplamente empregados por cientistas em diferentes áreas, tais como física de partículas (MANKEL, 2004; FECHNER et al., 2009), astronomia (KRISHNAN et al., 2004; ABBOTT et al., 2008), microbiologia (BEWES; SUCHOWERSKA; MCKENZIE, 2008; KüRNER; FRANGAKIS; BAUMEISTER, 2005), cristalografia (NAUMOVIć et al., 2001; LIU; RAABE; ZAEFFERER, 2008) e medicina (KOBATAKE; YOSHINAGA, 1996; DING; FENSTER, 2003). Na mineração de dados para comércio eletrônico, grandes conjuntos de dados têm sido analisados na busca por padrões no comportamento e na preferência de consumidores (BöHM et al., 2004). Detectores automáticos também são o componente central de muitas aplicações em visão computacional (HARTLEY; ZISSERMAN, 2004) e processamento de imagens (GONZALEZ; WOODS, 2008). O objetivo de detectores automáticos é identificar as instâncias de um tipo de alinhamento requerido que melhor se ajustam a um conjunto de dados não ordenados, mesmo na presença de ruído e de descontinuidade.

Esta tese introduz uma abordagem para a detecção de alinhamentos em conjuntos de dados multidimensionais não ordenados em ruidosos. A abordagem proposta é baseada na observação de que uma vasta classe de alinhamentos (*e.g.,* linhas retas, planos, círculos, esferas, seções cônicas, entre outros) pode ser representada como subespaços lineares. Portanto, ao invés de definir um detector diferente para cada caso específico de alinhamento e tipo de dado de entrada, é possível o desenvolvimento de uma abordagem unificada para detectar a ocorrência de subespaços emergentes em conjuntos de dados multidimensionais. A abordagem proposta apresenta as seguintes propriedades:

1. É uma solução com formulação fechada para a detecção de subespaços de uma dada dimensionalidade em conjuntos de dados que podem ser heterogêneos e conter elementos (*i.e.,* subespaços) com dimensionalidades arbitrárias;
2. Pode ser aplicada, sem alterações, na detecção de qualquer classe de alinhamento de dados que possa ser caracterizada por um subespaço linear;

