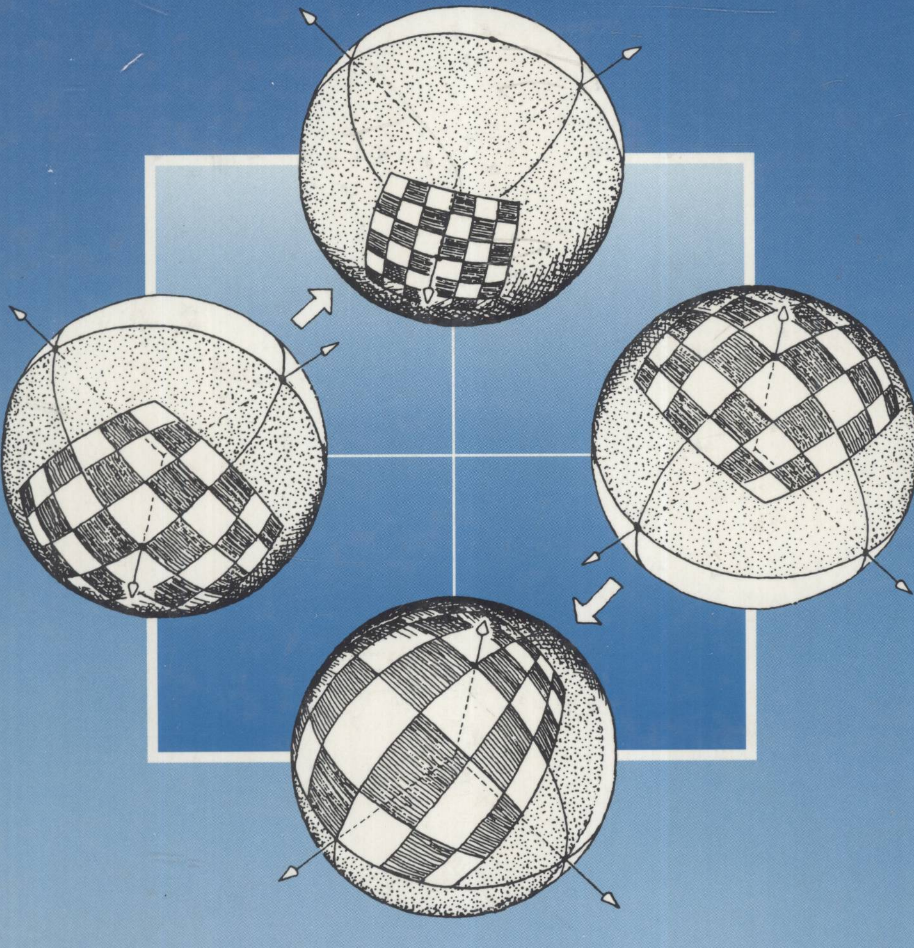
# Oriented Projective Geometry

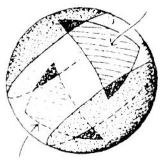
## A Framework for Geometric Computations



JORGE STOLFI

# ORIENTED PROJECTIVE GEOMETRY

## *A Framework for Geometric Computations*



JORGE STOLFI

Digital Systems Research Center
Palo Alto, California



ACADEMIC PRESS, INC.
*Harcourt Brace Jovanovich, Publishers*
Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper. ♾



ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101

*United Kingdom Edition published by*
ACADEMIC PRESS LIMITED
24–28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data
Stolfi, Jorge.
Oriented projective geometry : a framework for geometric computations / Jorge Stolfi.
p. cm.
Includes bibliographical references and index.
ISBN 0-12-672025-8 (alk. paper)
1.Geometry. Projective — Data processing. I. Title.
QA471.S88 1991
516′.5—dc20 91-16219
CIP

Printed in the United States of America
91 92 93 94 9 8 7 6 5 4 3 2 1

# Introduction

Programmers who use homogeneous coordinates for geometric computations are implicitly — and often unknowingly — working in the so-called *projective space*, a strange and wonderful world which only superficially resembles the Euclidean space we all know and love. One difference is that certain concepts that are fundamental to geometric computing, such as segments, triangle orientation, and convexity, cannot be consistently defined in the projective world.

