

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de doctorat:

AUTOMATITZACIÓ AVANÇADA I ROBÒTICA

Tesi doctoral

**A KINEMATIC-GEOMETRIC APPROACH TO SPATIAL
INTERPRETATION OF LINE DRAWINGS**

Lluís Ros i Giralt

Director: Federico Thomas Arroyo

Institut d'Organització i Control de Sistemes Industrials

Juliol de 2000

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de doctorat:
AUTOMATITZACIÓ AVANÇADA I ROBÒTICA

Tesi doctoral

**A KINEMATIC-GEOMETRIC APPROACH TO SPATIAL
INTERPRETATION OF LINE DRAWINGS**

Lluís Ros i Giralt

Director: Federico Thomas Arroyo

Institut d'Organització i Control de Sistemes Industrials
Institut de Robòtica i Informàtica Industrial

Juliol de 2000

A tu, Núria.

Acknowledgements

I express my sincere gratitude to Federico Thomas, not only for shaping this Thesis with his guidance, but also for his constant encouragement, friendship, and his generous sharing of ideas and time during many brainstormings at the blackboard. With him I have discovered the excitement of research.

Jordi Riera, my tutor during this PhD, always found enough time to hear my worries and orient me with good criteria. It has been really stimulating to devote part of the time to collaborate with him in his technology-transfer projects. This has brought me a rare reward in our “business”: to see the benefits and applications of previous basic research.

I owe a large debt to Walter Whiteley and Kokichi Sugihara for their kindness and for sharing part of their time with me, during two stays in Toronto and Tokyo, in the falls of 1997 and 1998, respectively. Many of the ideas in this Thesis rely on their results, and many of our contributions have developed thanks to the conversations we held. Walter introduced me to the powerful tools of Structural Geometry and Grassmann-Cayley algebra used in the first part of the Thesis. Kokichi’s very recent discovery of the Resolvable Representation is crucial to define the parametrization of correct drawings, and the correction of drawings through altering their incidence structure is joint work with him.

I am very grateful to Michel Devy from the LAAS lab at Toulouse, for allowing me to approach the world of real implementations, with the applications of this theory to the specification of unique shape using shading information, during a stay with his group in the fall of 1999.

I also thank my friends and colleagues from the Institut de Robòtica i Informàtica Industrial and from the previous Institut de Cibernètica, for their wholehearted help at all moments and the great times we spent together. I specially thank Tom Creemers, who volunteered for proof-reading the manuscript.

Núria, actually you made the single most important contribution: you showed me the true meaning of $\infty^{\infty!}$;-). Thanks for your patience, love and unconditional support.

Finally, I want to thank my family. To Cesca for the music she played for me at the piano during many breaks of the writing, at the Liceu of Barcelona. To Jordi for his encouragement and the fun we had in numerous technical discussions around this Thesis’s topics. And to my parents, even though they hardly understand it is really worth pursuing a scientific career, despite the difficulties of the Science System in this country.

This work has been partially supported by:

- My own and someone else's savings.
- The REST project, from the Spanish CICYT under contract TIC96-0721-C02-01,
- A grant from the Catalan Government, *Direcció General de Recerca*, ref. 1998BEAI200046, to support a foreign stay at the LAAS-CNRS in France.
- A Ph.D. fellowship granted by the Spanish Ministry of Education, ref. FPI94-46634232 to support the 4-year research plan, a stay at York University (Toronto), and a stay at the University of Tokyo.

This document can be retrieved from <http://www-iri.upc.es/people/ros>. The companion web page <http://www-iri.upc.es/people/ros/WebThesis/index.html> gives a visual introduction to this Thesis's topics, making use of draggeable and animated geometric constructions to illustrate the concepts. These constructions, as well as Figures 4.11, 4.12 and 4.13, have been built with the Geometer's Sketchpad, a powerful program for dynamic geometry, from Key Curriculum Press (<http://www.keypress.com/>).

Contents

I	Preliminaries	xi
1	Introduction	1
1.1	Line Drawing Interpretation	1
1.2	Objectives and Contributions	6
1.3	Outlook at the Dissertation	7
2	Basic Background	9
2.1	Realizability, Reconstruction and Correction	9
2.2	Polyhedral Surfaces	11
2.3	Obtaining Incidence Structures	12
3	Historical Development	15
3.1	Results from Machine Vision	15
3.1.1	Labelling algorithms	15
3.1.2	Sugihara's Algebraic Test	19
3.1.3	Generic Realizability	21
3.1.4	Superstrictness	22
3.2	Results from Structural Geometry	25
3.2.1	Maxwell's Theorem	25
3.2.2	Structural Geometry	27
3.2.3	The Cross-Section test	29
3.2.4	Algebraic Geometry	30
II	Interpretation of Drawings	33
4	Synthetic Geometric Tools	35
4.1	The Cross-Section Test	36
4.2	Realizability for Other Topologies	40
4.3	Graphical Construction of a Cross-Section	43
4.3.1	Trihedral Polyspheres and Polydisks	43
4.3.2	Trihedral Polydisks with Holes	45
4.3.3	General Drawings	48
5	Kinematics in the Grassmann-Cayley Algebra	53
5.1	Projective Spaces	54
5.2	Plücker Coordinates	55

5.2.1	Proper Lines in 3-Space	56
5.2.2	Lines at Infinity	56
5.2.3	Planes in 3-Space	57
5.2.4	Coordinatizations of Affine Subspaces	57
5.3	Grassmann-Cayley Algebra	58
5.3.1	Join	59
5.3.2	Meet	60
5.4	Kinematics in the Plane	61
5.4.1	Instantaneous Motion of a Point	61
5.4.2	Pure Rotations	62
5.4.3	Pure Translations	63
5.4.4	Composition of Motions	63
5.4.5	Centers of Two Hinged Bodies	64
5.5	Kinematics in 3-Space	65
5.5.1	Instantaneous Motion of a Point	65
5.5.2	Pure Rotations	66
5.5.3	Pure Translations	67
5.5.4	Composition of Motions	67
5.5.5	Screws of Two Hinged Bodies	68
6	Realizability and Reconstruction through Instantaneous Kinematics	71
6.1	Panel-and-Hinge Frameworks	72
6.2	A Kinematic Test of Realizability	73
6.2.1	A Correspondence between Liftings and Motions	73
6.2.2	Topologic Considerations	76
6.3	Reconstruction by Sweeping the Kernel	77
6.4	An Example	78
6.5	Testing Realizability in Floating-Point Arithmetic	81
6.6	Enumerating Concavities and Convexities	83
6.6.1	Feasible Sign Vectors	85
6.6.2	Constructing a Hyperplane Arrangement	88
III	Correction of Drawings	91
7	Modifying Vertex Positions	93
7.1	Introduction	93
7.2	The Overall Algorithm	96
7.3	Resolvable Sequences	96
7.4	Parameterizing Correct Projections	98
7.5	Implementation and Results	99
7.6	A Good Starting Point	102
7.7	Correction of Other Topologies	103
7.8	Future Extensions	105
8	Modifying the Incidence Structure	107
8.1	Introduction	107

8.1.1	Surface Polygonization	108
8.1.2	Unequivocal Pictures	108
8.1.3	Non-Redundancy, Minimality and Robustness	111
8.2	Trihedral Polygonal Meshes	112
8.3	Initial Polygonization	113
8.4	T and TT-Transformations	115
8.5	Optimal Propagations	117
8.5.1	Cyclic AND/OR Graphs	118
8.5.2	Feasible Height Propagations	119
8.5.3	Cost Function	121
8.5.4	Complexity Analysis	121
8.6	Further Extensions	122
9	Conclusions	125
	Bibliography	131
	Author Index	141
	Index	143

Notation

Spaces

\mathbb{R}^m	The m -dimensional vector space over the reals.
\mathcal{P}^m	The m -dimensional projective space.
\mathbb{V}	The $(m + 1)$ -dimensional vector space associated with \mathcal{P}^m .
U, U_i	Affine subspaces of \mathbb{R}^m .
\mathbb{U}, \mathbb{U}_i	Vector subspaces of \mathbb{V} , associated with U and U_i , respectively.
$\mathbb{V}^{(k)}$	The vector space of all k -extensors, and their linear combinations.
$\Lambda(\mathbb{V})$	$\Lambda(\mathbb{V}) = \mathbb{V}^{(0)} \oplus \mathbb{V}^{(1)} \oplus \dots \oplus \mathbb{V}^{(n)}$.

Elements

C, M, \dots	Points of Euclidean 2 or 3-space.
CP	The Euclidean line along the points C and P .
$C \rightarrow P$	The oriented line-segment from C to P .
PQR	The Euclidean plane through the points P , Q , and R .
$\alpha, \beta, \gamma, \dots$	Planes of Euclidean 3-space.
l, m, n, \dots	Lines of Euclidean 2 or 3-space.
$\mathbf{x}, \mathbf{c}, \mathbf{c}_i, \mathbf{p}, \dots$	Vectors of \mathbb{R}^m or \mathbb{V} , points of \mathcal{P}^m , or 1-extensors. Distinguished by context.
x_1, x_2, \dots	Coordinates of a vector or a point.
x_{ij}	The j th coordinate of the vector \mathbf{x}_i .
$\mathbf{A}, \mathbf{B}, \dots$	k -extensors, $k \geq 2$.
$\underline{\mathbf{P}}_{j_1, j_2, \dots, j_k}$	The (j_1, j_2, \dots, j_k) -th Plücker coordinate of an affine subspace.
\mathbf{P}_U	The Plücker coordinate vector of the affine subspace U .
$\mathbf{M}, \mathbf{P}, \mathbf{A}, \dots$	Matrices.
$\underline{\mathbf{M}}(i, j)$	The (i, j) -th entry of the matrix $\underline{\mathbf{M}}$.
$\underline{\mathbf{M}}_{m \times n}$	A matrix $\underline{\mathbf{M}}$ with m rows and n columns.

Operators

- Ordinary dot-product of two vectors.
- × Standard cross-product of two vectors.
- $\underline{\mathbf{A}} \underline{\mathbf{B}}$ The product of two matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$.
- $\mathbf{v}^t, \underline{\mathbf{A}}^t$ The transpose of a vector or a matrix, respectively.
- ∨ The join of the Grassmann-Cayley Algebra.
- ∧ The meet of the Grassmann-Cayley Algebra.
- $\bar{\mathbf{A}}$ The support of the k -extensor \mathbf{A} .
- ⊕ The direct sum of two vector spaces.

Kinematics

- $\mathbf{c}_{i,j}$ 1-extensor representing the instantaneous center of rotation of the i th body (or reference frame), relative to the j th reference frame (in planar motion).
- $\mathbf{L}_{i,j}$ 2-extensor representing the absolute instantaneous screw axis of the i th rigid body (or reference frame), relative to the j th reference frame (in spatial motion).
- $\mathbf{M}_{i,j}(P)$ Motion of the Euclidean point P of the i -th rigid body (or reference frame), relative to the j th reference frame.

Part I

Preliminaries

Chapter 1

Introduction

“Most of the fundamental ideas of science are essentially simple, and may, as a rule, be expressed in a language comprehensible to everyone”.

Albert Einstein —The Evolution of Physics, 1967 [27].

1.1 Line Drawing Interpretation

Emulating and explaining a human’s ability to mentally recover the spatial shape of objects from the images received on the retina has been a major goal of Machine Vision and Artificial Intelligence along the past decades. As explained by Barrow and Tenenbaum in their influential papers [6, 7], an important function of the early stages of our visual processing appears to be the transformation of brightness information in the input image into an intermediate representation that describes the intrinsic characteristics (depth, orientation, reflectance, color, and so on) of the three-dimensional surface element at each point in the image [42, 5, 60]. Support for this idea comes from several sources. Among them are the observed ability of humans to determine these characteristics, regardless of viewing conditions or familiarity with the scene; the direct value of such characteristics to applications like manipulation and obstacle avoidance; and the utility of such a representation for facilitating higher-level processing (like segmentation or object recognition) in computer vision systems, to name a few.

In principle, information about surfaces can be obtained from many sources like, for example, stereopsis, motion parallax, texture gradient, or shading. Each of these cues, however, is valid only for a particular class of situations. For example, stereopsis and motion parallax require multiple images. Determining surface shape from texture requires statistical regularity of the textural elements. Analytic techniques for determining shape from shading require accurate modelling of the incident illumination and surface photometry, which is difficult to do for most natural scenes.

However, even in the absence of such powerful cues, much valuable information about surface structure is still available. In particular, much is conveyed by *brightness disconti-*

nities, which occur wherever there are discontinuities in incident illumination (at shadow boundaries), reflectance (at surface markings) or surface orientation (at surface boundaries). The significance of surface discontinuities alone is evident from our ability to infer the three-dimensional structure of objects depicted in line drawings, such as the one in Figure 1.1. Boundary information is such a fundamental cue to tri-dimensionality that it is hard for humans to suppress it. While an exhaustive investigation of the effect of qualitative brightness events on perceived surface shape has not yet been completed, several experiments suggest that in many cases boundary information is the principal shape cue, even dominating shading, perspective and motion parallax [6]. This observation was probably the principal motivation that triggered the extraordinary boom in research on line drawing interpretation, from the late sixties to the middle eighties. Given that the ability to interpret drawings is also observed on pictures of polyhedral scenes (Figure 1.2), it seemed reasonable to focus on this class of simplified images to study the phenomenon. Over the years, research has been done to answer the following questions:

1. *Why do we see 3D scenes rather than flat drawings?* When presented with a rough sketch of a polyhedral scene, humans are able to reject “impossible figures” (Figure 1.2, a, b and c), and recover 3D shapes from correct ones (Figure 1.2, d, e, and f), despite the reduced information they offer, without textures on the surfaces, illumination patterns, or extra views. Why do we see volumetric objects and not just flat diagrams on a paper?
2. *How does the human visual system actually recover a spatial shape?* Which are the mechanisms used by the visual cortex to process the image information entering the retina and extract mental representations of the surrounding scene?
3. *How does a 2D-projection constrain the set of possible spatial interpretations that can be derived?* What plausible spatial information can be extracted by just analyzing the combinatorial and geometric structure of the drawing alone?
4. *Can we mimic the human performance on a machine?* Humans frequently communicate spatial shape information by exchanging 2D line diagrams on a blackboard, on a paper, or other flat surfaces. This diagrammatic means of communication is quite remarkable as, contrary to spoken languages, line drawings of spatial objects can be understood without much training by almost anyone in any country. Is it possible to emulate this capability on an electronic device, even if this emulation does not explain the human process at all?

As recognized by several researchers in the areas of Human Cognition and Brain Sciences (see e.g. the book by Ullman [108, page 150]), despite all efforts, we still have a narrow view of the psycho-physiological mechanisms used by a human in performing visual interpretation tasks. Thus, we still seem to be far from having a complete theory that satisfactorily answers the first two questions. On the other hand, as summarized in Chapter 3, significant progress has been made towards a complete answer to the third question and a positive one to the fourth. However, there are still several gaps to be filled

in and this Thesis's efforts are a contribution in this direction. Specifically, we have selected the following three problems and provided alternative algorithms for solving them that outperform previous existing ones:

- *Realizability*: given a drawing, decide whether it is a correct projection of some three-dimensional scene of polyhedral objects. Figure 1.2 illustrates this: the algorithm must be able to classify (a), (b) and (c) as incorrect drawings, and (d), (e) and (f) as correct ones. These examples look simple, and indeed one can check simple conditions to accept them as correct or discard them as incorrect (see Chapter 4). However, what are the conditions for a general scene? How can we detect that Escher's waterfall in Figure 1.3 cannot be physically constructed?

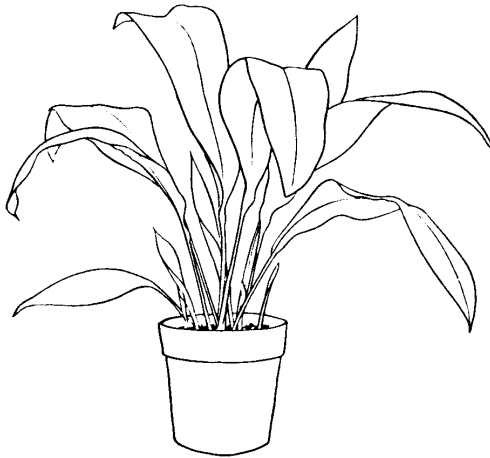


Figure 1.1: A line drawing of a three-dimensional scene (adapted from [6]).

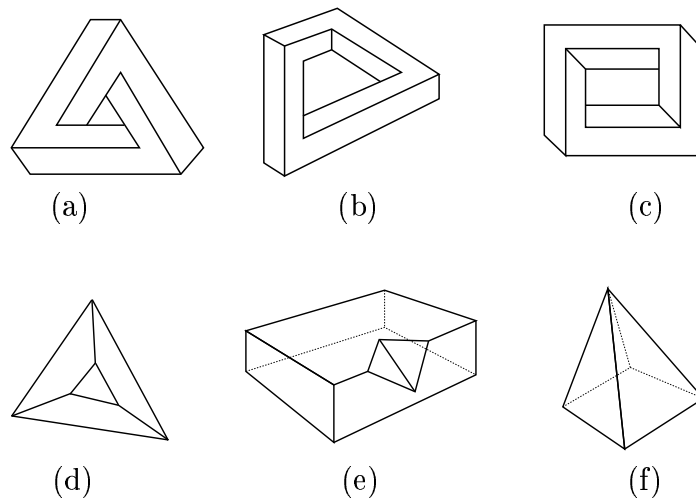


Figure 1.2: The realizability problem. Figure (a) is adapted from Penrose and Penrose [74], (b) from Draper [24], and (c) from Huffman [43].