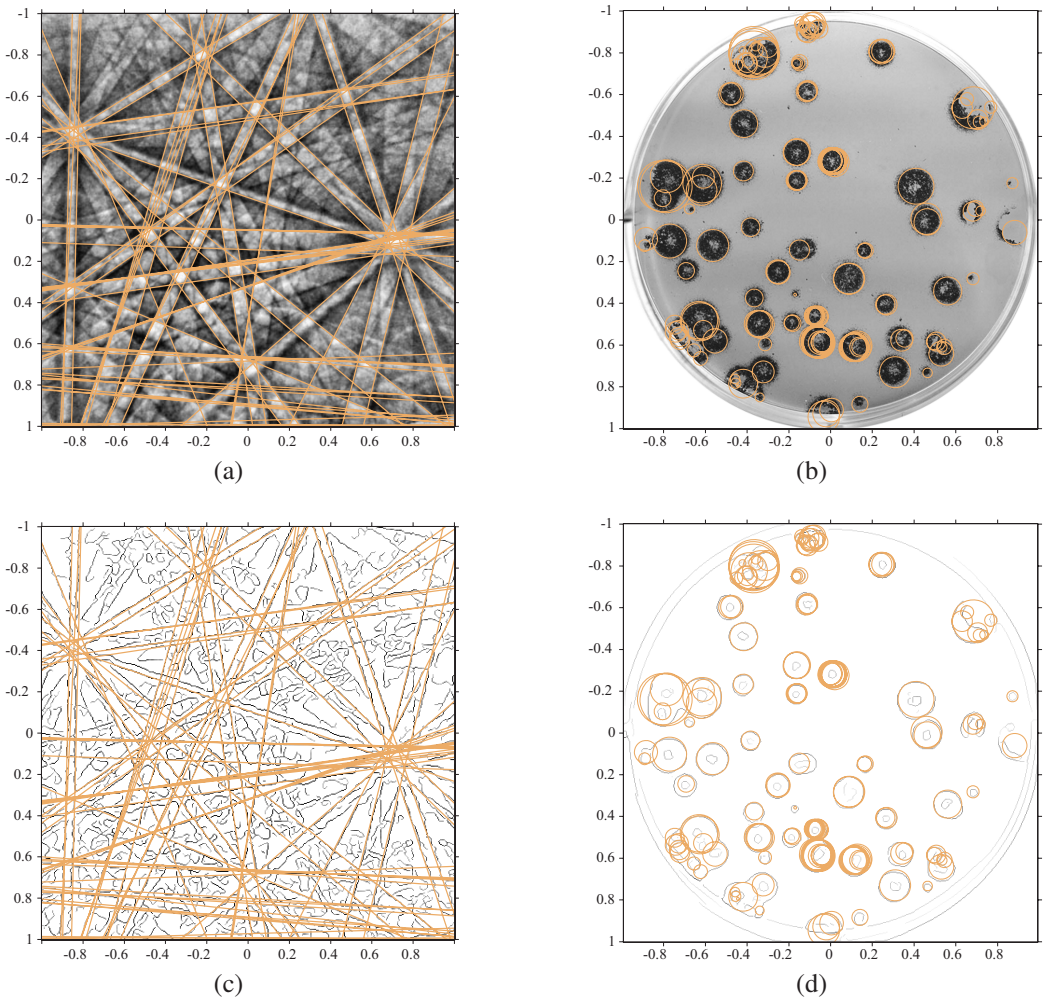


Figure E.1: Detecção automática de alinhamentos em conjuntos de dados reais. (a) Imagem produzidas por difração de elétrons retroespelhados (445×445 pixels) obtida de uma partícula de wulfenita ($PbMoO_4$). A detecção de linhas retas é chave para a identificação da fase de partículas cristalinas. (b) Imagem em tons de cinza (529×534 pixels) de infecção por A/Memphis/14/96-M (H1N1) em células MDCK-SIAT1. A detecção de círculos é importante para o processo automático de contagem em ensaios clonogênicos. A abordagem proposta foi usada, sem qualquer alteração, na identificação automática das linhas retas e dos círculos exibidos em (a) e (b) a partir da informação de borda exibida em (c) e (d), respectivamente.

3. É independente da dimensionalidade e da métrica do espaço onde os dados residem; e
4. Garante o uso da menor quantidade de parâmetros na representação do tipo de subespaço requerido. Logo, quando aplicada como um detector de formas geométricas, a técnica proposta sempre representa a forma requerida pela maneira mais compacta.

Eu formulei o detector de subespaços utilizando Álgebra Geométrica (GA, do inglês *Geometric Algebra*). GA é um sistema matemático poderoso que engloba muitos conceitos (*e.g.,* números complexos, álgebra de quaternions, álgebra de Grassmann-Cayley e coordenadas de Plücker) sob o mesmo formalismo (DORST; FONTIJINE; MANN, 2007;

PERWASS, 2009). GA é baseada principalmente na Álgebra de Clifford (LOUNESTO, 2001), mas com forte ênfase na interpretação geométrica. Em GA, subespaços são tratados como primitivas para computação. Deste modo, GA é uma ferramenta matemática apropriada para a modelagem do problema de detecção de subespaço. Além disso, GA tem se mostrado capaz de representar muitos tipos de geometria. Como exemplos de modelos de geometria (MOGs, do inglês *models of geometry*) codificados por GA, podem ser citados o espaço Euclidiano, Projetivo (HESTENES; ZIEGLER, 1991), Esférico (HESTENES, 1987, 2001), Hiperbólico (LI, 1997) e Cônico (PERWASS; FÖRSTNER, 2006). Esses MOGs provêem aplicações práticas para a abordagem proposta como um detector de formas geométricas emergentes em conjuntos de dados como, mas não limitados a, imagens bidimensionais, imagens volumétricas e nuvens de pontos.

E.1 Abordagens Convencionais

Muitas das técnicas para detecção de alinhamentos de dados são derivadas do paradigma da Transforma de Hough (HT, do inglês *Hough Transform* (HOUGH, 1959, 1962)), da RANSAC (do inglês *RANdom SAmple Consensus* (FISCHLER; BOLLES, 1981)) ou de Votação de Tensores (TV, do inglês *Tensor Voting* (MEDIONI; LEE; TANG, 2000)). Uma discussão sobre esses paradigmas e algumas de suas variações é apresentada no Capítulo 2.

A HT executa a detecção de um tipo requerido de forma geométrica por meio do mapeamento de cada elemento do conjunto de dados de entrada para pontos (em um espaço de parâmetros) representando as formas que passam sobre o elemento informado. Com base neste mecanismo, o problema de detecção de formas geométricas pode ser convertido no problema mais simples de identificação de picos de votos em uma grade de acumuladores que representa o espaço de parâmetros discretizado.