*Oriented projective geometry* is an alternative geometric model that combines the elegance and efficiency of projective geometry with the consistent handling of oriented lines and planes, signed angles, segments, convex sets, and many other concepts that the classical theory does not support. In this monograph I advance the thesis that *oriented* projective space — especially in its analytic form, based on *signed* homogeneous coordinates — is a better framework for geometric computations than their classical counterparts.

The differences between the classical and oriented models are largely confined to the mathematical formalism and its interpretation. Computationally, the differences are minimal; most geometric algorithms that use homogeneous coordinates can be easily converted to the oriented model with negligible effect on their performance. For many algorithms, the required changes are largely a matter of paying attention to the order of operands and to the signs of coordinates.

It is not the aim of this monograph to push the remote frontiers of mathematics or computer science. Theoreticians will not find here any deep theorems, intricate algorithms, or sophisticated data structures. Expert geometers will notice that oriented projective geometry is just anothr name for spherical (or double elliptic) geometry, which to them is an old and well-explored subject.

On the other hand, graphics programmers may be surprised to learn that the curved surface of the sphere is an excellent model for computations dealing with straight lines on the flat Euclidean plane. The aim of this monograph is to point out the value of this model for practical computing, and to develop it into a rich, consistent, and effective tool that those programmers can use in their everyday work. In keeping with this goal, I have strived to keep formal derivations and mathematical jargon to a minimum, and (at the risk of being tedious) to illustrate many general definitions and theorems with explicit examples in one, two, and three dimensions.

Here is a brief outline of this book. Chapter 1 gives a quick overview of classical and oriented projective geometry on the plane, and discusses their advantages and disadvantages as computational models. Chapters 2 through 7 define the canonical oriented projective spaces of arbitrary dimension, the operations of join and meet, and the concept of relative orientation, and study their properties. Chapter 8 defines projective maps, the space transformations that preserve incidence and orientation; these maps are used in chapter 9 to define abstract oriented projective spaces. Chapter 10 introduces the valuable notion of projective duality. Chapters 11, 12, and 13 deal with additional concepts related to projective maps, namely projective functions, projective frames, relative coordinates, and cross-ratio. Chapter 14 tells about convexity in oriented projective spaces. Chapters 15, 16, and 17 show how the affine, Euclidean, and linear vector spaces can be emulated with the oriented projective space. Finally, chapters 18 through 20 discuss the computer representation and manipulation of lines, planes, and other subspaces.

This monograph is a slightly edited and revised version of my Ph.D. dissertation, *Primitives for Computational Geometry*, submitted to Stanford University in May 1988, and printed under the same title as Technical Report 36 of the DEC Systems Research Center in January 1989. The present edition incorporates innumerable small corrections and improvements over these earlier versions.

**Acknowledgements:** I would like to express here my gratitude, first of all, to my advisor Leo Guibas, who helped me, prodded me, and supported me in more ways that I could possibly list here; and to my boss Bob Taylor, who gave me constant encouragement and the opportunity to experience the unique research environment that he created at Xerox PARC and DEC SRC.

I learned a great many things about this subject from discussions with my colleagues, especially with Lyle Ramshaw, John Hershberger, Mike Lowry, and Yuen Yu. I am grateful to Professors Ernst Mayr and Andrew Yao for their careful reading of my dissertation. From Mary-Claire van Leunen and Cynthia Hibbard I received a good deal of advice on the art and science of writing, which immensely improved my syntax and style — from truly atrocious to, I hope, merely dreadful.

I am also indebted to Ken Brooks, Marc Brown, Bill Kalsow and Lyle Ramshaw for taking time out of their own research to maintain the software on which I depended; and to the Digital Equipment Corporation, the Xerox Corporation, the University of São Paulo, and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) of Brazil for their generous financial support.

Finally, I wish to give my deepest thanks to all my dear friends, here and abroad, who never grew tired of asking when would I be finished; and to my wife Rumiko, who deserves more credit for this work than all of the above put together.