- *Reconstruction*: if the drawing is correct, obtain *all* 3D polyhedral scenes that project onto it. As an example, Figure 1.4 shows some reconstructions of the truncated tetrahedron in Figure 1.2d. Obviously, a correct drawing can be generated from an infinite number of scenes, and a logic desire is to get a parameterization of this infinite set of possibilities. Note that this is actually all the spatial information that can be inferred from a 2D line drawing.
- *Correction*: if the drawing does not correspond to the projection of some scene, try to modify it so that it does. Figure 1.5 shows an incorrect truncated tetrahedron (a), and the two types of corrections considered in this Thesis: one that slightly moves all vertex positions (b) and another that splits some faces (c).

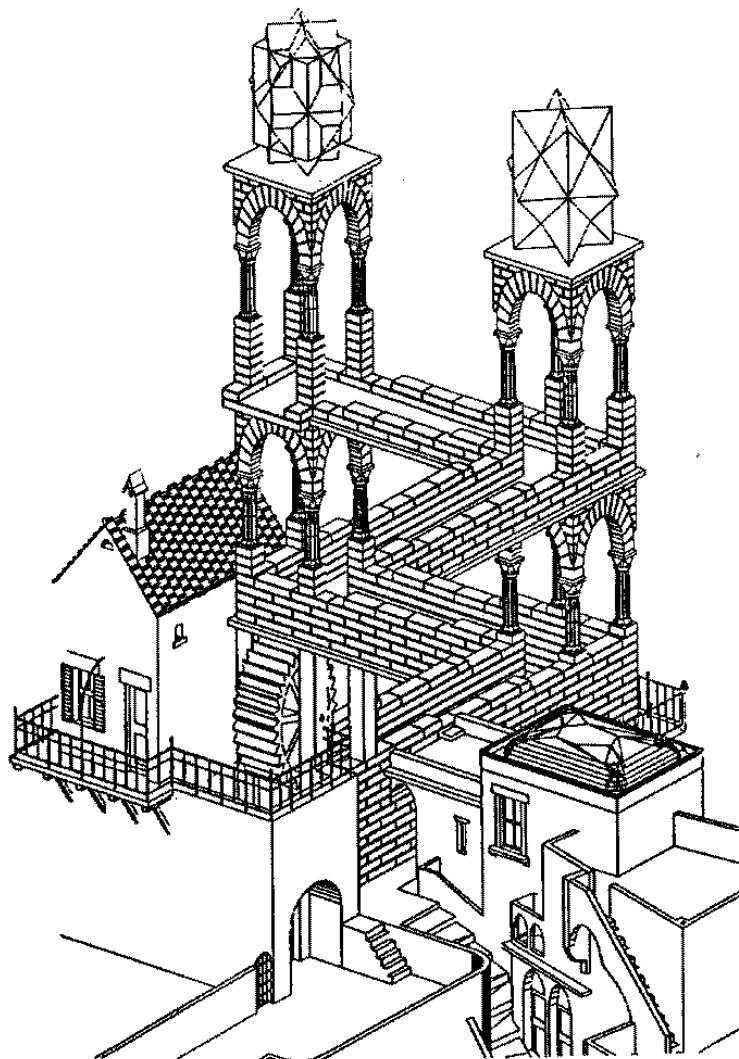


Figure 1.3: Escher's waterfall. Classify it as impossible!

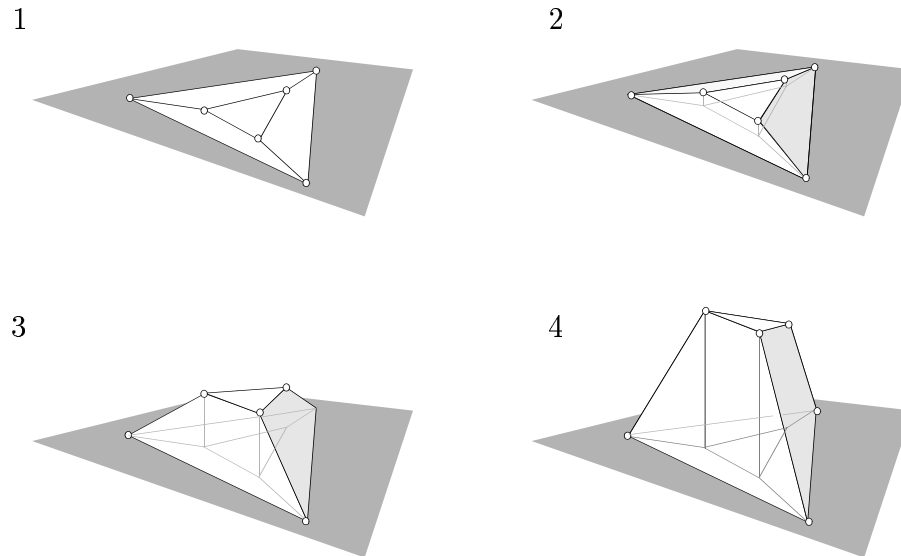


Figure 1.4: The reconstruction problem. The sequence shows how a plane drawing is *lifted* to reconstruct the spatial shape.

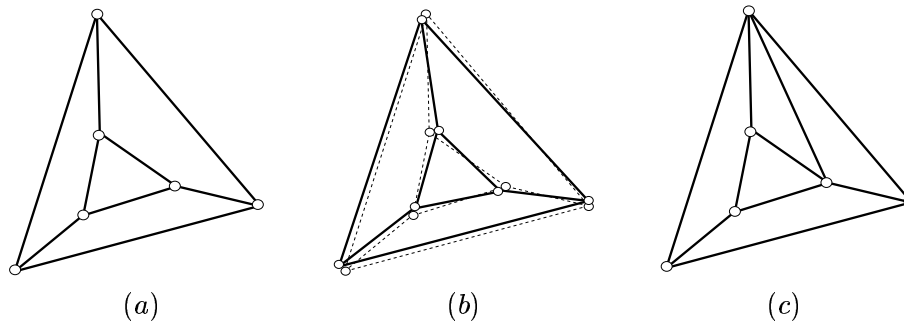


Figure 1.5: The correction problem.

The spatial interpretation of a line drawing can be done by combining modules that solve these three problems. We first take an input drawing and detect whether it is realizable or not with the first module. Then, we either get a parameterization of the whole infinite family of interpretations using the second module, or we could just select one among them if extra constraints on the 3D shape were added by the user. On the contrary, if the drawing is incorrect, we run the third module and apply the reconstruction process on the corrected drawing it produces.

When seeking a solution we have adopted an engineering point of view. We have used whatever mathematical tools that are useful to produce a system that can actually be implemented on a computer, without paying much attention as to whether the resulting methods keep some similarity or not with the corresponding visual and mental processes. Actually, it is the author's opinion that due to the highly symbolic nature of the approach, the developed algorithms have probably no relationship with their cognitive counterparts.

1.2 Objectives and Contributions

Geometric Realizability. Traditionally, the Machine Vision approach to line drawing realizability has been mainly algebraic. Although computer scientists have discovered¹ graphical techniques like the *reciprocal diagrams*², these have not been fully exploited or sometimes left apart, arguing that they only provide necessary (but not sufficient) conditions for realizability. However, a careful investigation of results from related areas of Geometry reveals the existence of complete and purely geometric tools to decide the correctness of whole families of drawings. A first goal of this Thesis has been to further exploit this geometric side, to clarify which of these tools are useful to our ends, and to complement them with additional ones where needed. Specifically, using Structural Geometry, Walter Whiteley proved in the eighties that a drawing of a spherical polyhedron is a correct projection if, and only if, one can draw a line diagram compatible with it, the *Maxwell reciprocal*. Further research led him to discover the *cross-section*, another reciprocal diagram whose existence is a necessary and sufficient condition for realizability of this type of drawings. However, usual scenes of opaque polyhedra yield line drawings with the topology of a *polyhedral surface* instead of a closed spherical polyhedron and these tests seem inappropriate at first glance. To overcome this limitation, we have selected the cross-section diagram and proved its validity also for this broader class of drawings. Along the way we have found a simpler proof of Whiteley’s result, using elementary arguments of synthetic geometry, and we show that a pencil and a straightedge are the only necessary elements to decide the correctness of drawings of trihedral polyhedral surfaces.

Algebraic Realizability. At present, however, no general realizability test using pencil and straightedge alone has been devised for general drawings so that relying on algebraic methods seems unavoidable. The most popular algebraic approach known so far is due to Kokichi Sugihara, who translated the basic problem to one of Linear Programming. Nevertheless, further improvements are still possible here. A second goal of the Thesis has been to exploit a known mapping between spatial polyhedra and the instantaneous motions of an articulated mechanism associated with their projections: a *panel-and-hinge framework*. The mapping was discovered by Whiteley and his colleagues within the Canadian Structural Topology Group and seems to be unknown to Machine Vision scientists working in these problems. As we will see, though, it yields a concise realizability test only involving elementary linear algebra. Specifically, one can decide the correctness of a drawing by simply computing a vector basis of the kernel of a matrix, made up from vertex coordinates of the drawing. As opposed to Sugihara’s, this method is numerically feasible and straightforwardly implementable in floating point arithmetic, as it can be made robust to the so-called *superstrictness* problem with a simple technique: the use of the singular value decomposition to get a basis of the aforementioned kernel.

¹Actually *rediscovered*, as explained in Section 3.2.2.

²Also known as the *gradient space* approach or the *dual diagram* technique. See for example [54, 44, 25].

Reconstruction. Moreover, the mapping above also elegantly solves the reconstruction problem. It can be shown that the instantaneous motions of the panel-and-hinge framework are linearly parameterizable and, since they are in one-to-one correspondence with the spatial reconstructions, these are also linearly described. This parameterization has other advantages as well. For example, under certain circumstances, one can obtain the $\{+, -\}$ -labellings of the drawing without solving a constraint satisfaction problem, and without consulting a junction dictionary. These labellings correspond to the different shapes that the spatial reconstruction can have, according to the concavity (+) or convexity (-) of their edges, and we will give an algorithm to enumerate them all. Finally, another derived result is a model explaining the Necker reversal phenomenon: given a spatial reconstruction there is always a mirror reconstruction that reverses the edge types, converting convex edges to concave ones and vice versa.

Correction. A correction module is usually necessary, since drawings with a generic geometry are seldom correct. This means that if we take the combinatorial elements of a drawing (the abstract vertices, edges and faces it depicts) and we give them an arbitrary geometric position, say by fixing the vertices, then the drawing will most probably be incorrect, because the only coordinates yielding correct drawings lie on a zero measure set of \mathbb{R}^{2v} (if v is the number of vertices). This happens on most (but not all) combinatorial structures, and Sugihara and Whiteley found combinatorial criteria that classify them. In practice, if a drawing is got from a hand-made sketch or by filtering a real image, we have the same effect: the coordinates will seldom lie in right places, and reconstruction will not be possible. The third goal was to efficiently solve this problem. To this end, we have devised the first algorithm for line drawing correction that allows the movement of all the vertices simultaneously. This yields better solutions than existing approaches, which only allowed the displacement of a subset of the vertices, or failed to correct some combinatorial structures. Moreover, we have provided a second correction strategy that keeps the vertices fixed but alters the combinatorial structure, with the side-effect of relaxing the constraints over the set of possible reconstructions.

1.3 Outlook at the Dissertation

The Dissertation is structured in three parts. Part I groups the introduction, basic background and a historical perspective of related work. Then, Part II presents the solutions to the realizability and reconstruction problems, which are intimately linked. Finally, Part III contains two chapters, each one of them devoted to a different drawing correction strategy. Each chapter's contents can be outlined as follows.

Chapter 2 starts with formal definitions of the realizability, reconstruction and correction problems. We then proceed with the basic background and vocabulary needed in later explanations. A basic assumption of the Thesis is that we already know the *incidence structure* of the drawing. This structure is formally presented, and methods to obtain it are briefly reviewed, with pointers to the literature for details on them.

Chapter 3 gives an account of previous work in the area. We have made an effort not only to survey the well-known contributions due to the Machine Vision community, but also several applicable results from Structural Geometry that seem to be ignored by our community.

Chapter 4 treats the realizability problem from a synthetic geometric perspective. The cross-section test is presented in detail and a proof of its validity for polyhedral disks is given. Then we delimit the class of *trihedral drawings* and show that their correctness is decidable just using pencil and straightedge.

Chapter 5 is an introductory exposition to the Grassmann-Cayley algebra and its application to the kinematic analysis of mechanisms. The algebra is presented in the least abstract way possible, having an engineering audience in mind. To make the presentation self-contained, the chapter begins with an introduction to projective spaces and Plücker coordinates.

Chapter 6 uses the tools from the previous chapter to describe the algebraic algorithms for realizability and reconstruction, and follows their application on simple examples. The enumeration of all feasible $\{+, -\}$ -labellings is then considered, and we see that it can be reduced to finding the sign vectors of all cells in a hyperplane arrangement. Computational Geometry has already developed efficient algorithms for this canonical problem, and one is described there.

Chapter 7 provides a solution to the correction problem that only involves the modification of the vertex coordinates, while keeping fixed the combinatorial structure of the drawing.

Chapter 8, on the contrary, solves the correction problem with a rather opposite strategy. The numerical part remains fixed and only the combinatorial structure is conveniently modified. This approach is developed for trihedral drawings in the context of an application to 3D robust modelling, but is equally applicable to any type of line drawing.

Chapter 9 concludes the Thesis and suggests several points for further research.

Author and keyword indices. We include author and keyword indices at the end. For each author, the former lists the pages of the manuscript that contain citations to one of his/her publications. The latter gives pointers from key concepts to the places where they appear in the text.

Chapter 2

Basic Background

“The individual mathematician feels free to define his notions and set up his axioms as he pleases. But the question is: will he get his fellow mathematician interested in the constructs of his imagination?”

Hermann Weyl, 1951 [67].

2.1 Realizability, Reconstruction and Correction

Before we give precise definitions for the three problems stated above, we must first decide how a line drawing is described to a computer —i.e. what information is supposed to be known about such a diagram of lines. Actually, it is useful to distinguish between two types of descriptions, a more primitive one, which we call *raw line drawing*, and a richer one, called *augmented line drawing*. Informally speaking, a *raw line drawing* is a 2D diagram made up of two entities, called *segments* and *junctions*, corresponding to the spatial edges and vertices of a polyhedron, respectively. Also, one can give additional information about which junctions correspond to vertices of a same face of the spatial polyhedron. This information is usually collected in a bipartite graph, called the *incidence structure* and we call a drawing where this structure is provided an *augmented line drawing*. In general, given a raw drawing, one can identify several incidence structures on it, and hence several augmented line drawings can correspond to one raw line drawing. Next, we formalise these concepts.

A *raw line drawing* is a triple $\mathcal{D}_{raw} = (J, S, \mathbf{r})$, where J is an abstract set of *junctions*, S is an abstract set of *segments*, with $S \subset J \times J$, and \mathbf{r} is a plane location map that assigns a point of the XY plane to every junction in J :

$$\begin{aligned} \mathbf{r} : J &\longrightarrow \mathbb{R}^3 \\ \mathbf{r}(v_i) &= (x_i, y_i, 0) . \end{aligned}$$

An *incidence structure* is a triple $S = (V, F, I)$, where $V = \{v_1, \dots, v_n\}$ is an abstract set of *vertices*, $F = \{f_1, \dots, f_m\}$ is an abstract set of *faces*, and $I \subset V \times F$ is called the

incidence set. When we refer to the vertices of face f_i , we mean all vertices $v_i \in V$ such that $(v_i, f_i) \in I$. Two faces f_i and f_j are *adjacent* if they share at least two vertices, say v_k and v_l —i.e., $(v_k, f_i), (v_k, f_j), (v_l, f_i), (v_l, f_j) \in I$.

An *augmented line drawing* \mathcal{D}_{aug} is an incidence structure $S = (V, F, I)$ plus a plane location map \mathbf{q} that assigns a point of the XY plane to every vertex in V :

$$\begin{aligned} \mathbf{q} : V &\longrightarrow \mathbb{R}^3 \\ \mathbf{q}(v_i) &= (x_i, y_i, 0) . \end{aligned}$$

A *lifting* or *interpretation* of \mathcal{D}_{aug} is a spatial location map \mathbf{p} that assigns a point (x_i, y_i, z_i) to every vertex v_i of V on the vertical line through $\mathbf{q}(v_i) = (x_i, y_i, 0)$,

$$\begin{aligned} \mathbf{p} : V &\longrightarrow \mathbb{R}^3 \\ \mathbf{p}(v_i) &= (x_i, y_i, z_i) , \end{aligned}$$

and such that for every face $f_i \in F$, all its vertices lie on a same plane of \mathbb{R}^3 , called the *face plane* of f_i . The lifting is *trivial* if \mathbf{p} maps all the vertices to a same plane of \mathbb{R}^3 . The lifting is *sharp* if every two adjacent faces receive different face planes.