A RANSAC é uma técnica não determinística onde as instâncias mais significativas de um tipo de alinhamento de dados são identificadas por meio de ajustes sucessivos de instâncias candidatas sobre amostras aleatórias dos dados de entrada. Após certo número de tentativas, a instância candidata que aproxima a maior quantidade de elementos de entrada é indicada como sendo a instância detectada.

O paradigma de TV recupera as superfícies, curvas e junções mais salientes em um conjunto de dados através da codificação da informação associada a cada elemento do conjunto na forma de um tensor. A informação codificada é propagada por uma vizinhança por meio de campos de votação de tensores. Após duas rodadas de votação e refinamento, as características salientes (com qualquer dimensionalidade) são identificadas como picos em mapas de votos densos calculados (ao mesmo tempo) para cada tipo de característica saliente.

Para utilizar as abordagens da HT ou RANSAC, é preciso assumir um modelo matemático para o tipo requerido de alinhamento, definido com respeito ao tipo esperado de dado de entrada. Então, tradicionalmente, as variações da HT e RANSAC têm sido projetadas para detectar tipos específicos de estruturas em um tipo específico de dado. Este nível de especialização impede o desenvolvimento de técnicas que possam ser aplicadas de maneira geral, e de otimizações que possam ser empregadas em todas as formulações. Apesar de existirem generalizações parciais da HT e RANSAC projetadas para algumas classes de formas geométricas analíticas (BALLARD, 1981; SCHNABEL; WAHL; KLEIN, 2007; ACHTERT et al., 2008) e HTs para formas não analíticas (BALLARD, 1981; WANG; REEVES, 1990), tais abordagens são ainda restritivas quanto ao

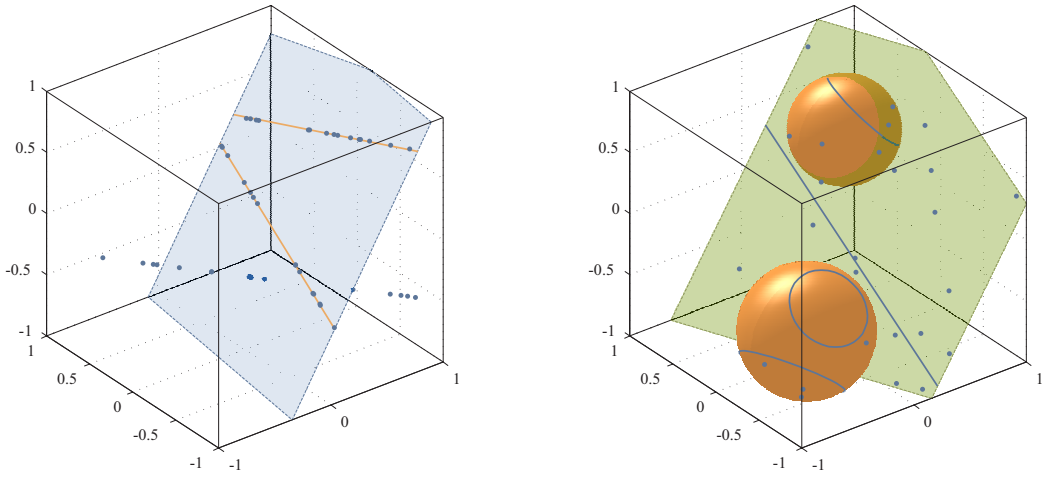


Figure E.2: Detecção automática de alinhamentos de dados em conjuntos sintéticos de dados heterogêneos. (a) Detecção das linhas retas que melhor se ajustam aos dados de entrada compostos por 45 pontos e 1 plano. Neste exemplo, nós estamos interessados na detecção das retas que estejam sobre o plano e que também se ajustem a subconjuntos de pontos. (b) Detecção concorrente de planos e esferas a partir de uma única aplicação da técnica proposta. O conjunto de dados de entrada é composto por 43 pontos, 1 linha reta e 3 círculos.

tipo de dado de entrada assumido ou ao tipo de estrutura a ser detectada.