Jorge Stolfi
March 1991

TABLE OF CONTENTS

Chapter 0. Introduction . . . . . . . . . . 1

Chapter 1. Projective geometry . . . . . . . . . . 3
1.1. The classic projective plane . . . . . . . . . . 3
1.2. Advantages of projective geometry . . . . . . . . . . 6
1.3. Drawbacks of classical projective geometry . . . . . . . . . . 8
1.4. Oriented projective geometry . . . . . . . . . . 10
1.5. Related work . . . . . . . . . . 11

Chapter 2. Oriented projective spaces . . . . . . . . . . 13
2.1. Models of two-sided space . . . . . . . . . . 13
2.2. Central projection . . . . . . . . . . 16

Chapter 3. Flats . . . . . . . . . . 19
3.1. Definition . . . . . . . . . . 19
3.2. Points . . . . . . . . . . 19
3.3. Lines . . . . . . . . . . 20
3.4. Planes . . . . . . . . . . 23
3.5. Three-spaces . . . . . . . . . . 26
3.6. Ranks . . . . . . . . . . 27
3.7. Incidence and independence . . . . . . . . . . 27

Chapter 4. Simplices and orientation . . . . . . . . . . 29
4.1. Simplices . . . . . . . . . . 29
4.2. Simplex equivalence . . . . . . . . . . 30
4.3. Point location relative to a simplex . . . . . . . . . . 34
4.4. The vector space model . . . . . . . . . . 37

Chapter 5. The join operation . . . . . . . . . . 39
5.1. The join of two points . . . . . . . . . . 39
5.2. The join of a point and a line . . . . . . . . . . 41
5.3. The join of two arbitrary flats . . . . . . . . . . 42
5.4. Properties of join . . . . . . . . . . 43
5.5. Null objects . . . . . . . . . . 44
5.6. Complementary flats . . . . . . . . . . 45

Chapter 6. The meet operation . . . . . . . . . . 47
6.1. The meeting point of two lines . . . . . . . . . . 47
6.2. The general meet operation . . . . . . . . . . 49
6.3. Meet in three dimensions . . . . . . . . . . 51
6.4. Properties of meet . . . . . . . . . . 53

Chapter 7. Relative orientation . . . . . . . . . . . . . . . . . . . . 59
7.1. The two sides of a line . . . . . . . . . . . . . . . . . . . . 59
7.2. Relative position of arbitrary flats . . . . . . . . . . . . . . . 60
7.3. The separation theorem . . . . . . . . . . . . . . . . . . . . 64
7.4. The coefficients of a hyperplane . . . . . . . . . . . . . . . . 66

Chapter 8. Projective maps . . . . . . . . . . . . . . . . . . . . 67
8.1. Formal definition . . . . . . . . . . . . . . . . . . . . . . 68
8.2. Examples . . . . . . . . . . . . . . . . . . . . . . . . . . 70
8.3. Properties of projective maps . . . . . . . . . . . . . . . . . 72
8.4. The matrix of a map . . . . . . . . . . . . . . . . . . . . . 73

Chapter 9. General two-sided spaces . . . . . . . . . . . . . . . . 77
9.1. Formal definition . . . . . . . . . . . . . . . . . . . . . . 77
9.2. Subspaces . . . . . . . . . . . . . . . . . . . . . . . . . . 78

Chapter 10. Duality . . . . . . . . . . . . . . . . . . . . . . . . 83
10.1. Duomorphisms . . . . . . . . . . . . . . . . . . . . . . . . 83
10.2. The polar complement . . . . . . . . . . . . . . . . . . . . 85
10.3. Polar complements as duomorphisms . . . . . . . . . . . . . 89
10.4. Relative polar complements . . . . . . . . . . . . . . . . . 91
10.5. General duomorphisms . . . . . . . . . . . . . . . . . . . . 92
10.6. The power of duality . . . . . . . . . . . . . . . . . . . . 93

Chapter 11. Generalized projective maps . . . . . . . . . . . . . . 95
11.1. Projective functions . . . . . . . . . . . . . . . . . . . . . 95
11.2. Computer representation . . . . . . . . . . . . . . . . . . . 101

Chapter 12. Projective frames . . . . . . . . . . . . . . . . . . . 107
12.1. Nature of projective frames . . . . . . . . . . . . . . . . . 107
12.2. Classification of frames . . . . . . . . . . . . . . . . . . . 110
12.3. Standard frames . . . . . . . . . . . . . . . . . . . . . . . 113
12.4. Coordinates relative to a frame . . . . . . . . . . . . . . . 120