The realizability problem is to decide whether the vertices of \mathcal{D}_{aug} can be “vertically lifted” to produce a sharp lifting. The drawing is said to be *correct*, or *realizable*, if such a lifting exists, and *incorrect* otherwise. The reconstruction problem is to obtain a parameterization of all liftings of \mathcal{D}_{aug} , including the trivial ones. The correction problem is to minimally alter \mathcal{D}_{aug} , either modifying the incidence structure or the vertex positions $\mathbf{q}(v_i)$, to produce a correct line drawing as close to \mathcal{D}_{aug} as possible, under some pre-specified criterion.

Normally, we want to feed a computer with just a raw line drawing, and let it derive all spatial shapes that can be inferred from it, if any. This would be the use if the computer is, for example, interpreting a designer’s hand-made sketch . In this Thesis, however, we will assume that an incidence structure has already been identified on such a raw drawing and that the input is actually an augmented line drawing. Here, an obvious question is how one can derive all augmented drawings that could correspond to a raw drawing. Although it is not our concern to develop efficient methods for this purpose, in Section 2.3 we shortly comment how this can be done and point the reader to existing literature on the topic. Hereafter, the term line drawing (or drawing, for short) will refer to an augmented line drawing, unless otherwise stated.

Moreover, although the definition of incidence structure leaves plenty of room for the topology of the spatial liftings that can be derived, in this Thesis we will be primarily interested in the incidence structures induced by usual scenes of opaque polyhedra . These are described in Section 2.2.

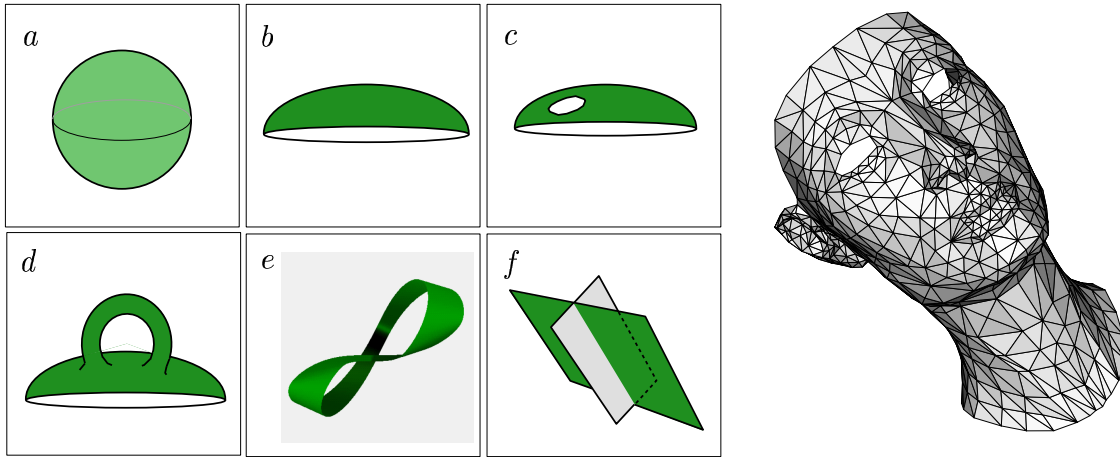


Figure 2.1: Some valid topologies for a polyhedral surface: a sphere (a), a disk (b), a disk with one hole (c) and a disk with one handle (d). Some excluded topologies: a Möbius band (e) and two intersecting sheets (f). Right: a triangulated polysurface with two holes.

2.2 Polyhedral Surfaces

Polyhedral scenes are typically made of solid objects. However, we employ a more convenient model for the liftings of \mathcal{D} : we assume that these are polyhedral surfaces homeomorphic¹ to a disk with any number of holes. These and other surfaces used along the Thesis are defined next.

A *polyhedral surface*, or *polysurface* for short, is a piecewise linear and continuous 2-manifold, made up with planar polygons called *faces*, glued in pairs along their edges. (A *2-manifold* is a two-dimensional orientable surface in 3-space where the neighbourhood around any point is homeomorphic to a disk.) Figure 2.1 depicts some possible topologic shapes that a polysurface can have, differing only in the number of *holes* and *handles* they have. The shapes in Figures 2.1 e and f are not included in our definition, as they are not 2-manifolds: a Möbius band is not orientable, and 2 intersecting sheets have points where the neighbourhood is not homeomorphic to a disk.

A *polyhedral sphere*, or *polysphere* for short, is a polysurface homeomorphic to a sphere. Analogously, a *polyhedral disk*, or *polydisk*, is a polysurface homeomorphic to a disk. A polydisk with n holes is a polysurface homeomorphic to a disk with n holes. We say that a polysurface is *trihedral* if each of its interior vertices has exactly three incident faces. Trihedral polysurfaces yield the so-called *trihedral drawings*, when projected.

The *faces* of a polysurface are the maximal 2-dimensional planar subsets of points on it. The *edges* are the maximal 1-dimensional linear subsets of the boundaries of the faces. The *vertices* are the endpoints of the edges. Two faces are *adjacent* if they share at least one edge. Also, an edge is said to be adjacent to the faces it belongs to. We distinguish between *interior* and *boundary* edges, depending on whether they have *two* or just *one*

¹Homeomorphic means “topologically equivalent”.

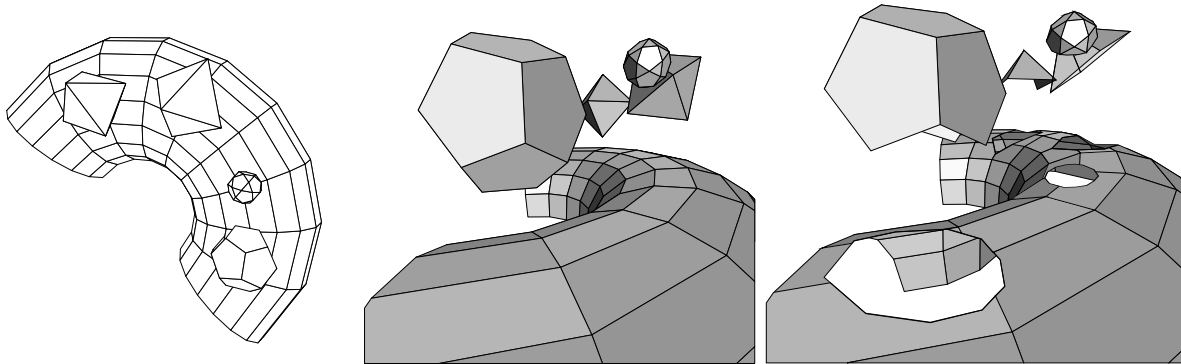


Figure 2.2: Drawings of opaque polyhedra are spatially interpreted as a collection of polysurfaces.

adjacent face, respectively. We call *boundary vertex* one that has at least one incident boundary edge, otherwise the vertex is said to be *interior*.

The polydisk with holes is an accurate model to represent the spatial shapes that can be recovered from \mathcal{D} . Indeed, since typical objects are opaque, the drawing does not usually display the projections of the hidden vertices, edges, and faces. Actually, it only provides information on the “topmost” viewed part of the scene, which, as seen from the center of projection, can be regarded as a collection of polyhedral disks, possibly containing interior holes. This is illustrated in Figure 2.2. The line drawing to the left is a vertical projection of the scene in the center, where hidden parts below the topmost surface are clearly revealed. But all we can recover are spatial liftings as the one to the right, with five polysurfaces. Observe the holes on the half-torus, produced by the shadows of the floating objects above.

2.3 Obtaining Incidence Structures

We outline how a candidate incidence structure can be obtained from a raw line drawing. As above, we focus on drawings of opaque polyhedra, showing only the visible edges. The case of drawings depicting also the hidden edges is thoroughly studied in the works by Marti et al. [62], Sugihara [104, chapter 4] and Lipson et al. [53].

$S = \{V, F, I\}$ is derived with the following process. We start adding a vertex in V for every junction of \mathcal{D} and a face in F for every polygonal region. Initially, I is empty. The process continues by replacing some vertices in V by new ones, and by adding incidence pairs in I .

First, note that a line segment of \mathcal{D} arises either as the projection of an edge of the spatial object (one where the two faces coincide in 3-space), or as the projection of a polyhedral part occluding another behind (and the two regions at both sides of the segment are actually separated faces in space). In correspondence with the polysurface model introduced above these two types of edges will be called *interior* and *boundary*,

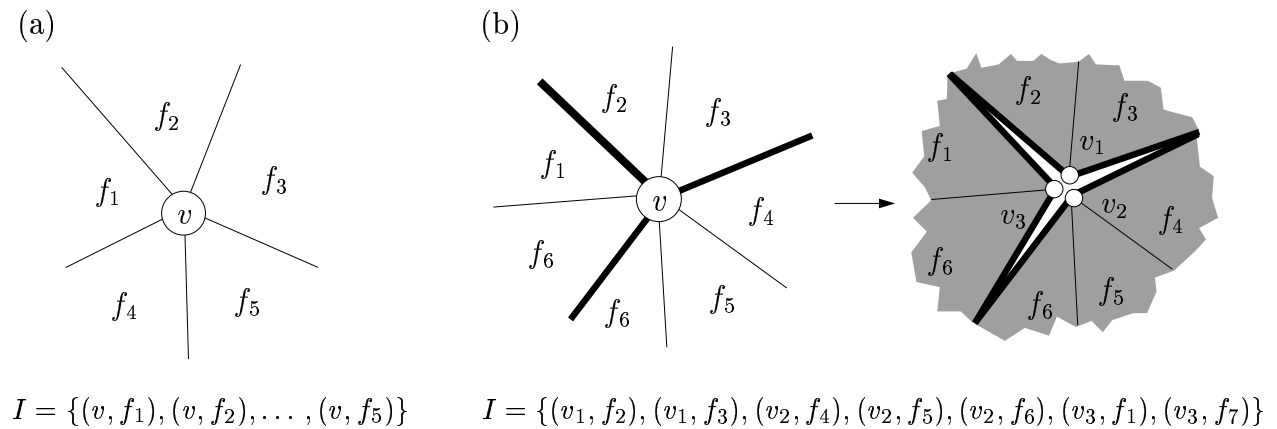


Figure 2.3: Incidence structure around a vertex with (a) and without (b) boundary edges.

respectively.

Assume that every segment type is known. If all the incident segments at a junction are interior, then the vertex v of this junction actually lies on all faces around it in 3-space, and we simply add an incidence pair (v, f) for every face f around v . On the other hand, if v has n incident boundary edges, then it can be thought of as n aligned spatial vertices v_1, v_2, \dots, v_n that coincide when projected, but they lie at different depths when seen from the center of projection. Figure 2.3 illustrates this. The n boundary edges divide the faces around v into n “connected” sectors: inside every such sector, the faces are separated by interior edges and if we travel from one face to the other, no discontinuity of height is encountered. To construct the incidence structure around v we substitute v by v_1, v_2, \dots, v_n in V and we add an incidence pair (v_i, f_j) for every face f_j around the connected sector of v_i .

It remains to see how an edge can be classified as either interior or boundary. This would depend on the application. For example, if the drawing comes from a hand-made sketch, the designer may have included easy-to-recognize symbols on the edges, to indicate their type. If this is not possible, we can use any of the methods that obtain a *consistent labelling* of the drawing, one that assigns either a *convex* (+), a *concave* (−) or an *occlusive*

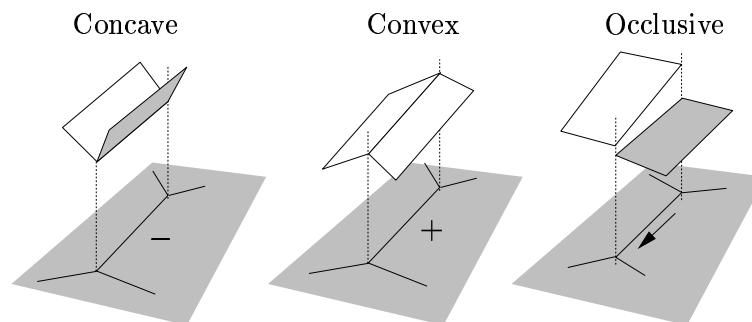


Figure 2.4: Types of edge labels.

(\rightarrow , \leftarrow) label to every edge, according to some consistency constraints. Figure 2.4 gives the meanings of these labels. An edge is convex (concave) if it can be spatially interpreted as a convex (concave) edge between two spatial faces. An occlusive edge indicates a discontinuity of height from one face to the other, with the arrow pointing to the direction where the face to the right is closer to an observer at the center of projection.

There is extensive literature on methods that produce such labellings. Although Section 3.1.1 will expand more on this topic, let us mention here that the techniques of Huffman [43], Clowes [16] and Waltz [109] are quite celebrated in the literature. See for example [104, chapter 2] for a survey of results until 1986. Recent developments due to Parodi show that it is possible to quickly get consistent labellings once the vanishing points of the scene are known [71, 72, 68]. The computation of vanishing points is itself a thoroughly studied problem in Computer Vision. A recent publication comparing several methods is [94].

Once a $\{+, -, \rightarrow, \leftarrow\}$ -labelling has been computed, a $\{boundary, interior\}$ -labelling is simply obtained by replacing $+$ or $-$ labels by an *interior* label, and \rightarrow or \leftarrow labels by a *boundary* label.

The process outlined so far will generate an augmented line drawing \mathcal{D} whose incidence structure, in fact, can be seen as the union of the incidence structures of possibly *several* polydisks with holes. Actually, it is not necessary to deal with \mathcal{D} as a whole to decide its correctness. It suffices to divide \mathcal{D} into several drawings, say $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{n_p}$, each depicting the projection of one of the polydisks, and then individually test each of these drawings. Clearly, the whole drawing is realizable if and only if all of the derived drawings $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{n_p}$ are realizable too. Moreover, the liftings of \mathcal{D} are simply found by lifting individually each of the \mathcal{D}_i 's. Given these observations, and without loss of generality, we will hereafter assume that the input line drawing depicts the projection of a *single* polydisk, with any number of holes.

Chapter 3

Historical Development

“We all live in the great, dynamic web of change. It links us to one another and, in some ways, to everything in the past. And in the way that each of us influences the course of events, it also links us to the future we are all busy making, every second”.

James Burke —The Pinball Effect. How Renaissance Water Gardens Made The Carburetor Possible, 1996 [14].

The spatial interpretation of line drawings has drawn attention from mainly two scientific communities: Machine Vision and Structural Geometry. However, one appreciates a considerable separation in their respective works, as can be seen from the scarce collaboration they have maintained, with a few exceptions. This is probably due to the rather different applications they pursue: while the former want to construct a theory of vision, the latter concentrate on the kinematic and static rigidity of articulated frameworks. Nevertheless, along the years several threads connecting both problems have emerged, which justify an exchange of ideas and results. This chapter offers a historical perspective of them, to aid delimiting the research frontiers we found when we started this work. Along the way, we describe the previous existing techniques and point the reader to related sections of the Thesis where an enhancement is achieved or an alternative is given.

3.1 Results from Machine Vision

3.1.1 Labelling algorithms

In Machine Vision, the interest in line drawing interpretation starts with the pioneering work by Guzman [36, 37], who tried to automatically decompose a drawing of a pile of objects into regions, so that each region corresponds to one object, without any a priori knowledge of the models present in the scene. He used configurations of lines at junctions to achieve such a decomposition and successfully tested the method on several

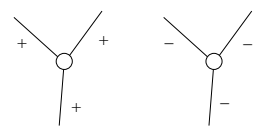
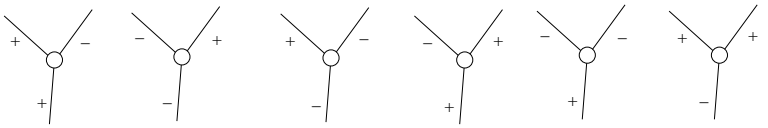
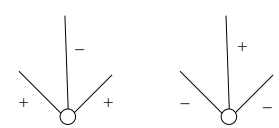
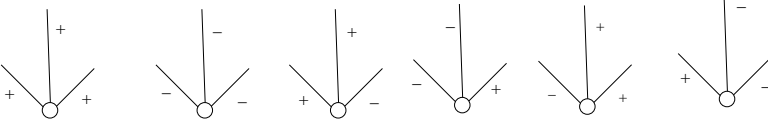
	Possible	Impossible
Y shaped		
W shaped		

Figure 3.1: Possible convex ($-$) and concave ($+$) configurations around a trihedral vertex.

complicated drawings. However, his algorithms are strongly based on heuristic rules for which no logical or geometric foundation is provided, and one can easily find examples that cheat them [4].