A técnica de TV, por outro lado, segue uma definição generalizada. Esta abordagem é projetada para a detecção de alinhamentos com dimensionalidades arbitrárias em conjuntos de dados heterogêneos. TV, entretanto, retorna todas as estruturas possíveis ao mesmo tempo. Tal comportamento impede a detecção eficiente de tipos de alinhamentos pré-definidos, pois a técnica iria requer uma etapa subseqüente de filtragem das detecções.

E.2 Idéia Central

Eu proponho a generalização das técnicas baseadas em votação para a detecção de alinhamentos de dados que possam ser caracterizados por um subespaço em qualquer MOG. A idéia central defendida nesta pesquisa é apresentada abaixo:

> *É possível definir uma parametrização para subespaços lineares que seja independente da dimensionalidade do subespaço requerido, do tipo de dado de entrada e da métrica do espaço como um todo. Além disso, desde que subespaços lineares de uma dada dimensionalidade podem ser interpretados como algum tipo de alinhamento de dados em um modelo de geometria assumido, é possível utilizar tal parametrização no desenvolvimento de uma técnica, baseada em votação, para a detecção automática de alinhamentos em conjuntos de dados multidimensionais.*

Para explorar o domínio associado ao esquema de detecção proposto, é preciso que três problemas sejam resolvidos. O primeiro é definir uma parametrização para subespaços lineares de uma dada dimensionalidade em um espaço n-dimensional. O Capítulo 4 apresenta uma derivação onde subespaços p-dimensionais, para $0 \leq p \leq n$, são

parametrizados por $p\,(n-p)$ operações de rotação aplicadas a um subespaço canônico com a mesma dimensionalidade. A parametrização é definida de tal modo que ela é independente da interpretação do subespaço no contexto atual (*i.e.,* a interpretação geométrica em algum MOG).

O segundo problema é definir a relação de dualidade entre os dados de entrada e os parâmetros que caracterizam os subespaços requeridos. Esta relação é utilizada no desenvolvimento de um procedimento que mapeie, para o espaço de parâmetros, elementos com dimensionalidades arbitrárias vindos do conjunto de dados de entrada. Tal mapeamento é o componente central de um esquema de votação, onde uma representação discreta do espaço de parâmetros é usada a fim de identificar quais são os subespaços p-dimensionais que melhor se ajustam às entradas. A questão da votação é explorada no Capítulo 5, onde procedimentos gerais de mapeamento e votação são definidos para subespaços r-dimensionais ($0 \leq r \leq n$) de entrada, de tal modo que o conhecimento prévio da interpretação geométrica dos dados não é requerido. Os procedimentos de mapeamento e votação são estendidos no Capítulo 6 para o tratamento de subespaços de entrada com distribuição Gaussiana de incerteza. Neste caso, a incerteza intrínseca a dados experimentais é levada em consideração durante a distribuição de votos no mapa discreto.

O terceiro problema a ser resolvido é como encontrar os subespaços mais relevantes a partir da representação discreta do espaço de parâmetros, após o procedimento de votação ter sido executado. Este tópico é discutido no Capítulo 7, onde é proposta uma técnica para detecção de picos em histogramas multidimensionais.

E.3 Breve Descrição da Técnica Geral para Detecção de Subespaços

Um subespaço p-dimensional $\mathbf{B}_{\langle p \rangle}$ inserido em um espaço n-dimensional pode ser caracterizado por um conjunto de $m = p\,(n-p)$ rotações aplicadas a um subespaço canônico ($\mathbf{E}_{\langle p \rangle}$) utilizado como referência. Mais precisamente, na notação de GA:

$$\mathbf{B}_{\langle p \rangle} = \boldsymbol{T}\,\mathbf{E}_{\langle p \rangle} \,/\, \boldsymbol{T}, \tag{E.1}$$

onde $\boldsymbol{T}$ é o rotor (rotores são definidos no Capítulo 3) que codifica uma seqüência de m operações de rotação. A derivação de $\boldsymbol{T}$ é apresentada no Capítulo 4. Na Equação E.1, a transformação codificada por $\boldsymbol{T}$ é aplicada sobre $\mathbf{E}_{\langle p \rangle}$ com o intuito de obter $\mathbf{B}_{\langle p \rangle}$ como resultado. Os m ângulos de rotação são os parâmetros que caracterizam $\mathbf{B}_{\langle p \rangle}$ e os valores de p e n dependem do tipo requerido de alinhamento de dados. Por exemplo, ao assumir o MOG homogêneo (ou projetivo) para a detecção de retas em imagens (Figura E.1a), $n = 2 + 1 = 3$ e $p = 2$, levando a $m = 2\,(3-2) = 2$.