Chapter 13. Cross ratio . . . . . . . . . . . . . . . . . . . . . . 123
13.1. Cross ratio in unoriented geometry . . . . . . . . . . . . . 123
13.2. Cross ratio in the oriented framework . . . . . . . . . . . . 126

Chapter 14. Convexity . . . . . . . . . . . . . . . . . . . . . . . 131
14.1. Convexity in classical projective space . . . . . . . . . . . 131
14.2. Convexity in oriented projective spaces . . . . . . . . . . . 132
14.3. Properties of convex sets . . . . . . . . . . . . . . . . . . 134
14.4. The half-space property . . . . . . . . . . . . . . . . . . . 138
14.5. The convex hull . . . . . . . . . . . . . . . . . . . . . . . 141
14.6. Convexity and duality . . . . . . . . . . . . . . . . . . . . 143

Chapter 15. Affine geometry . . . 151
15.1. The Cartesian connection . . . 151
15.2. Two-sided affine spaces . . . 153
Chapter 16. Vector algebra . . . 167
16.1. Two-sided vector spaces . . . 167
16.2. Translations . . . 168
16.3. Vector algebra . . . 168
16.4. The two-sided real line . . . 171
16.5. Linear maps . . . 171
Chapter 17. Euclidean geometry on the two-sided plane . . . 173
17.1. Perpendicularity . . . 173
17.2. Two-sided Euclidean spaces . . . 177
17.3. Euclidean maps . . . 178
17.4. Length and distance . . . 183
17.5. Angular measure and congruence . . . 187
17.6. Non-Euclidean geometries . . . 189
Chapter 18. Representing flats by simplices . . . 191
18.1. The simplex representation . . . 191
18.2. The dual simplex representation . . . 193
18.3. The reduced simplex representation . . . 195
Chapter 19. Plücker coordinates . . . 197
19.2. The canonical embedding . . . 202
19.3. Plücker coefficients . . . 203
19.4. Storage efficiency . . . 204
19.5. The Grassmann manifolds . . . 204
Chapter 20. Formulas for Plücker coordinates . . . 207
20.1. Algebraic formulas . . . 207
20.2. Formulas for computers . . . 212
20.3. Projective maps in Plücker coordinates . . . 217
20.4. Directions and parallelism . . . 221
References . . . 223
List of symbols . . . 225
Index . . . 227

# Introduction

Programmers who use homogeneous coordinates for geometric computations are implicitly — and often unknowingly — working in the so-called *projective space*, a strange and wonderful world which only superficially resembles the Euclidean space we all know and love. One difference is that certain concepts that are fundamental to geometric computing, such as segments, triangle orientation, and convexity, cannot be consistently defined in the projective world.

*Oriented projective geometry* is an alternative geometric model that combines the elegance and efficiency of projective geometry with the consistent handling of oriented lines and planes, signed angles, segments, convex sets, and many other concepts that the classical theory does not support. In this monograph I advance the thesis that *oriented* projective space — especially in its analytic form, based on *signed* homogeneous coordinates — is a better framework for geometric computations than their classical counterparts.

The differences between the classical and oriented models are largely confined to the mathematical formalism and its interpretation. Computationally, the differences are minimal; most geometric algorithms that use homogeneous coordinates can be easily converted to the oriented model with negligible effect on their performance. For many algorithms, the required changes are largely a matter of paying attention to the order of operands and to the signs of coordinates.

It is not the aim of this monograph to push the remote frontiers of mathematics or computer science. Theoreticians will not find here any deep theorems, intricate algorithms, or sophisticated data structures. Expert geometers will notice that oriented projective geometry is just anothr name for spherical (or double elliptic) geometry, which to them is an old and well-explored subject.

On the other hand, graphics programmers may be surprised to learn that the curved surface of the sphere is an excellent model for computations dealing with straight lines on the flat Euclidean plane. The aim of this monograph is to point out the value of this model for practical computing, and to develop it into a rich, consistent, and effective tool that those programmers can use in their everyday work. In keeping with this goal, I have strived to keep formal derivations and mathematical jargon to a minimum, and (at the risk of being tedious) to illustrate many general definitions and theorems with explicit examples in one, two, and three dimensions.