In 1971, Huffman [43] and Clowes [16] directly attack the realizability problem in a more organized way, exploiting the fact that there is only a limited number of different feasible assignments of *concave* ($-$), *convex* ($+$) or *occlusive* (\rightarrow , \leftarrow) labels to the segments around a given junction. (See Section 2.3 and Figure 2.4 for the meaning of these labels.) This can be seen in Figure 3.1, where a *trihedral junction* with exactly three incident faces is depicted, along with its feasible and impossible $\{+, -\}$ -labellings. If the junction is “Y-shaped” then either all three edges are convex or all three are concave, but mixed configurations are not possible. Analogously, only two configurations arise when the junction is “W-shaped”. By inspection, we can enumerate all configurations and register them in a *junction dictionary* extending the one in Figure 3.1. Obviously, a drawing is only correct if it has a *consistent* $\{+, -, \rightarrow, \leftarrow\}$ -labelling, one that assigns a configuration in the dictionary to each junction, while ensuring that the two junctions of a segment yield the same label for this segment. Then Huffman and Clowes see that the set of all valid labellings can be found by solving a constraint satisfaction problem with the junctions as variables and the entries in the dictionary as possible values for them.

Several other authors refined this scheme to accept a more complex input. For example, in 1975 Waltz treated pictures with shadows and cracks [109]. His algorithms for obtaining consistent label assignments were extended by Mackworth in 1977 [55] and nowadays both works are considered among the pioneers of Constraint Programming. Procedures to treat pictures with hidden segments were provided by Sanker in 1977 [83], and Sugihara in 1978 [98]. Drawings of paper-made objects were considered in 1980 by Kanade in his theory of the Origami World [48, 49]. Quite recently, algorithms for testing the realizability of line drawings of curved objects have also been investigated by Cooper [17].

The main drawback that arises with the labelling scheme is that line drawings having consistent $\{+, -, \rightarrow, \leftarrow\}$ -labellings are not guaranteed to be a projection of a polyhedron. For example, although all correct drawings have a consistent labelling (Figure 3.2a) and many incorrect drawings cannot be legally labelled (Figure 3.2, b and c) the drawings in

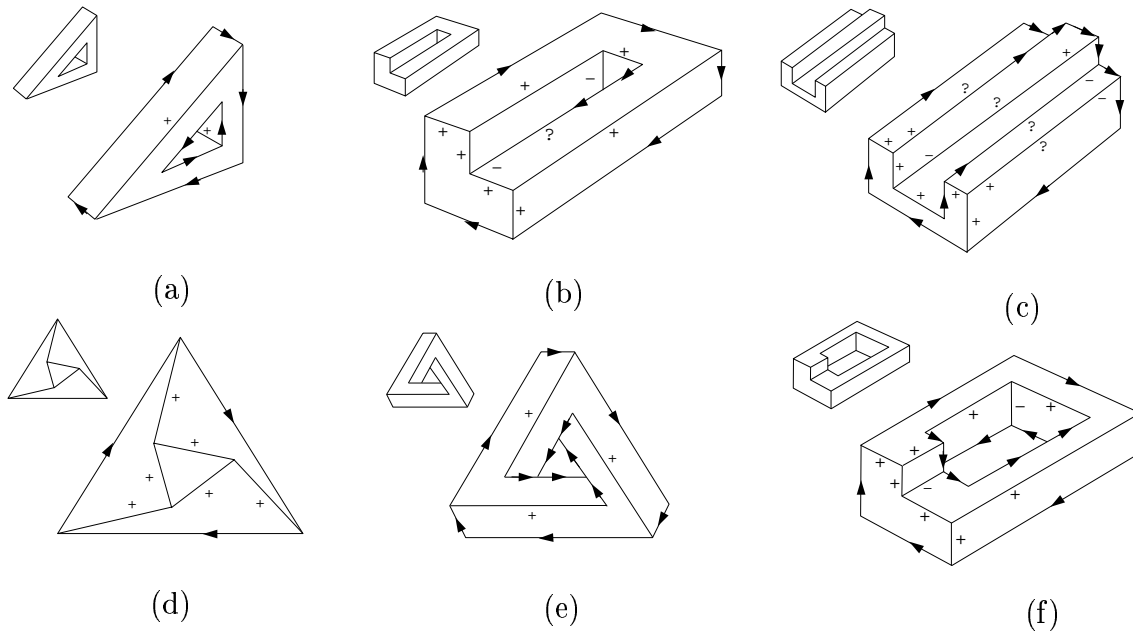


Figure 3.2: The existence of a consistent edge labelling does not guarantee the correctness.

Figure 3.2d, e and f, are incorrect but admit the indicated consistent labellings.

A second drawback is the need of pre-computed junction dictionaries. Indeed, it is impossible to enumerate all feasible label configurations for the junctions because, in principle, any number of faces can meet on them. But even if we assume a maximal amount of copunctual faces, the dictionary is not easy to construct since, as pointed out in [104, pag. 33], it is difficult to list all possible junctions having a given number of faces, simply by inspection. Nevertheless, this second limitation is overcome for a subclass of drawings in Chapter 6. Namely, for line drawings of polysurfaces without boundary edges the realizability can be decided just finding a vector base of the kernel of a matrix, without solving a constraint satisfaction problem nor consulting a dictionary.

Finding a consistent labelling is NP-complete in general. This was proven in 1988 by Kirousis and Papadimitriou, who reduced the labelling of drawings of trihedral opaque polyhedra from an instance of the boolean satisfiability problem 3-SAT¹ [50]. However, there are particular cases where the labelling can be found in linear time in the number of edges or junctions. In [50] Kirousis and Papadimitriou also show that this happens when the drawing depicts a “Legoland scene”, where all edges take only one of three *known* orthogonal directions. Alevizos proves in 1991 that the same complexity holds for scenes of polyhedra where the hidden edges are provided and identified [2]. In 1994, Parodi and Torre enlarge the class of drawings labellable in polynomial time, proving that if the vanishing points of the scene are known, then a labelling can be found in $O(Nn)$, where

¹3-SAT is the problem of finding a truth assignment for the variables occurring in a boolean expression in 3-conjunctive normal form —i.e., with at most three literals per clause.

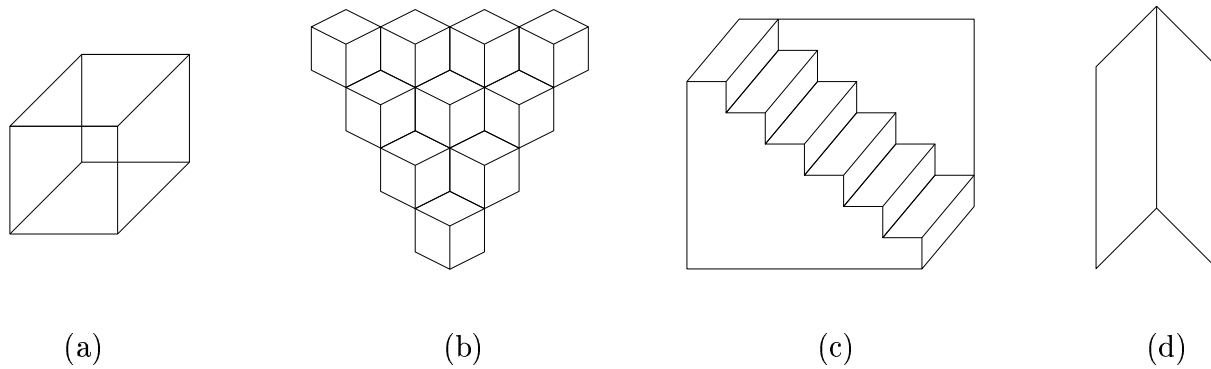


Figure 3.3: The Necker reversal phenomenon.

N is the number of edges, and n is the number of vanishing points [72, 71, 70, 68]. Several methods for computing the vanishing points are surveyed and compared in [94].

Despite the difficulty of finding a consistent labelling, the human brain seems to grasp at a glance the 3D structure associated with a line drawing. Kirousis and Papadimitriou note that a possible explanation of this discrepancy is that the worst-case complexity does not reflect the real difficulty of labelling line drawings, which might be far less in the average or “typical” cases. In 1998, Parodi et al. provide an empirical analysis that supports this conjecture [69]. They generate many random instances of polyhedral scenes, project them, and label the resulting drawings. The analysis certainly reveals that the median-case complexity is linear in the number of junctions.

An intriguing psychologic phenomenon related to the labelling of drawings is that of the *Necker reversal*. An illustrative example is the famous Necker cube, named after his creator, the Swiss crystallographer Louis Albert Necker (Figure 3.3a), who in the mid-1800’s saw cubic shapes spontaneously reverse in perspective. The image that such a cube projects onto the retina may be interpreted by the brain in two different ways, depending on which square is thought to be in the foreground and which in the background. The human mind is unable to decide which is the true lifting of the cube and keeps switching between one interpretation and the other. This may happen virtually in any line drawing, but there are other famous examples where it easily arises, such as the cubic structure in Figure 3.3b, the reversible staircase by the German mathematician Ernst Schröder (Figure 3.3c), or the reversible book by the Austrian physicist Ernst Mach (Figure 3.3d). In all cases the inferred liftings seem to come in pairs: if we see a lifting where some edges are convex and some others concave, then there exists a “mirror” lifting that reverses the edge types converting convex edges to concave ones and vice versa. In Chapter 6 we will provide a mathematical justification of this empirical fact, together with an algorithm that computes all Necker reversals of a line drawing. Although a need to characterize them has been recognized [56, 52], we believe this is the first time that an explicit process to compute all Necker reversals is given.

Triple type	Relationship	Translated to constraint
vertex-face	$(v \text{ ON } f)$	$(v_x, v_y, v_z, 1) \cdot (A_f, B_f, 1, D_f)^t = 0$
	$(v \text{ ABOVE } f)$	$(v_x, v_y, v_z, 1) \cdot (A_f, B_f, 1, D_f)^t \geq 0$
	$(v \text{ BELOW } f)$	$(v_x, v_y, v_z, 1) \cdot (A_f, B_f, 1, D_f)^t \leq 0$
	$(v \text{ STRICTLY ABOVE } f)$	$(v_x, v_y, v_z, 1) \cdot (A_f, B_f, 1, D_f)^t > 0$
	$(v \text{ STRICTLY BELOW } f)$	$(v_x, v_y, v_z, 1) \cdot (A_f, B_f, 1, D_f)^t < 0$
vertex-vertex	$(v_i \text{ ABOVE } v_j)$	$v_{1_z} - v_{2_z} \geq 0$
	$(v_i \text{ BELOW } v_j)$	$v_{1_z} - v_{2_z} \leq 0$

Table 3.1: Translation of incidence triples.

3.1.2 Sugihara's Algebraic Test

After several other attempts to provide not only necessary but also sufficient conditions [90, 91, 88, 89], within the period 1982-1984 Sugihara finally proposes a complete set of constraints that characterizes realizable line drawings [100, 103, 102, 101]. These are derived as follows. First, the Huffman/Clowes approach to get a $\{+, -, \rightarrow, \leftarrow\}$ -labelling of the edges is used. Then, the *spatial structure* is derived from the edge labels: a set of abstract triples expressing position relationships between vertices and faces. There are two types of triples:

- Vertex-face triples of the form (v, r, f) , where v is a vertex of the drawing, f is a face, and r is a relationship giving the position of v relative to f . The possible values for r are “ON”, “STRICTLY ABOVE”, “STRICTLY BELOW”, “ABOVE”, and “BELOW”, according to whether v must lie exactly *on* the face-plane f , or strictly above/below f , or above/below f but possibly on f too, respectively.
- Vertex-vertex triples of the form (v_i, r, v_j) , where v_i and v_j are vertices and r is a relationship giving the position of v_i relative to v_j . The possible values for r are, “ABOVE”, and “BELOW”, depending on whether the z-coordinate of v_i is greater than that of v_j or conversely.

According to the labels every vertex and edge produce a number of such triples (see Figure 3.4), and the final step is to translate them into an algebraic constraint, as indicated in Table 3.1. There, v_x, v_y and v_z are the coordinates of vertex v , and A_f, B_f and D_f are the plane coefficients of face f . These constraints are linear, since v_x and v_y are known and equal to the coordinates of the corresponding junction in the drawing. If we have n vertices and m faces, then there are $3m + n$ unknowns. We can collect them in a $(3m + n)$ -tuple w , and write all equations in matrix form as

$$\left. \begin{array}{l} \underline{\mathbf{A}} \cdot w = 0 \\ \underline{\mathbf{B}} \cdot w \geq 0 \end{array} \right\}, \quad (3.1)$$

where $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ are constant matrices made up of vertex coordinates. Sugihara proves that a drawing is correct if and only if this system has a solution, which can be efficiently

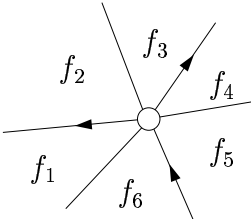
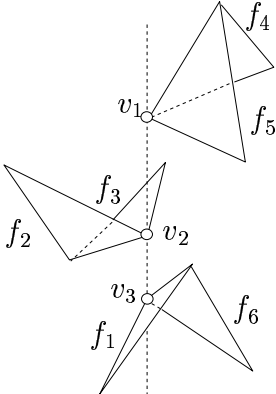
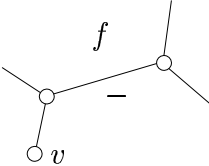
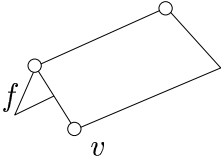
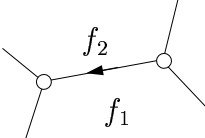
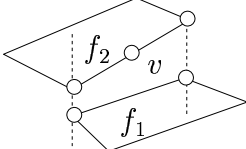
Feature	Spatially interpreted as	Triples added
		$(v_1, ABOVE, v_2)$ $(v_2, ABOVE, v_3)$ $(v_3, BELOW, v_1)$ (v_2, ON, f_2) (v_2, ON, f_3) (v_1, ON, f_4) (v_1, ON, f_5) (v_3, ON, f_1) (v_3, ON, f_6)
		$(v, STRICTLY BELOW, f)$
		(v, ON, f_2) $(v, STRICTLY ABOVE, f_1)$

Figure 3.4: The spatial structure.

decided using linear programming. (Actually there are some strict inequalities in $\underline{\mathbf{B}} \cdot w \geq 0$, and linear programming is not directly applicable, but Sugihara translates the system to an equivalent one, without strict inequalities.)

It is worth mentioning that in 1982 Sugihara already thought having defined a set of necessary and sufficient conditions for realizability [99], but in 1984 Shapira gave a counterexample proving that the test, as defined in [99], was still giving only necessary conditions [87]. This impelled Sugihara to refine his equations in [103], changing the definitions of $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ to the ones given above. It is easy to follow Shapira's counterexample on Figure 3.5. First, she draws the labelled line drawing in the center and notes that it is correct, as it can be obtained projecting the scene on the left. Hence, Sugihara's equations for this drawing, $\underline{\mathbf{A}} \cdot w = 0$ and $\underline{\mathbf{B}} \cdot w \geq 0$, do have a solution, say w^* . However, if we add just one edge e between v_1 and v_2 , labelled as shown in Figure 3.5-right, we get an incorrect line drawing, because the label " \leftarrow " on e asks for a discontinuity of height when we cross from one side of e to the other, but, at the same time, both sides are part of the same *planar* face, which is impossible to accomplish. The point is that the addition of e does not add nor remove any equation to the system $\underline{\mathbf{A}} \cdot w = 0$ and $\underline{\mathbf{B}} \cdot w \geq 0$, because the spatial structure of both drawings, as defined by Sugihara in [99], is the same. Since

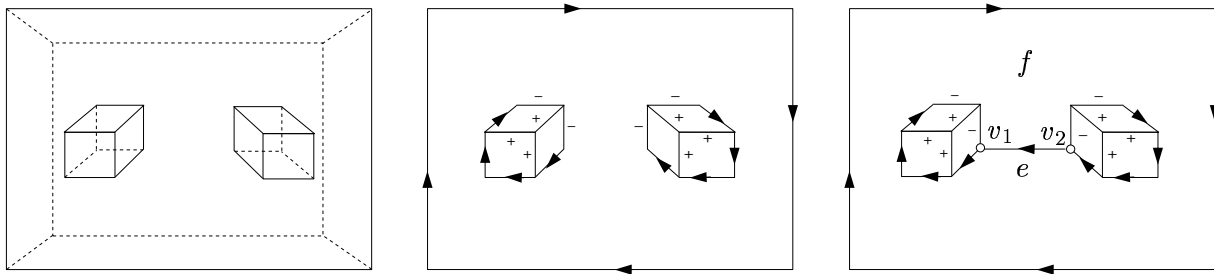


Figure 3.5: Shapira's counterexample.

we saw that these equations have a solution (w^*), they classify the drawing on the right as correct, contrarily to what is expected. In particular, for e to be interpreted as a true occlusive edge, the equations $\underline{\mathbf{A}} \cdot w = 0$ and $\underline{\mathbf{B}} \cdot w \geq 0$, should include the constraints “ v_1 must be above face f ”, “ v_2 must be above face f ” and “the midpoint of e must be above face f ”. These relations are neglected in the spatial structure proposed in [99], but they are properly introduced in [103].