Os m ângulos de rotação relacionados com a seqüência de rotações na Equação E.1 definem um espaço de parâmetros para subespaços p-dimensionais. O detector proposto utiliza este espaço de parâmetros. A aplicação da técnica consiste de três passos:

1. Criar uma grade de acumuladores como sendo a representação discreta do espaço de parâmetros;
2. Executar o procedimento de votação, onde o conjunto de dados de entrada é mapeado para a grade de acumuladores; e
3. Buscar pelos picos de votos na grade de acumuladores, pois eles correspondem aos subespaços p-dimensionais que melhor se ajustam aos dados de entrada.

A Tabela E.1 resume os três passos. O passo (1) define um espaço de parâmetros ($\mathbb{P}^m$)

para os m graus de liberdade:

$$\mathbb{P}^m = \{(\theta^1, \theta^2, \cdots, \theta^m) \mid \theta^t \in [-\pi/2,\pi/2)\}, \tag{E.2}$$

onde cada vetor de parâmetros $(\theta^1, \theta^2, \cdots, \theta^m) \in \mathbb{P}^m$ caracteriza uma instância de um subespaço p-dimensional em um espaço vetorial n-dimensional. Os valores de p e n estão relacionados ao MOG onde os dados foram codificados e ao tipo de alinhamento (também representado neste MOG) que se deseja detectar. Na prática, é preciso discretizar $\mathbb{P}^m$, para o qual uma grade de acumuladores é criada para receber "votos" e inicializada com zeros.

O passo (2) mapeia os dados de entrada para o espaço de parâmetros. Essencialmente, o mapeamento pega cada subespaço r-dimensional $\mathbf{X}_{\langle r \rangle}$ no conjunto de dados de entrada e identifica os parâmetros (coordenadas em $\mathbb{P}^m$) de todos os subespaços p-dimensionais relacionados a ele. Quando $r \leq p$, o mapeamento identifica em $\mathbb{P}^m$ todos os subespaços p-dimensionais que contém $\mathbf{X}_{\langle r \rangle}$ (*e.g.,* as linhas contendo pontos dados como entrada na Figura E.2a). Se $r \geq p$, o procedimento identifica em $\mathbb{P}^m$ todos os subespaços p-dimensionais contidos em $\mathbf{X}_{\langle r \rangle}$ (*e.g.,* as linhas no plano dado como entrada na Figura E.2a).

Após o procedimento de votação ter sido executado para todo $\mathbf{X}_{\langle r \rangle}$, o número de votos depositado em cada célula do acumulador define a importância dos subespaços representados pelas células com relação aos dados de entrada. Logo, as células que receberam

Table E.1: Os passos da técnica proposta para detecção de subespaços. Como entrada é requerido um conjunto $\mathcal{X}$ de subespaços (*i.e.,* os dados de entrada codificados em um MOG), a dimensionalidade p dos subespaços interpretados como os tipos de alinhamento requeridos e a dimensionalidade n do espaço vetorial como um todo. O algoritmo retorna os subespaços p-dimensionais que melhor se ajustam ao conjunto de entrada $\mathcal{X}$.

Passo	**Descrição**
1. Inicialização	Utilize a função que modela subespaços p-dimensionais (Equação E.1) na definição do espaço de parâmetros $\mathbb{P}^m$ (Equação E.2). Então, crie (e inicialize com zeros) uma grade m-dimensional de acumuladores como sendo a representação discreta de $\mathbb{P}^m$.
2. Votação	Mapeie cada subespace de entrada $\mathbf{X}_{\langle r \rangle} \in \mathcal{X}$ para $\mathbb{P}^m$ utilizando o procedimento descrito no Capítulo 5 ou no Capítulo 6. Durante o mapeamento, alguns parâmetros assumem um único valor, enquanto que outros assumem todos os valores no intervalo $[-\pi/2, \pi/2)$. Entretanto, por conta da natureza discreta da grade de acumuladores, esses parâmetros precisam assumir apenas um subconjunto de valores discretos em $[-\pi/2, \pi/2)$. Como conseqüência, o mapeamento retorna um conjunto finito de pontos em $\mathbb{P}^m$. Esses pontos são usados para endereçar células na grade de acumuladores. A votação é realizada pelo incremento dessas células conforme a importância ω de $\mathbf{X}_{\langle r \rangle}$.
3. Detecção de Picos	Identifique as células que correspondem aos máximos locais na grade de acumuladores. Utilize as coordenadas dessas células (*i.e.,* vetores de parâmetros) para obter os subespaços p-dimensionais mais significativos.