Here is a brief outline of this book. Chapter 1 gives a quick overview of classical and oriented projective geometry on the plane, and discusses their advantages and disadvantages as computational models. Chapters 2 through 7 define the canonical oriented projective spaces of arbitrary dimension, the operations of join and meet, and the concept of relative orientation, and study their properties. Chapter 8 defines projective maps, the space transformations that preserve incidence and orientation; these maps are used in chapter 9 to define abstract oriented projective spaces. Chapter 10 introduces the valuable notion of projective duality. Chapters 11, 12, and 13 deal with additional concepts related to projective maps, namely projective functions, projective frames, relative coordinates, and cross-ratio. Chapter 14 tells about convexity in oriented projective spaces. Chapters 15, 16, and 17 show how the affine, Euclidean, and linear vector spaces can be emulated with the oriented projective space. Finally, chapters 18 through 20 discuss the computer representation and manipulation of lines, planes, and other subspaces.

This monograph is a slightly edited and revised version of my Ph.D. dissertation, *Primitives for Computational Geometry*, submitted to Stanford University in May 1988, and printed under the same title as Technical Report 36 of the DEC Systems Research Center in January 1989. The present edition incorporates innumerable small corrections and improvements over these earlier versions.

**Acknowledgements:** I would like to express here my gratitude, first of all, to my advisor Leo Guibas, who helped me, prodded me, and supported me in more ways that I could possibly list here; and to my boss Bob Taylor, who gave me constant encouragement and the opportunity to experience the unique research environment that he created at Xerox PARC and DEC SRC.

I learned a great many things about this subject from discussions with my colleagues, especially with Lyle Ramshaw, John Hershberger, Mike Lowry, and Yuen Yu. I am grateful to Professors Ernst Mayr and Andrew Yao for their careful reading of my dissertation. From Mary-Claire van Leunen and Cynthia Hibbard I received a good deal of advice on the art and science of writing, which immensely improved my syntax and style — from truly atrocious to, I hope, merely dreadful.

I am also indebted to Ken Brooks, Marc Brown, Bill Kalsow and Lyle Ramshaw for taking time out of their own research to maintain the software on which I depended; and to the Digital Equipment Corporation, the Xerox Corporation, the University of São Paulo, and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) of Brazil for their generous financial support.

Finally, I wish to give my deepest thanks to all my dear friends, here and abroad, who never grew tired of asking when would I be finished; and to my wife Rumiko, who deserves more credit for this work than all of the above put together.

Jorge Stolfi
March 1991

# Chapter 1
# Projective geometry

The bulk of this chapter is a quick overview of the standard (unsigned) homogeneous coordinates for the plane, and the classical (unoriented) projective geometry which they implicitly define. In order to motivate what follows, I will discuss at some length the advantages and disadvantages of homogeneous coordinates as a computational model, compared to ordinary Cartesian coordinates. The chapter concludes with a quick overview of *oriented* projective geometry, the alternative computational model which is the subject of this book, and which I define formally in the following chapters.

The description of projective geometry given below below is necessarily sketchy, and does not even begin to make justice to the richness and elegance of the subject. Mathematically inclined readers who wish to know more are urged to start from any basic textbook on the subject, such as Coxeter's [6], and follow the leads from there. Readers interested in practical applications of projective geometry to computer graphics are advised to read the the book by Penna and Patterson [16].

## 1. The classic projective plane

The projective plane can be defined either by means of a "concrete" model, borrowing concepts from linear algebra or Euclidean geometry [15], or as an abstract structure satisfying certain axioms [4, 6].

Definitions that follow the axiomatic approach have the advantage of being concise and elegant, but unfortunately they cannot be generalized easily to spaces of arbitrary dimension. Moreover, the axiomatic approach seems better suited to formalizing intuitive knowledge already acquired, than at developing and teaching such knowledge. Therefore, considering the aims of this monograph, I have chosen to avoid the axiomatic approach, and to base all definitions on four concrete models of projective space: the *straight*, *spherical*, *analytic*, and *vector space* models, which are described below.