Despite the attempts to establish broader theories of line drawings [66, 65], it seems that after the appearance of Sugihara's book, the interest in the realizability problem decayed among the Machine Vision community, partly because his test was considered a definite solution, and partly due to the gradual turn of the efforts to the interpretation of real images, instead of perfect line drawings of ideal polyhedral worlds. However, improvements are still possible and the method in Chapter 6 is a simpler alternative that avoids linear programming and permits a simultaneous test of several labellings of the drawing, among other advantages.

3.1.3 Generic Realizability

Usually, a drawing is only correct for very specific positions of its vertices. This can be illustrated on the truncated tetrahedron of Figure 4.1a. At the depicted configuration the drawing is correct, since the three edge-lines l , m , and n are concurrent to a common point—the apex of the (imaginary) original tetrahedron—but just a slight perturbation of any vertex will destroy the concurrence and yield an incorrect drawing. This characteristic is common to many drawings since their correctness is usually expressible as a set of concurrence conditions on groups of three lines (see Chapter 4).

Sugihara notes that not all drawings are so sensitive to slight perturbations of the coordinates. Certain drawings are always realizable if the vertices lie at generic positions in the plane. The vertices are said to be in a *generic position* when there is no meaningful algebraic relationship between their coordinates. That is, if $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ are the coordinates, seen as variables, there is no polynomial

$$P \in \mathbb{R}[x_1, y_1, x_2, y_2, \dots, x_n, y_n], \quad P \neq 0,$$

such that

$$P(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = 0$$

when evaluated at the current values of the coordinates. (Note that demanding that the vertices are in a generic position is stronger than demanding they are in a general position.)

A drawing is said to be *generically realizable* (or *generically correct*) if it is realizable whenever its vertices are in generic position. A triangulated drawing—one where all faces are triangles—is a trivial example of such drawings.

Sugihara argues, convincingly, that a usual drawing obtained by processing an image or a hand-made sketch will certainly use generic positions for the vertices, since any algebraic relationships among the coordinates will be lost during the digitalization or any further processing. Thus, it would be useful to have a test deciding when a drawing is generically realizable or not. Sugihara realizes that such a test must be strictly combinatorial, only depending on the incidence structure, and in 1982 he gives the following characterization [99]:

Theorem 3.1. *Let $S = (V, F, R)$ be the incidence structure of a drawing \mathcal{D} . Then, \mathcal{D} is generically realizable if, and only if, for any subset $X \subseteq R$ such that $|F(X)| \geq 2$,*

$$|V(X)| + 3|F(X)| \geq |X| + 4, \tag{3.2}$$

where $F(X)$ and $V(X)$ are, respectively, all faces and vertices involved in the incidence pairs of X .

Sugihara proved the theorem for incidence structures of trihedral or convex polyhedra [99], and in 1984 Whiteley extended its validity to arbitrary incidence structures [123]. Although from this theorem it seems that deciding whether a drawing is generically realizable takes $O(2^{|R|})$ time, Sugihara gives a graph-flow algorithm that efficiently checks the conditions in $O(|R|^2)$ time (see [104, Chapter 8]).

3.1.4 Superstrictness

Unfortunately, usual drawings of general scenes are not generically realizable, because their incidence structures rarely satisfy the above combinatorial conditions (3.2). For example, for the truncated tetrahedron in Figure 3.6, and assuming that every vertex is incident to three faces, the counts give:

$$\begin{aligned} |R| &= 18, & |V(R)| &= 6, & F(R) &= 5, \\ |V(R)| + 3|F(R)| &= 21 < 22 = |R| + 4. \end{aligned}$$

This proves that in general it is not possible to decide the realizability only using combinatorial counts, but it has another important consequence: a drawing with vertices in generic positions will always be classified as incorrect by equations (3.1), unless it is

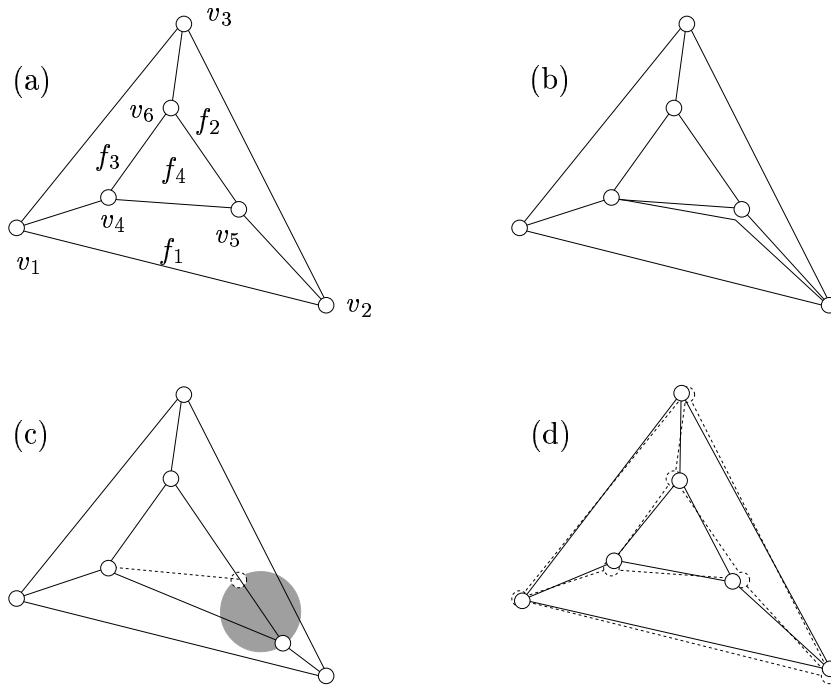


Figure 3.6: Sugihara's correction strategy.

generically realizable, which seldom occurs. This test is said to be *superstrict* in the sense that it classifies “almost correct” drawings as incorrect.

To overcome this difficulty, Sugihara provides a correction strategy that takes an incorrect line drawing \mathcal{D}^{inc} as input and outputs a corrected version of it \mathcal{D}^{cor} [104, Chapter 7]. \mathcal{D}^{cor} has the same incidence structure as \mathcal{D}^{inc} , but its vertex positions have been modified so that it is realizable. Then, although \mathcal{D}^{inc} is incorrect, it is judged as *practically correct* if the vertex positions of \mathcal{D}^{cor} are not too far from the original ones in \mathcal{D}^{inc} .

Roughly speaking, Sugihara's correction method works as follows. Think about the truncated tetrahedron in Figure 3.6a. We see that fixing the heights of v_1, v_2, v_3 and v_4 is enough to determine the heights of the others, as we can use the coplanarity constraint of each face to derive them. However, the height of v_5 is overconstrained, as it can be deduced from *both* the coplanarity of f_1 and f_2 . Only when the projection is correct, this height will be identical when computed from both faces. As Figure 3.6a is incorrect, a possibility is to drop out the constraint that v_5 must lie on f_1 (Figure 3.6b), fix the heights of v_1, v_2, v_3, v_4 , compute the resulting planes for the faces, intersect f_1, f_2, f_4 to get a corrected position for v_5 in 3-space, and project v_5 onto the XY -plane to get a corrected position for it. Summarized, the steps are:

1. Take \mathcal{D}^{inc} with its incidence structure (V, F, R) and remove some incidence pairs (v, f) from R , until a generically realizable drawing appears, with a new incidence set R^* .

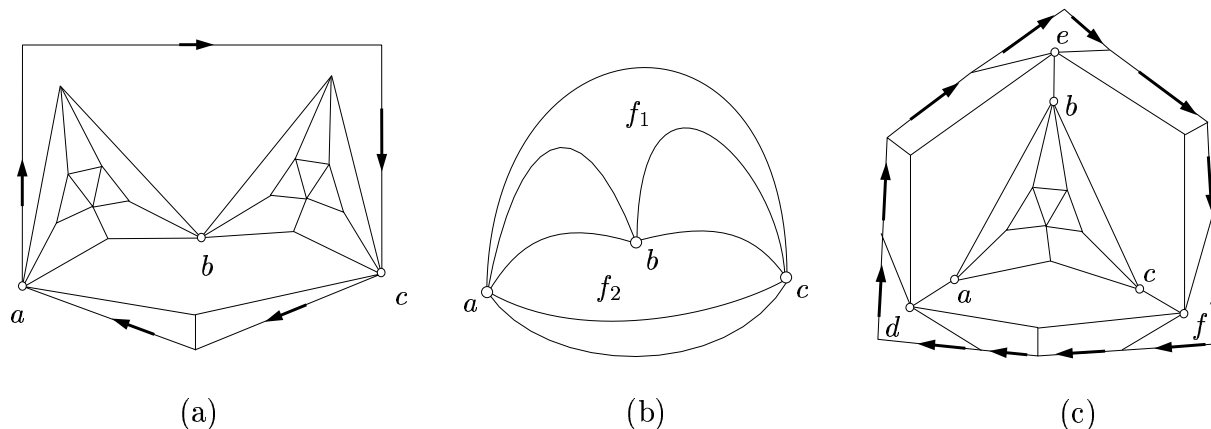


Figure 3.7: Sugihara's correction strategy can not be applied on these examples (adapted from [122]). The arrows indicate occlusive edges.

2. Remove from V all vertices that are involved in some of the removed incidence pairs.
3. Lift the new drawing to 3-space, computing the spatial planes of all faces.
4. Derive new spatial positions for the vertices removed in step 2, by computing the intersections of the planes around them. Project these vertices to the XY -plane.
5. Output the incidence structure of the original drawing, but replace the original vertex coordinates by the corrected ones.

For step 1, Sugihara provides an efficient graph-flow algorithm that removes the least possible number of incidence pairs, finding a maximal generically realizable drawing. For example, in Figure 3.6, it would only remove the incidence pair (v_5, f_1) from R , getting $R^* = R - \{(v_5, f_1)\}$, for which the counts are satisfied. For example, if $X = R^*$:

$$\begin{aligned} |R^*| &= 15, & |V(R^*)| &= 5, & |F(R^*)| &= 5, \\ |V(R^*)| + 3|F(R^*)| &= 20 & \geq & 19 = |R^*| + 4. \end{aligned}$$

However, as already noted by himself, this correction is possible provided that the removed vertices lie on at most three non-triangular faces. (If a vertex lies on more than three non-triangular faces, the intersection of their planes is not a single point in general, and step 4 above cannot be performed.) Unfortunately, as Whiteley notes in [122], one can find drawings where this does not happen. Figure 3.7a shows a superstrict drawing ($|V(X)| + 3|F(X)| < |X| + 4$ for the subset X in Figure 3.7b). Removing one of the vertices a , b or c we get a generically reconstructible drawing, but these are incident with four non-triangular faces. Figure 3.7c shows a second 'bad' example: the corrected drawing must make the lines ad , be , cf concurrent, but all six vertices are incident with at least four non-triangular faces.

Also, another problem of Sugihara's technique is that the corrected drawing may deviate substantially from some original vertices, when, moving all of them just a bit, one can find drawings that fall in a smaller neighbourhood (compare Figures 3.6c and d).

In Chapter 7 we will present an alternative scheme that avoids these two inconveniences: all drawings with the incidence structure of a polydisk will be correctable, and we will allow the movement of all vertices to get correct drawings closer to the input incorrect one.

In 1992, Ponce and Shimshoni propose a different approach to overcome the superstrictness problem [77, 92]. They define a system similar to (3.1) but, unlike Sugihara, they do not eliminate constraints that lead to a superstrict set of equations, but explicitly introduce uncertainty in these constraints. Specifically, they consider that a vertex' true position (x_i, y_i) is unknown in the drawing, but that must fall in a square of side 2ϵ around the measured position $(\tilde{x}_i, \tilde{y}_i)$. Then, they take Sugihara's linear equations (3.1), they perform the change of variables $x_i = \tilde{x}_i + \mu_i$, $y_i = \tilde{y}_i + \nu_i$ for every vertex (x_i, y_i) , and add the constraints $|\mu_i| \leq \epsilon$, $|\nu_i| \leq \epsilon$. This leads to a system of *nonlinear* equalities and inequalities that, after a clever addition of gradient-space constraints, and some algebraic manipulation, they are able to linearize again. The linearization, however, is gained at the cost of the sufficiency of the test and, as they note, the resulting constraints are only necessary for a drawing to be correct. Our drawing correction strategy, though, will not suffer from this drawback (see Chapter 7).

3.2 Results from Structural Geometry

3.2.1 Maxwell's Theorem

Given a drawing \mathcal{D} with the incidence structure of a spherical polyhedron, we can also build \mathcal{D} as a plane bar-and-joint framework \mathcal{F}^{bj} , putting a universal joint for each vertex and a rigid bar for each edge. A *self-stress* on \mathcal{F}^{bj} is an assignment of a force to every bar of \mathcal{F}^{bj} so that every vertex is in equilibrium —i.e., the sum of all forces from bars incident to the vertex is zero. The self-stress is *strict* if all bars receive non-null forces. Now, we can ask two very different questions:

- Are the vertices well positioned so that \mathcal{D} can be lifted to 3-space as a correct spherical polyhedron with plane faces and non-zero dihedral angles at the edges?
- Does \mathcal{F}^{bj} support a strict self-stress?

Surprisingly, both questions lead to identical answers: a drawing is a picture of a spherical polyhedron if, and only if, the framework supports such a self-stress. This two-way connection was already discovered (but not completely proved) more than a century ago by James Clerk Maxwell, when seeking a graphical tool to calculate the internal forces of a bar-and-joint framework in equilibrium and avoid the explicit use of systems of linear equations [63, 64]. He found that such a tool exists, in the form of a *reciprocal figure*, which is constructed as follows. First, a framework is abstractly seen as a diagram of *vertices* (the joints), joined in pairs by straight *edges* (the bars), and dividing the plane into several *regions* (the polygonal areas). The reciprocal figure is also a diagram of vertices and edges:

it has a vertex for each region of the framework, and an edge joining two vertices if their corresponding regions are adjacent. Hence, every edge between two vertices, say v_1 and v_2 , of the reciprocal diagram corresponds to an edge of the original: the edge separating the regions corresponding to v_1 and v_2 . The reciprocal diagram must accomplish the additional property that each of its edges must be orthogonal to its corresponding edge in the original framework. Figure 3.8a shows the framework of a tetrahedron and one of its reciprocals, overlaid.

Maxwell proves that a reciprocal figure can be constructed if, and only if, there exists an assignment of forces to the bars of the framework that keep it in equilibrium —i.e., a self-stress. This can be graphically seen in Figure 3.8 b, c and d. Every edge of the reciprocal represents a force on the corresponding bar of \mathcal{F}^{bj} and since all bars incident to a joint of \mathcal{F}^{bj} have a corresponding closed polygon of forces in the reciprocal, by the parallelogram rule all vertices are in equilibrium.

The diagram is called “reciprocal” because exactly the same property holds when the roles of “framework” and “reciprocal diagram” are interchanged: if either diagram is taken as representing the framework, the edges of the other diagram will represent a self-stress. However, Maxwell went one step further and proved the following:

Theorem 3.2 (Maxwell, 1864). *If a drawing \mathcal{D} represents the plane projection of a spherical polyhedron where all adjacent faces have different planes, then its framework \mathcal{F}^{bj} has a strict self-stress (and, hence, a reciprocal diagram).*

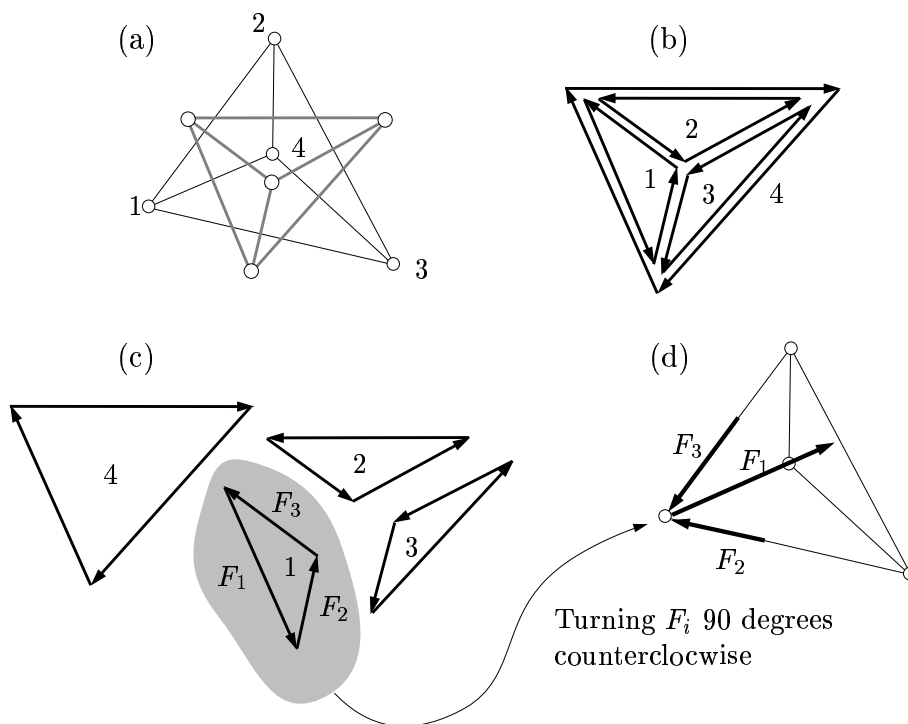


Figure 3.8: Maxwell's reciprocal.