mais votos representam os subespaços p-dimensionais detectados. O passo final da abordagem busca pelos máximos locais na grade de acumuladores. Os vetores de parâmetros associados a essas células são utilizados na Equação E.1 a fim de se obter os subespaços detectados. Isso é feito pela aplicação da seqüência de rotações especificadas pelo endereço das células $(\theta^1, \theta^2, \cdots, \theta^m)$ sobre um subespaço canônico $\mathbf{E}_{\langle p \rangle}$.

E.4 Resultados

Esta dissertação apresenta resultados originais que incluem:

- Uma abordagem geral para a detecção de subespaços em conjuntos de dados multidimensionais não ordenados;
- Um esquema para a parametrização de subespaços baseado na rotação de um subespaço canônico de mesma dimensionalidade;
- Um algoritmo que enumera todas as instâncias de subespaços, com uma dada dimensionalidade p, que contenha ou que esteja contido em um subespaço de dimensionalidade arbitrária informado como entrada;
- Um procedimento que mapeia subespaços com distribuição Gaussiana de incerteza para o espaço de parâmetros que caracteriza subespaços p-dimensionais;
- Um conjunto de evidências experimentais mostrando que a cobertura afim da Grassmaniana (HARRIS, 1992) pode ser usada como um espaço auxiliar, onde a incerteza de estruturas geométricas como linhas retas, planos, círculos, esferas e seções cônicas, entre outras, é tratada de maneira unificada; e
- Um algoritmo para identificar máximos locais em um histograma multidimensional.

Além dessas contribuições, as seguintes afirmações são demonstradas com relação à abordagem proposta:

- Ela é uma generalização das HTs para formas analíticas que podem ser representadas por subespaços lineares;
- Ela permite a detecção dos subespaços que melhor se ajustam a conjuntos de dados compostos por subespaços com diferentes dimensionalidades e diferentes interpretações geométricas (*e.g.,* a detecção de linhas retas que melhor se ajustam a pontos e planos – Figura E.2a);
- Ela permite a detecção concorrente de subespaços com interpretações geométricas diferentes (*e.g.,* planos e esferas – Figura E.2b) mas com a mesma dimensionalidade em um dado MOG;
- Os subespaços p-dimensionais requeridos são representados com a menor quantidade possível de parâmetros, levando à parametrização mais compacta de formas analíticas (*e.g.,* linhas retas, círculos e seções cônicas no plano são parametrizadas com dois, três e quatro parâmetros, respectivamente);
- A detecção pode ser direcionada a um tipo específico de alinhamento com a simples troca do MOG assumido na codificação dos dados, enquanto que a formulação do esquema de detecção permanece inalterada; e
- Uma aproximação do diagrama de Voronoi de d-ésima ordem (VORONOI, 1908) para um conjunto de pontos em $\mathbb{R}^d$ pode ser obtida como um subproduto da detecção de subespaços geometricamente interpretados como círculos, esferas e seus equivalentes de dimensionalidade mais alta.

É importante enfatizar que a abordagem proposta não está restrita à detecção de formas geométricas. Ela pode ser aplicada sobre qualquer domínio no qual um problema possa ser caracterizado por um problema de detecção de subespaços. Por exemplo, o problema de agrupamento de subespaços em aplicações de mineração de dados, onde o objetivo é encontrar os subespaços que acomodem a maior quantidade de objetos da base de dados. Além disso, como uma solução fechada, a mesma implementação da abordagem proposta e de suas otimizações naturalmente generaliza para todos os casos de detecção.

E.5 Demonstração e Validação das Técnicas

As técnicas propostas nesta tese são demonstradas pela implementação dos algoritmos descritos. As implementações são validadas pela aplicação do esquema de detecção de subespaços sobre conjuntos de dados reais (Figura E.1) e sintéticos (Figura E.2). A abordagem proposta apresenta uma faixa de aplicações bastante ampla como uma ferramenta para detecção de padrões. Porém, Para efeitos de ilustração, os exemplos exibidos nesta tese são restritos ao problema importante de detecção de formas geométricas analíticas em espaços com dimensionalidade arbitrária.