## 1.1. The straight model

The *straight model* of the projective plane $\mathbf{P}_2$ consists of the real plane $\mathbf{R}^2$, augmented by a *line at infinity* $\Omega$, and by an *infinity point* $d\infty$ for each pair of opposite directions $\{d, -d\}$. The point $d\infty = (-d)\infty$ is by definition on the line $\Omega$ and also on every line that is parallel to the direction $d$. See figure 1.

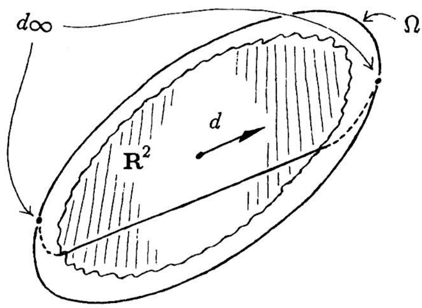



Figure 1. The straight model of the projective plane $\mathbf{P}_2$.

## 1.2. The spherical model

The *spherical model* of $\mathbf{P}_2$ consists of the surface of a sphere, with diametrically opposite points identified. The lines of $\mathbf{P}_2$ are represented by the great circles of the sphere, again with opposite points identified. See figure 2.

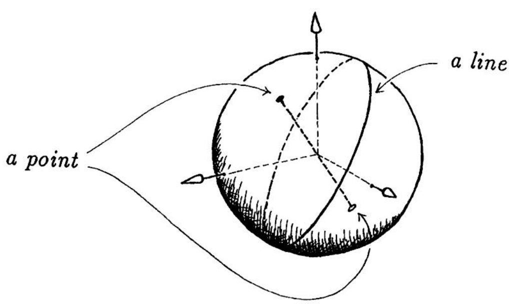



Figure 2. The spherical model.

The spherical model clearly shows that all lines and points are equivalent in their topological and incidence properties. The special role that $\Omega$ and the infinite points seem to play in the straight model is a mere artifact of the latter's representation.

## 1.3. The analytic model

The *analytic model* represents points of $\mathbf{P}_2$ by their *homogeneous coordinates*, and lines by their *homogeneous coefficients*. A point is by definition a non-zero triplet of real numbers $[w, x, y]$, with scalar multiples identified. In other words, $[w, x, y]$ and $[\lambda w, \lambda x, \lambda y]$ are the same point, for all $\lambda \neq 0$. A line is also represented by a non-zero real triplet $\langle W, X, Y\rangle$, which by definition is incident to all points $[w, x, y]$ such that $Ww + Xx + Yy = 0$. Note that $\langle W, X, Y\rangle$ and $\langle \lambda X, \lambda Y, \lambda Z\rangle$ are the same line for all $\lambda \neq 0$.

## 1.4. The vector space model

Geometrically, we can identify the point $[w, x, y]$ of $\mathbf{P}_2$ with the line of $\mathbf{R}^3$ passing through the origin and through the point $(w, x, y)$. The line $\langle W, X, Y\rangle$ of $\mathbf{P}_2$ then corresponds to the plane of $\mathbf{R}^3$ passing through the origin and perpendicular to the vector $(W, X, Y)$. In other words, we can identify points and lines of $\mathbf{P}_2$ with one- and two-dimensional linear subspaces of the three-dimensional vector space $\mathbf{R}^3$. In this way we get the *vector space model* of $\mathbf{P}_2$. See figure 3.

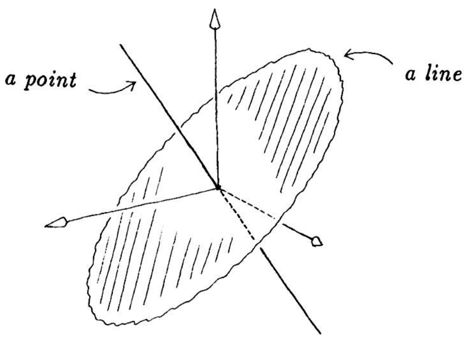



Figure 3. The vector space model of $\mathbf{P}_2$.

## 1.5. Correspondence between the models

The analytic and straight models of $\mathbf{P}_2$ are connected by the homogeneous-to-Cartesian coordinate transformation well-known to graphics programmers, which takes the homogeneous triplet $[w, x, y]$ is mapped to the point $(x/w, y/w)$ of the Cartesian plane. We can view this transformation as choosing among all equivalent homogeneous triplets a *weight-normalized* representative $(1, x/w, y/w)$ (the first coordinate $w$ being called the *weight* of the triplet). As a special case, homogeneous triplets with weight $w = 0$ are mapped to the infinity points of the straight model.