The converse result was conjectured to be true and even accepted in some texts, but it was not until 1982 that Whiteley gave a formal proof (see Theorem 3.3 below). This large time-span between both proofs is probably symptomatic of the evolution of Structural Mechanics and related areas like Kinematics and Geometry along the 20th Century: with the advent of computers, problems were gradually approached via large-scale numerical calculations, usually disregarding the wealth of applicable results in many 19th century treatises on these matters [106, 22, 3]. The situation is gradually changing though, after recognising that Geometry is essential to many human activities and it is so deeply embodied in how humans think [126].

3.2.2 Structural Geometry

Although for half a century the reciprocal diagram was the method of choice when studying the static behaviour of frameworks (for example, it was extensively used to design the Eiffel tower), Maxwell's theorem remained completely unnoticed until a Canadian group working on Structural Topology realized its value. Inspired by the conjectures of a structural engineer, Janos Baracs, this group has been working since the late seventies to construct a new theoretic body known as *Structural Geometry*, a new geometry that unifies projective geometry with structural engineering. This field has developed out of one fundamental problem: which frameworks of rigid bars and flexible joints are *infinitesimally rigid* (i.e., no assignment of velocities to the joints infinitesimally deforms the structure) or *statically rigid* (the framework is able to absorb all external forces in equilibrium)?

The genesis of the group took place in the 1974-1977 period [19]. During those years Baracs was able to pass on his interests to an interdisciplinary group of geometers, architects and engineers. Some of the regular members included Walter Whiteley, Henry Crapo, Ethan Bolker, Ben Roth, Neil White, Anton Kotzig and Tiong Seng Tay. Baracs was arguing that it should be possible to use a *single* two-dimensional projection to determine both the spatial realizability of a drawing, and the static and kinematic properties of its associated bar-and-joint framework. Clearly, Baracs was mentioning Maxwell's theorem without knowing it.

Stimulated by this conjecture, the group found a treatise on Graphical Statics by Luigi Cremona in 1975 [22], which contained all relevant references to the literature of the 1864-1888 period, from the discovery of the reciprocal figures by Maxwell and his colleague Rankine, to the establishment of a science of "Graphical Statics" by the German engineer Culmann. With this, they quickly found Maxwell's papers in the initial volumes of the *Philosophical Transactions* [63] and the *Royal Society Proceedings* [64]. Maxwell's theorem was then rediscovered and used as the starting point that confirmed Baracs' suspects.

After realising that the rigidity of a framework is a projectively invariant property —if a framework is rigid, then a projective transformation of itself is rigid too—, it became clear that the most suitable language to describe the static and kinematic behaviour of frameworks was projective geometry [20]. Then, it was necessary to reformulate the whole theories of statics of forces and kinematics of rigid bodies with this new vision in mind.

Several articles collect this formulation, along the period 1977-1982 [119, 120, 20, 121]. At an algebraic level, it seemed that the ideal tool for symbolic manipulation was the Grassmann-Cayley algebra, developed by Doubilet, Rota and Stein in the early seventies [23].

Using the Grassmann-Cayley algebra, forces and moments could be modelled in a homogeneous and compact way. A force f applied to a point p can be represented as a 2-extensor $F = f \vee p$ in this algebra. Also, a couple of two forces f and $-f$, respectively acting on points p and q , is represented by a 2-extensor $P = f \vee (p - q)$. Surprisingly, a couple turns out to be just a special kind of force: one acting on a point at infinity. Hence, the equations of static equilibrium can be summarized saying that a system of n forces $\{F_1, \dots, F_n\}$, each seen as a 2-extensor $F_i = f_i \vee p_i$, is in equilibrium if and only if $\sum F_i = 0$, and the usual separate sums of forces and moments are here reduced to a single expression.

The kinematics of rigid bodies also received a similar treatment. The velocity $M(p)$ of a point p on a body can be expressed as $M(p) = S \vee p$, where S is the 2-extensor $S = a \vee b$, built up from two projective points a and b on the instantaneous screw center of the body. In this way, translational velocities are a particular case of rotational ones, since a point p under translation is actually rotating about a screw center at infinity, represented by a 2-extensor of two improper points $S_\infty = a_\infty \vee b_\infty$, with velocity $M(p) = S_\infty \vee p$.

Thanks to this formulation, it was possible to relate the static behaviour of a bar-and-joint framework \mathcal{F}^{bj} associated with the skeleton of a spherical polyhedron \mathcal{P} —where each edge corresponds to a rigid bar and each vertex to a universal joint— with the kinematics of an associated panel-and-hinge framework \mathcal{F}^{ph} —with a rigid panel for each face of \mathcal{P} , and a hinge articulating two panels if their faces are adjacent in \mathcal{P} . Indeed, in [20] Crapo and Whiteley see that \mathcal{F}^{ph} has an infinitesimal motion if, and only if, \mathcal{F}^{bj} has a self-stress, and this was the fundamental piece that led Whiteley to prove the converse of Maxwell's theorem [121]:

Theorem 3.3 (Whiteley, 1982). *Let \mathcal{F}^{bj} be a bar-and-joint framework with the combinatorial structure of a spherical polyhedron. Let \mathcal{D} be the corresponding drawing. \mathcal{D} is realizable as a spherical polyhedron with different planes for every two adjacent faces if, and only if, \mathcal{F}^{bj} has a strict self-stress (i.e., with non-null forces on all bars).*

Note that this theorem gives a criterion to tell whether a drawing is realizable: we just need to check whether the associated bar-and-joint framework has a self-stress and this can be done either algebraically, or by constructing a reciprocal diagram. This technique, however, has two drawbacks. On the one hand, as the theorem reads, the test is only valid for drawings with the incidence structure of a spherical polyhedron. On the other hand, the test is superstrict in the sense given in Section 3.1.4. For example, we clearly see that while a correct truncated tetrahedron has a proper reciprocal diagram (Figure 3.9c), a slight perturbation of it does not (Figure 3.9f).

It must be noted that the use of Maxwell's reciprocal as a test for realizability has been rediscovered and repeatedly used by the Machine Vision community. For example, Mackworth in 1973 [54], Huffman in 1977 [44], and Draper in 1981 [25] use this tool under

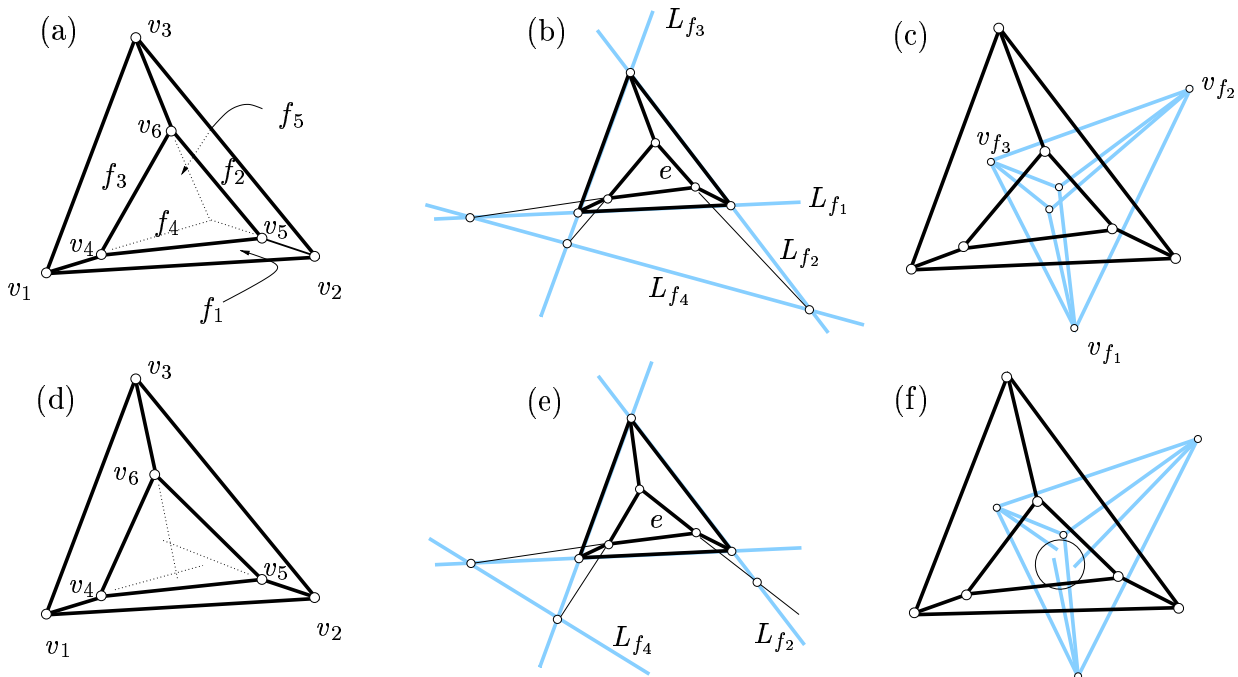


Figure 3.9: A correct truncated tetrahedron (a), with two compatible diagrams: the cross-section (b), and Maxwell's reciprocal (c). If v_6 is slightly moved (d), the diagrams are not compatible anymore (e and f).

the name of a *dual diagram* embedded in a gradient space. Some of these authors insist that the existence of a reciprocal figure is only a necessary condition for realizability, whereas from Theorem 3.3 we now know that it is also sufficient when the drawing has the structure of a spherical polyhedron. We observe a general confusion in the Machine Vision literature were the sufficiency in such cases is seldom recognized or even negated.

3.2.3 The Cross-Section test

In 1991, Whiteley presents another reciprocal diagram known as the *cross-section* [124]². Although in Chapter 4 we will explain it in full detail, let us roughly anticipate how it works.

Like Maxwell's, the cross-section reciprocal offers necessary and sufficient conditions for correct drawings of spherical polyhedra: the drawing is correct if, and only if, it is possible to draw a cross-section "compatible" with it. The cross-section has a straight line L_{f_i} corresponding to every face f_i of the drawing. Every line L_{f_i} must be interpreted as the intersection of the face plane for f_i with an arbitrary *cutting* plane. The cross-section is *compatible* if the line of any edge between a pair of faces contains the point of intersection of the cross-section lines of these faces. Figure 3.9b depicts a cross-section (in bold grey) for a correct truncated tetrahedron. In this example, to simplify, the background face f_5 is taken as the cutting plane.

²Actually, this diagram was already announced in 1987 [122], but the full development appears in [124].

The cross-section reciprocal is an elegant tool that graphically allows to test the correctness of a drawing by just verifying concurrence conditions of lines on it. However, there are three difficulties that must be overcome before it can be thoroughly applied:

1. First, in its original form the test can only be used on drawings of spherical polyhedra, while we saw in Section 2.2 that most scenes of opaque objects generate drawings of polydisks, possibly with holes.
2. On the other hand, as recognized by Whiteley himself “*in spite of the long history of graphical statics, including numerous engineering textbooks, no classical or modern theory of reciprocals has solved the fundamental problem of algorithms to generate a reciprocal or to show that no reciprocal exists. The absence of general algorithms is a fundamental difficulty in using such geometric methods on computers*” [122, page 447].
3. Finally, like Maxwell’s, this is a superstrict test. This can be seen in Figure 3.9e: slight perturbations of the vertices destroy the compatibility of the cross-section with the drawing.

We proposed a way around the first drawback in [81] (see Chapter 4) where we extended the validity of the cross-section test including the case of polydisks with any number of holes. This extension is fully explained in Chapter 4. The second difficulty is solved in Chapter 6, where an algorithm is given to generate all cross-sections, if there are any. Finally, the superstrictness can be overcome with our methods [82] and [78]. These are fully explained in Chapters 7 and 8.

3.2.4 Algebraic Geometry

The algebraic side of structural geometry was also studied using Invariant Theory [117, 118]. For example, in 1983 White and Whiteley find a polynomial condition that must be verified by the vertex coordinates of an isostatic bar-and-joint framework³ so that it can have a self-stress in equilibrium (see [117] and the related works [118] and [73]). This polynomial is called the *pure condition*, and can be compactly written using bracket algebra. For example, the pure condition that guarantees the existence of a self-stress on a truncated tetrahedron is

$$[abc][a'b'c']([abb'][a'c'c] - [a'bb'][ac'c]) = 0, \quad (3.3)$$

where a, b, c , and a', b', c' are the vertices of the two triangles, and a *bracket* $[pqr]$ represents the 3×3 determinant of the vertices p, q and r , all written in homogeneous coordinates. Now, thanks to Theorem 3.3 above, if an isostatic framework has the combinatorial structure of a spherical polyhedron, and accomplishes the pure condition, it has a self-stress. If the self-stress is strict, the corresponding drawing is realizable.

³A framework is *isostatic* if it is infinitesimally rigid, and the deletion of any arbitrary bar makes the bar infinitesimally flexible [73, 119].

The pure condition alone, however, does not tell whether the self-stress is strict —there could be bars with null forces— and has the additional drawback that it has only been developed for isostatic frameworks, which limits its use to drawings where the number of edges is exactly $2v - 3$. However, the potential applications of this formalism are clear from the work by Crapo [18], where he obtains the bracket polynomials that define the correctness of several line drawings.

Although a pure condition is a bracket expression difficult to interpret in terms of synthetic geometric conditions, it is possible to apply a Cayley factorization algorithm to translate it into a Grassmann-Cayley expression involving joins and meets [112, 97]. If we do this for the polynomial (3.3), we get:

$$(a \vee b \vee c) \cdot (a' \vee b' \vee c') \cdot ((a \vee a') \wedge (b \vee b') \wedge (c \vee c')) = 0, \quad (3.4)$$

This makes the condition directly interpretable, since every factor is a smaller condition with a known geometric meaning (see the tables in [116]). For example, Equation 3.4 reads:

“the bar-and-joint framework of a truncated tetrahedron has a self-stress if, and only if, either the triangles a, b, c and a', b', c' are collinear, or they are perspective from a common point”.

In 1991, White gave a Cayley factorization algorithm for the multilinear case [113], i.e., when every point of the bracket polynomial occurs exactly once in each monomial. This algorithm is practical up to about 20 points. Although no practical algorithm is known for Cayley factorization in general, this is a promising tool in the field of Geometric Reasoning, since, as it has been done in this example, it might be possible to recover the geometric meaning of any bracket-polynomial expression as (3.3), in terms of meaningful sentences of natural language. Other potential applications include Automated Geometric Theorem-Proving and the detection of singular configurations of mechanisms [114, 116].

Part II

Interpretation of Drawings

Chapter 4

Synthetic Geometric Tools

“The knowledge at which geometry aims is the knowledge of the eternal”.

Plato —The Republic, Book VII, 427-347 BC.

“Plato seems to have foreseen, on the basis of what must have been very sparse evidence indeed at that time, that: on the one hand, mathematics must be studied and understood for its own sake, and one must not demand completely accurate applicability to the objects of physical experience; on the other hand, the workings of the actual external world can ultimately be understood only in terms of precise mathematics”.

Roger Penrose —The Emperor’s New Mind, Chapter 5, 1991 [75].

Using Structural Geometry, Whiteley showed that a line drawing is a correct projection of a spherical polyhedron if and only if it has a cross-section compatible with it [124]. We here enlarge the class of drawings to which this test applies, including those of polydisks and polydisks with holes. This extension is necessary in order to deal with usual scenes of opaque polyhedra, where occluded regions are not shown (see Section 2.2). Our proof is constructive, showing how to derive all spatial interpretations, and it relies on elementary synthetic geometric arguments. Also, as a by-product, it yields a simpler and shorter proof of Whiteley’s result. Moreover, important properties of line drawings are visually derived as corollaries: realizability is independent of the adopted projection, it is an invariant projective property, and for trihedral drawings it can be checked with pencil and straightedge¹ alone. Whiteley’s proof is very valuable from the standpoint of Structural Geometry, as it clearly shows the connections of the realizability problem with that of finding a self-stress—a pattern of forces in equilibrium—on a corresponding bar framework, called a *grillage*. Although our proof loses this connection, it is highly visual, intuitive and simple.

¹A straightedge is a ruler without distance marks.