The analytic and spherical models are connected by the transformation that takes the homogeneous triplet $[w, x, y]$ to the point

$$\frac{(w, x, y)}{\sqrt{w^2 + x^2 + y^2}}$$

on the unit sphere of $\mathbf{R}^3$.

Geometrically, these mappings corresponds to *central projection* of $\mathbf{R}^3$ onto the unit sphere, or onto the plane $\pi$ tangent to the sphere at $(1, 0, 0)$. See figure 4. This projection takes a pair of diametrically opposite points $p, p'$ of the sphere to the point $q$ where the line $pp'$ meets the tangent plane $\pi$. The great circle of the sphere that is parallel to the plane $\pi$ is by definition projected onto the line at infinity $\Omega$ of the straight model. Observe how this correspondence preserves points, lines, and their incidence relationships.

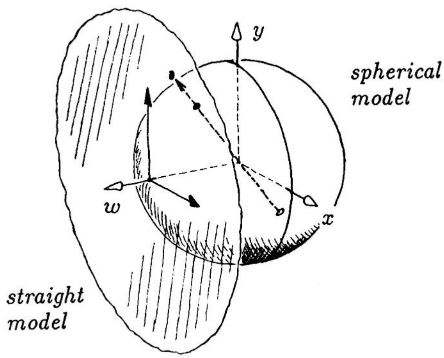



Figure 4. Central projection between the models of $\mathbf{P}_2$.

## 2. Advantages of projective geometry

Projective geometry and homogeneous coordinates have many well-known advantages over their Cartesian counterparts. From the point of geometrical computing, the following ones are particularly important:

- *Simpler formulas.* Projective geometry and homogeneous coordinates have many well-known advantages over their Cartesian counterparts. For one thing, the use of homogeneous coordinates generally leads to simpler formulas that involve only the basic operations of linear algebra: determinants, dot and cross products, matrix multiplications, and the like. All Euclidean and affine transformations, and all perspective projections, can be expressed as linear maps acting on the homogeneous coordinates of points. For example, the Cartesian coordinates of the point where the lines $ax + by + c = 0$ and $rx + sy + t = 0$ intersect are

$$\frac{(bt - cs,\ cr - at)}{as - br}$$

In homogeneous coordinates, the intersection of $\langle a, b, c\rangle$ and $\langle r, s, t\rangle$ is

$$[\,bt - cs,\ cr - at,\ as - br\,]$$

which is easily recognized as the cross product of the vectors $(a, b, c)$ and $(r, s, t)$. As this example shows, with homogeneous coordinates we can eliminate most of the division steps in geometric formulas; the savings are usually enough to offset the cost of handling an extra coordinate. The absence of division steps also makes it possible to do exact geometric computations with all-integer arithmetic.

- *Less special cases.* Homogeneous coordinates let us represent points and lines at infinity in a natural way, without any *ad hoc* flags and conditional statements. Such objects are valid inputs in many geometric applications, and are generally useful as "sentinels" in algorithms (in sorting, merging, list traversal, and so forth). They also allow us to reduce the number of special cases in theorems and computations. For example, when computing the intersection of two lines we don't have to check whether they are parallel. The general line intersection formula will work even in this case, producing a point at infinity. This point can be used in further computations as if it were any ordinary point. By contrast, in the Euclidean or Cartesian models we must disallow this special case, or explicitly test for it and handle it separately. Note that when we compose two procedures or theorems, the number of special cases usually grows multiplicatively rather than additively. Therefore, even a small reduction in the special cases of basic operations — say, from three to two — will greatly simplify many geometric algorithms.

- *Unification and extension of concepts.* Another advantage of projective geometry is its ability to unify seemingly disparate concepts. For example, the differences between circles, ellipses, parabolas, and hyperbolas all but disappear in projective geometry, where they become instances of the same curve, the non-degenerate conic.

  As another example, all Euclidean and affine transformations — translations, rotations, similarities, and so on — are unified in the concept of *projective map*, a function of points to points and lines to lines that preserves incidence. As is often the case with new unifying concepts, the class of projective maps turns out to include new interesting transformations, such as the perspective projections, that were not in any of the original classes. In fact, these maps cannot be properly defined in Euclidean geometry, since they exchange some finite points with infinite ones.

- *Duality.* Consider the one-to-one function '$*$' that associates the point $[w, x, y]$ to the line $\langle w, x, y \rangle$, and vice-versa. This mapping preserves incidence: if point $p$ is on line $l$, then line $p^*$ passes through point $l^*$. The existence of such a map ultimately implies that every definition, theorem, or algorithm of projective geometry has a *dual*, obtained by exchanging the word "point" with the word "line," and any previously defined concepts by their duals. For example, the assertion "there is a unique line incident to any two distinct points" dualizes to "there is a unique point incident to any two distinct lines."
  This *projective duality* is an extremely useful tool, in theory and in practice. Thanks to it, every proof automatically establishes the correctness of two very different theorems, and every geometrical algorithm automatically solves two very different problems. It turns out that a geometric duality with these properties can be defined only in the full projective plane. In the Euclidean plane one can construct only imperfect dualities, that do not apply to certain lines and/or points. The use of such pseudo-dualities often leads to unnecessarily complicated algorithms and proofs, with many spurious special cases [17].

## 3. Drawbacks of classical projective geometry

In spite of its advantages, the projective plane has a few peculiar features that are rather annoying from the viewpoint of computational geometry. Some of those problems, which were described in detail by Riesenfeld [19], are:

- *The projective plane is not orientable.* Informally, this means there is no way of defining "clockwise" or "counterclockwise" turns that is consistent over the whole plane $\mathbf{P}_2$. The reason is that a turn can be continuously transported over the projective plane in such a way that it comes back to its original position but with its sense reversed. For the same reason, it is impossible to tell whether two triangles (ordered triplets of points) have the same or opposite handedness. This limitation is quite annoying, since these two tests are the building blocks of many geometric algorithms.

- *Lines have only one side.* If we remove a straight line from the projective plane, what remains is a *single* connected set of points, topologically equivalent to a disk. Therefore, we cannot meaningfully ask whether two points are on the same side of a given line. More generally, *Jordan's theorem is not true* in the projective plane, since a simple closed curve (of which a straight line is a special case) need not divide the plane in two distinct regions. Even if we consider only the immediate neighborhood of a line, we still cannot distinguish its two sides, since that neighborhood has the topology of a Möbius band. See figure 5.

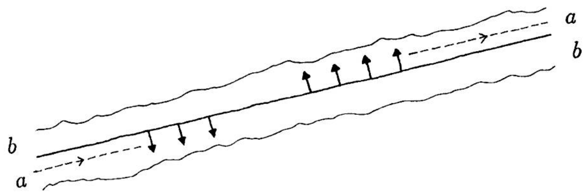



Figure 5. The neighborhood of a straight line of $\mathbf{P}_2$.

- *Segments are ambiguous.* In projective geometry we cannot define *the* line segment connecting two points in a consistent way. Two points divide the line passing through them in *two* simple arcs, and there is no consistent way to distinguish one from the other. It is therefore impossible to tell whether a point $r$ lies between two given points $p, q$.

- *Directions are ambiguous.* By the same token, we cannot define *the* direction from point $p$ to point $q$. In particular, each point at infinity lies simultaneously in two opposite directions, as seen from a finite point. This property often makes it hard to use points at infinity as "sentinels" in geometric algorithms and data structures.

- *There are no convex figures.* The notion of convex set has no meaning in projective geometry. The problem is not just that the standard definition of convex set ("one that contains every segment joining two of its points") becomes meaningless, but in fact that there is no consistent way to distinguish between convex and non-convex sets.

Of course, we can avoid all these problems by letting our definitions of segment, direction, and so on depend on a special line $\Omega$. However, we would then have to exclude certain "degenerate" cases, such as segments with endpoints on $\Omega$. The concepts thus defined will not be preserved by arbitrary projective maps and will have uninteresting duals. In fact, this "solution" means giving up on projective geometry, and retreating to the Euclidean world.