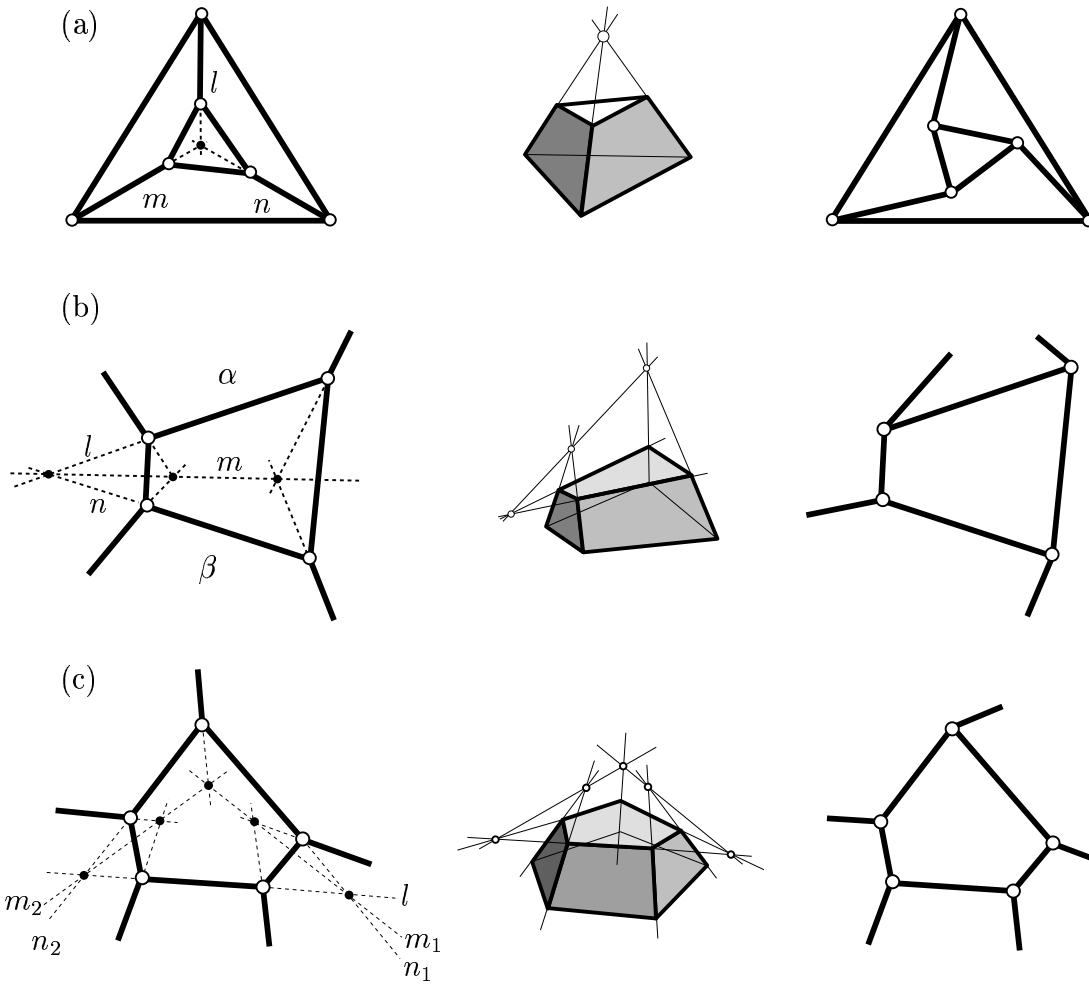


Figure 4.1: To be correct, the drawings to the left must verify the indicated concurrence conditions, since they hold on any of their spatial interpretations (center). This conditions allow to classify the drawings to the right as incorrect.

4.1 The Cross-Section Test

It is well known that the problem of deciding whether a line drawing is *realizable* was solved by Sugihara in his series of papers [103, 102, 101], where he reduced it to an instance of linear programming. What is less known, however, is that for spherical polyhedra the correctness can be decided by only checking the concurrence of groups of three lines derived from the drawing itself. Although we independently proved this in [79], we later found that a cleaner version of the result had already appeared in [124], due to Whiteley.

Some examples suggest that a drawing's correctness is checkeable using only concurrence conditions. The truncated tetrahedron in Figure 4.1a is only correct when its three edges l , m , and n meet at a common point. The *4-calotte* in Figure 4.1b, a configuration of a quadrilateral face and its four neighboring faces, is only correct when the three lines l, m, n are concurrent or, equivalently, when the three bold points are aligned, since

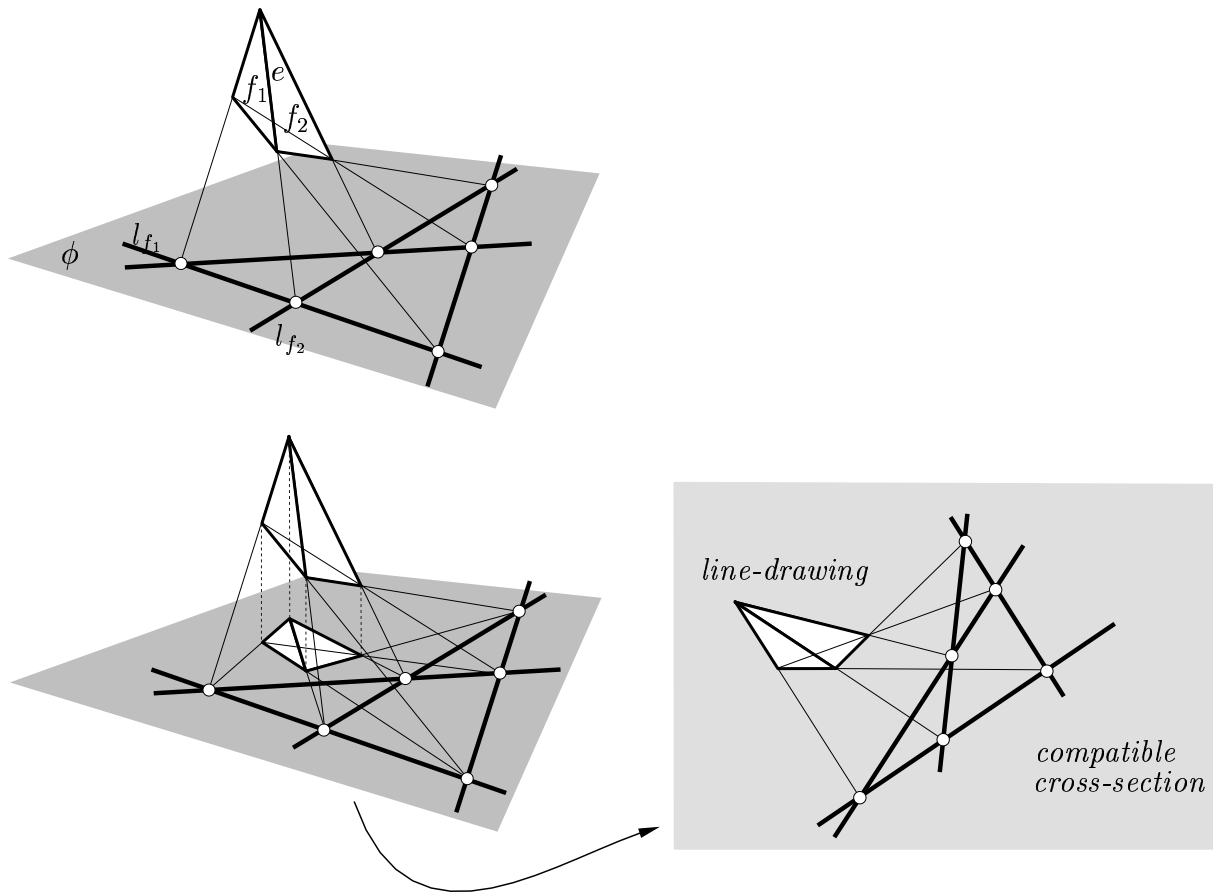


Figure 4.2: The cross-section test.

they all lie at the line m of intersection of the planes α and β . Likewise, the *5-calotte* in Figure 4.1c is only correct when l , m_1 , n_1 , and l , m_2 , n_2 are concurrent too. These are all necessary conditions for realizability, since they hold in any spatial reconstruction of the drawing and projection preserves all incidence relations between points and lines (Figure 4.1, center). These conditions are useful to discard some drawings as incorrect (Figure 4.1, right), but the challenge was to characterize a set of also *sufficient* concurrence conditions of realizability. We next present one.

Consider a spherical polyhedron in 3-space such as, for example, the tetrahedron in Figure 4.2, top. Now, obtain the intersections of the planes of its faces with an external plane ϕ in general position. The resulting arrangement of lines is called a *cross-section* of the polyhedron. It is clear that the edge-line e between two faces, say f_1 and f_2 , must be concurrent to the point of intersection of the lines l_{f_1} and l_{f_2} of intersection of the planes of f_1 and f_2 with ϕ . These trivial concurrence conditions in 3-space will clearly hold too when projecting the whole construction onto the plane ϕ , because projection preserves collinearity of points and all incidence relations (Figure 4.2, central). Hence, we have a set of concurrence constraints that are necessary for a drawing to correctly represent the projection of a spherical polyhedron: the drawing can only be correct if we are able to draw a *compatible* cross-section diagram, one where these concurrences hold (Figure 4.2,

bottom). Whiteley's theorem states that the converse is also true. We will also prove the following.

Theorem 4.1. *A line drawing of a polydisk is realizable if and only if it has a compatible cross-section, with non-coincident cross-section lines.*

By “compatible” we mean here that all non-boundary edges of the polydisk are concurrent to their corresponding point in the cross-section, leaving the boundary ones unconstrained.

For the proof, we need the following definition. On a polydisk \mathcal{P} in 3-space, we say that a face f of \mathcal{P} is *sequentially adjacent* to faces $f_{i_1}, f_{i_2}, \dots, f_{i_m}$ of \mathcal{P} if the edges between f and these faces are *sequentially linked*, meaning that if (f, f_{i_k}) denotes the edge between faces f and f_{i_k} , then in the sequence $s = \{(f, f_{i_1}), (f, f_{i_2}), \dots, (f, f_{i_m})\}$ every edge shares a vertex with the previous and the next one, except for the first and last edges in s , which only share a vertex with the next and the previous edge, respectively.

Proof. The “only if” part is proven by the above arguments. We will prove the “if” part using the drawing and its compatible cross-section to explicitly construct a polydisk, one whose face-planes generate the given cross-section when intersected with the plane of the drawing.

The proof proceeds by induction. First we show how to lift a first face of the drawing keeping all its vertices coplanar. Then we assume that a polydisk \mathcal{L}_k with k faces has already been lifted correctly and prove that any other face that is sequentially adjacent to some faces in \mathcal{L}_k can be properly lifted too, to form a lifted polydisk \mathcal{L}_{k+1} with $k + 1$ faces.

To start with, take any of the faces of the drawing and consider its cross-section line. A lifting of this face can be fixed by giving an arbitrary height to any one of its vertices not in the cross-section line. The cross-section line and the lifted vertex define the plane of the face, and all edges and vertices of the face are then lifted vertically to lie on this plane.

As induction hypothesis, assume that we have a polydisk \mathcal{L}_k already lifted correctly, where for every edge between two faces, its line meets the point where the cross-section lines of these two faces intersect. This applies to all edges of \mathcal{L}_k , even those at its boundary. Note that this hypothesis is true if \mathcal{L}_k only contains the first lifted face.

Now, we can lift an additional face f_0 , not in \mathcal{L}_k , that is sequentially adjacent to m faces f_1, \dots, f_m of \mathcal{L}_k (and to no other face of \mathcal{L}_k) through m edges PQ, QR, RS, \dots (see Figure 4.3, where we represent the case $m = 3$). We will prove that these edges and the cross-section line l_{f_0} of f_0 are all coplanar, and define a plane α that is different from all planes assigned to the faces f_1, \dots, f_m .

To see this, note that, since the edge-line PQ and l_{f_0} are incident (by induction hypothesis), these two lines are coplanar. Let us call α the plane they define. Moreover, the edge-line QR is also coplanar with α , as it contains two points of this plane: the point where it intersects with l_{f_0} and the point Q of line PQ . The same applies to line RS as it

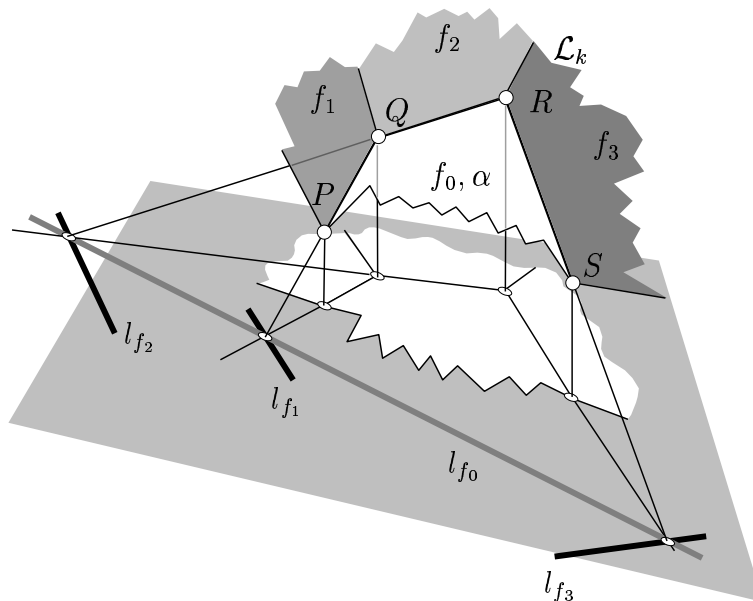


Figure 4.3: Lifting an intermediate face.

is incident with R and l_{f_0} . Clearly, the argument can be iterated to prove that all other edges between f_0 and \mathcal{L}_k are coplanar with α .

With the plane for f_0 already fixed, all other edges between f_0 and faces not in \mathcal{L}_k can be fixed too by lifting them vertically to lie in this plane. For every such edge, say between face f_0 and face f_i , we must prove that its line of support meets the point T where the cross-section lines l_{f_i} and l_{f_0} meet. Clearly, this line is defined by the intersection of the plane α and a vertical plane containing the projection of the edge. But both planes meet T : α meets T because T is a point of l_{f_0} , and the vertical plane meets T because the projection of the edge meets T in the cross-section.

It remains to prove that the plane α is different from all the planes given to the faces f_1, \dots, f_m . But this is trivially true, as the only way for α to coincide with one of such planes would be that the corresponding cross-section lines were identical, which is not the case, by the premises of the theorem. \square

Corollary 4.1 (Whiteley's theorem). *A line drawing of a spherical polyhedron is realizable if, and only if, it has a compatible cross-section with non-coincident cross-section lines.*

Proof. The spherical polyhedron can be constructed by generating a sequence of lifted polydisks $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \dots$, adding one face at a time. At the end, a last face will close the polyhedron, but all its edges will be coplanar, as they all are sequentially linked with edges of the previous polydisk (Figure 4.4). \square

Corollary 4.2. *Realizability does not depend on the type of projection assumed.*

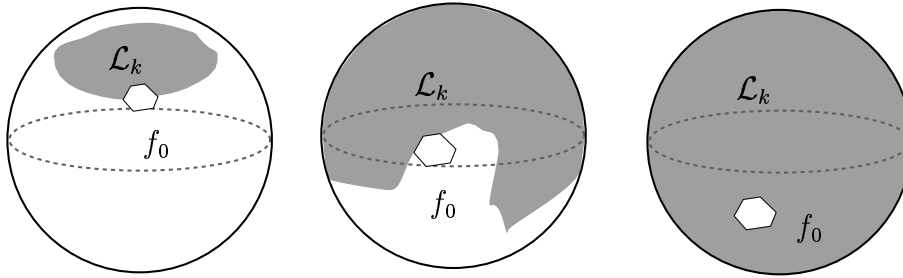


Figure 4.4: Evolution of \mathcal{L}_k .

Proof. If we observe Figure 4.3 we see that the same proof works for central projection, the only difference being that the vertices P, Q, R, \dots , are lifted along lines that meet at the center of projection. Similarly, the proof is valid for oblique parallel projection. \square

Corollary 4.3. *Realizability is a projectively invariant property.*

Proof. Non-singular projective transformations of the plane map lines to lines and points of intersection of two lines to the points of intersection of the transformed two lines [51]. Hence, if a drawing has a compatible cross-section, the transformed drawing will also have one. \square

4.2 Realizability for Other Topologies

So far, the sufficiency of the cross-section test has been proved for drawings of polyhedral surfaces homeomorphic to a disk (Theorem 4.1) or a sphere (Corollary 4.1). Can we extend the test to surfaces of other topologies? The following considerations will depict its full range of applicability. They are summarized in Table 4.1.

First, note that there is no trouble in considering surfaces having self-penetrations, or with self-intersecting faces (Figure 4.5). As long as the underlying combinatorial structure has the topology of a disk or a sphere, these intersections or penetrations do not change any of the proofs given above, as they do not produce any new faces, vertices, or edges. From now on, we will use the topology of the combinatorial structure, rather than that of the spatial object itself.

At a topologic level, an orientable surface S can be fully characterized by two quantities: the number μ of its boundary curves, and the number p of closed cuts that do not separate it into parts—also called the *genus* of S . In fact, a necessary and sufficient condition that two orientable surfaces are homeomorphic is that these two numbers shall be the same for both surfaces [51, page 106].

Note that, if $C_{\mu,p}$ denotes the class of all orientable homeomorphic surfaces with genus p and μ boundaries, what we have done so far is to prove the validity of the cross-section test for surfaces in the classes $C_{0,0}$ and $C_{1,0}$.

For polyhedral surfaces in $C_{\mu,0}, \mu \geq 2$, i.e., polydisks “with holes”, the cross-section test cannot directly be applied. For such objects, the proof of Theorem 4.1 would fail, as their drawings cannot be lifted by subsequently adding faces that are sequentially adjacent to a previous polydisk. See this, e.g., for a topologic disk with a hole in Figure 4.6a. As a counterexample, consider three pairwise adjacent faces with a triangular hole (Figure 4.6b): this drawing always has a compatible cross-section, but it is not realizable unless the three non-boundary edges, l, m and n , are concurrent. However, a useful modification allows cross-sections to be used even in such cases.

Proposition 4.1. *If \mathcal{D}_1 is a drawing of a polydisk \mathcal{P} with holes, we can triangulate every hole to produce a new drawing \mathcal{D}_2 whose incidence structure has the topology of a disk. Then \mathcal{D}_1 is realizable if and only if \mathcal{D}_2 is realizable.*

Proof. If \mathcal{D}_1 is realizable, it has at least one lifting \mathcal{L}_1 and a realization of \mathcal{D}_2 can be found by spatially triangulating the holes of \mathcal{L}_1 . Conversely, a realization of \mathcal{D}_1 can be found from one of \mathcal{D}_2 by removing the triangles that cover the holes. This argument also







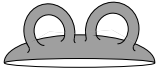
<i>Class</i>	μ (n. boundaries)	p (genus)	<i>Example</i>	<i>Applicability of cross-sections</i>
$C_{0,0}$	0	0		Yes, with Whiteley's theorem
$C_{\mu,0}$ $\mu \geq 1$	1	0		Yes, with Theorem 4.1 Yes, after triangulating $\mu - 1$ holes
	2	0		
	
$C_{0,p}$ $p \geq 1$	0	1		No
	0	2		
	
$C_{\mu,p}$ $\mu \geq 1$ $p \geq 1$	1	1		No, but reducible to $C_{0,p}$, triangulating the boundaries
	1	2		
	

Table 4.1: Applicability of cross-sections for several topologies.

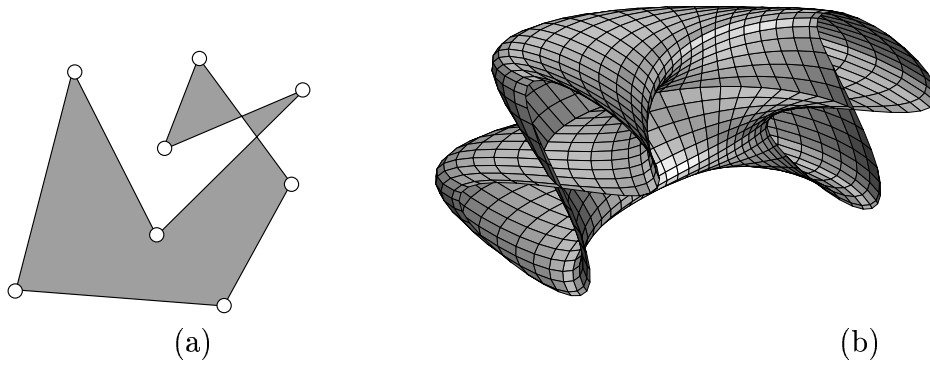


Figure 4.5: (a) Self intersecting face. (b) Self-penetrating polysurface.

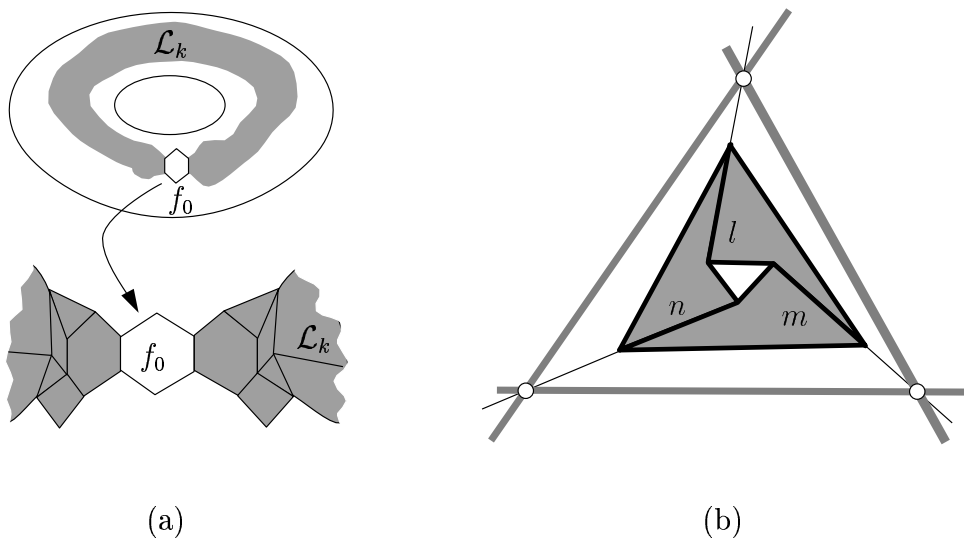


Figure 4.6: The cross-section test is invalid for non-simply connected objects. (a) If the polysurface has holes, then there will be some face f_0 that is not sequentially adjacent to a previous polydisk \mathcal{L}_k . (b) A simple incorrect drawing with a compatible cross-section.

shows that, actually, there is a one-to-one correspondence between the liftings of \mathcal{D}_1 and those of \mathcal{D}_2 . \square

In sum, the cross-section test can be used for objects in the class $C_{\mu,0}$, $\mu \geq 2$, as long as they are converted to the class $C_{1,0}$, by the triangulation of $\mu - 1$ boundaries.

No synthetic geometric test using cross-sections has been found for the rest of topologic objects in $C_{0,p}$, $p \geq 1, \mu \geq 0$. Actually, Crapo and Whiteley prove in [21] that the cross-section test is not valid for checking drawings of a prismatic torus, made up of three triangular prisms glued together in pairs through their bases. They show a compatible cross-section that does not correspond to a correct spatial lifting of this torus.

4.3 Graphical Construction of a Cross-Section

4.3.1 Trihedral Polyspheres and Polydisks

Although the existence of a compatible cross-section is a necessary and sufficient condition for realizability, we still need some graphical process to find all compatible cross-sections or otherwise show that none exists. For drawings of trihedral polyspheres or polydisks we give the following method, called the *incremental construction*.

First, note that when lifting the vertices of any correct drawing, one can always choose the heights of four vertices independently, namely, those of the two vertices of an edge, and the height of one vertex in each of the adjacent faces of this edge. Thus, these two faces can receive arbitrary planes, provided that their intersection line projects onto their edge. This means that, when constructing a cross-section, the lines of two adjacent faces can be chosen with arbitrary orientation, as long as they are different and concurrent to the same point of their common edge-line. For drawings of trihedral polyspheres and polydisks these two initial lines completely determine the rest of the cross-section. We see this in Figure 4.7a, where, after fixing lines l_1 and l_2 for faces f_1 and f_2 , l_3 can be automatically deduced as it must contain P and Q , the points where the edge-lines between f_3 and f_1 , and f_3 and f_2 meet with the cross-section lines of f_1 and f_2 , respectively. We can obtain

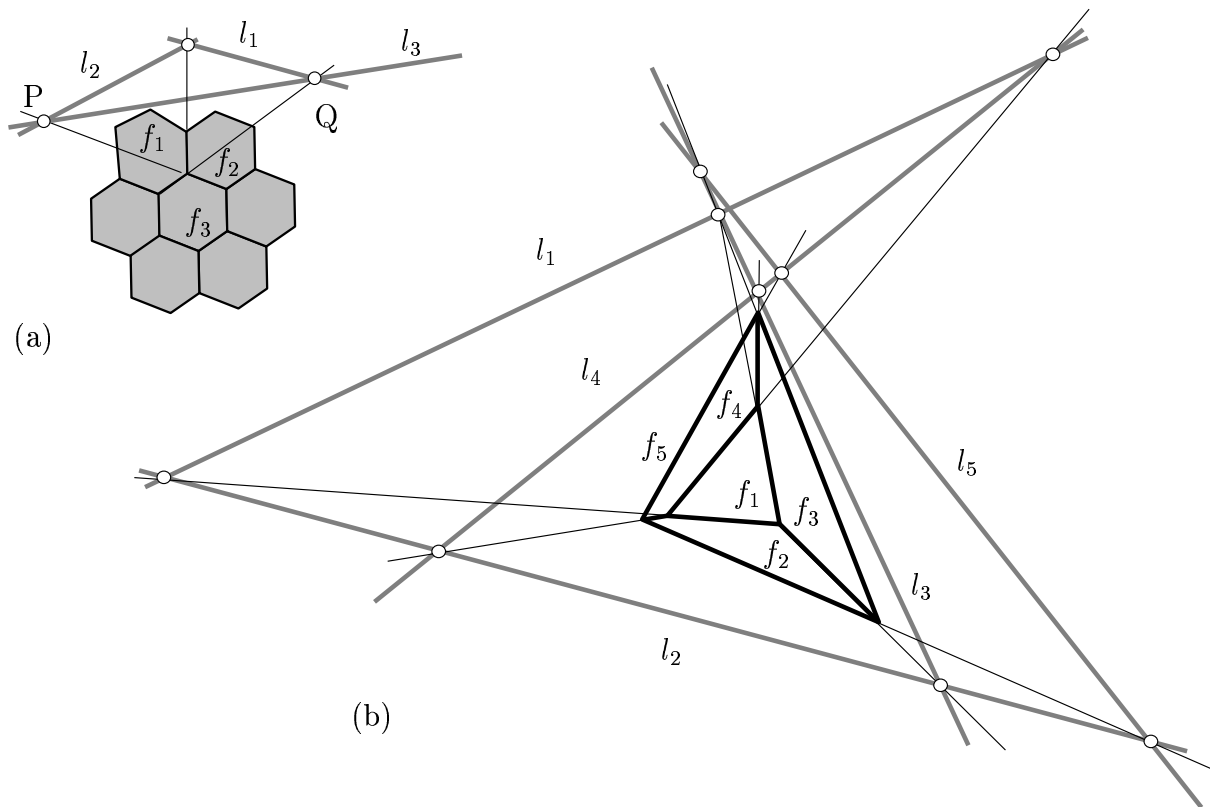


Figure 4.7: Testing a trihedral drawing with pencil and straightedge.

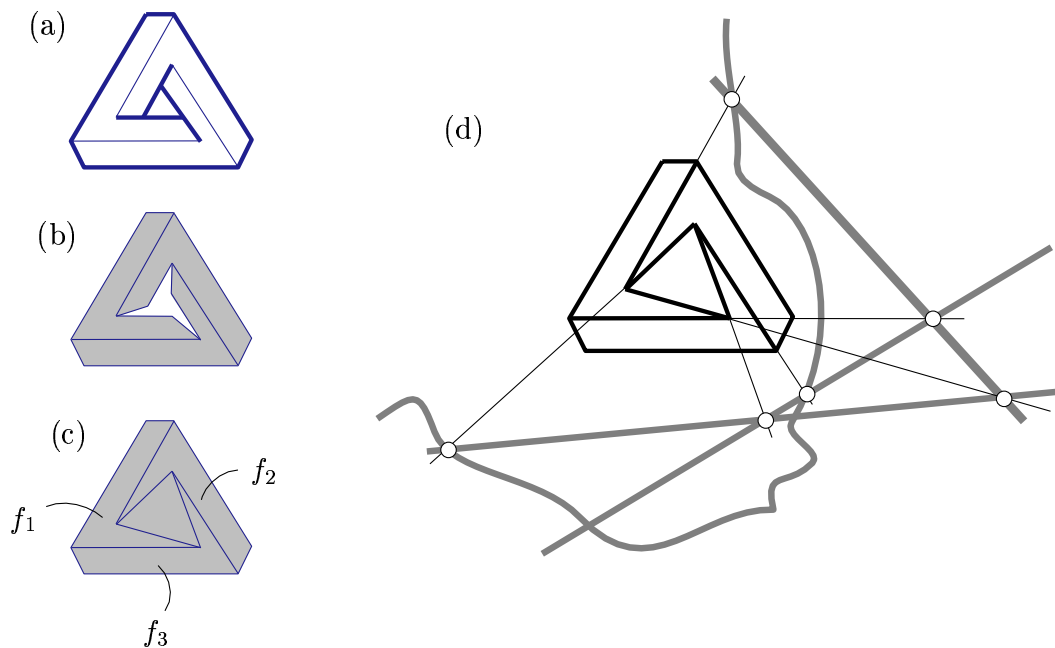


Figure 4.8: Checking Penrose's triangle.

lines for the other faces iterating this process, and construct the cross-section. If at some point a concurrence condition does not hold, we conclude that the drawing is incorrect. If we complete the construction, then it is correct by Corollary 4.1. Clearly, this algorithm takes $O(f)$ time, since it adds one cross-section line at a time, and each line is determined by two previously obtained points. This proves the following.

Proposition 4.2. *Realizability of drawings of trihedral polyspheres and polydisks can be checked with pencil and straightedge in $O(f)$ time.*

Figure 4.7b illustrates the incremental construction on a truncated tetrahedron. We consider the outer triangle as a face too and, hence, the incidence structure is that of a polysphere. We start fixing l_1 and l_2 , with arbitrary orientation, making them concurrent to the edge-line between faces f_1 and f_2 . The rest of lines are then fixed in this order: l_3 , l_4 , l_5 .

As another example, we employ the incremental construction on Penrose's triangle (Figure 4.8). A usual labelling of the triangle marks the bold edges in Figure 4.8a as boundary, yielding a drawing of a polydisk with one hole (Figure 4.8b). Note that we can cover the hole with a triangle to get the trihedral polydisk of Figure 4.8c. This does not alter the realizability of the original drawing: any realization of Figure 4.8b can be transformed to one of Figure 4.8c by the simple addition of this triangle and properly removing the 2-valent boundary vertices. Conversely, any realization of Figure 4.8b can be obtained from one of Figure 4.8c by removing the triangle and properly extending faces f_1 , f_2 and f_3 . If we now check this polydisk, we conclude that it is incorrect, since the three black points on the "weavy line" of Figure 4.8d are not aligned.

On trihedral drawings it is easy to generate the whole family of cross-sections. To this

end, we just need to launch the incremental construction repeatedly, each time starting from two different initial lines, and draw the resulting diagrams. Figures 4.9 and 4.10 illustrate this on a regular dodecahedron.

A graphical reconstruction is also possible on trihedral drawings. Recall that a lifting is fully determined after choosing a specific cross-section and the height of one vertex. To view all liftings we repeatedly apply the reconstruction method of Theorem 4.1 while simultaneously sweeping the cross-section lines and varying the height of a selected vertex. Let us do this for the tetrahedron and the starting cross-section depicted in Figure 4.11. Keeping this cross-section and moving P upwards, we have the liftings in Figure 4.12, top. We obtain other liftings by keeping P fixed, but moving one of the three points Q , R and S that determine the two initial lines of the cross-section (Figures 4.12 and 4.13).

4.3.2 Trihedral Polydisks with Holes

In a general situation the depicted polydisks will have holes in them. This may happen when we deal with scenes of several opaque polyhedra, with some objects occluding others behind. For these topologies, we saw that the existence of a cross-section is merely a necessary condition of realizability and cannot be used as a complete test, unless we triangulate the holes. However, the following gives two cases where the cross-section is still a valid tool.

Proposition 4.3. *Let \mathcal{D} be a drawing of a polydisk with holes. If each hole has not more than four boundary edges, the correctness of \mathcal{D} can be checked with pencil and straightedge.*

Proof. The trick is to construct “trihedral roofs” over the holes, to get a derived drawing with the topology of a polydisk. Consider a triangular hole like the one in Figure 4.14a. Note that we can remove the hole by extending its three faces f_1 , f_2 and f_3 , until the common point of intersection of the edge-lines e_1 , e_2 and e_3 . If these lines do not concur on a common point we can reject the drawing as incorrect. If they concur we get the transformed drawing in Figure 4.14b, that retains the same realizability properties as the original. In other words, the original drawing is correct, if and only if the transformed one is. The new drawing is trihedral, it has the topology of a polydisk and by Proposition 4.2 it can be checked with pencil and straightedge alone.

If the holes are quadrilateral, we proceed as in Figure 4.14c. The point P of intersection of the faces f_1 , f_2 and f_4 must lie on the intersection of the edge-lines e_1 and e_2 . Analogously the point where f_2 , f_3 and f_4 meet is Q , on the intersection of e_3 and e_4 . Clearly, PQ is the line of intersection of f_2 with f_4 . The hole is covered by extending f_1 , f_2 , f_3 and f_4 until they hit the dotted lines. The resulting drawing is trihedral (Figure 4.14d) and it is realizable if, and only if, the original one is. \square

Unfortunately, the construction of trihedral roofs is not possible on holes with more than four boundary edges, because the intersection lines between pairs of faces in these roofs cannot be determined from the edge-lines incident to the hole.

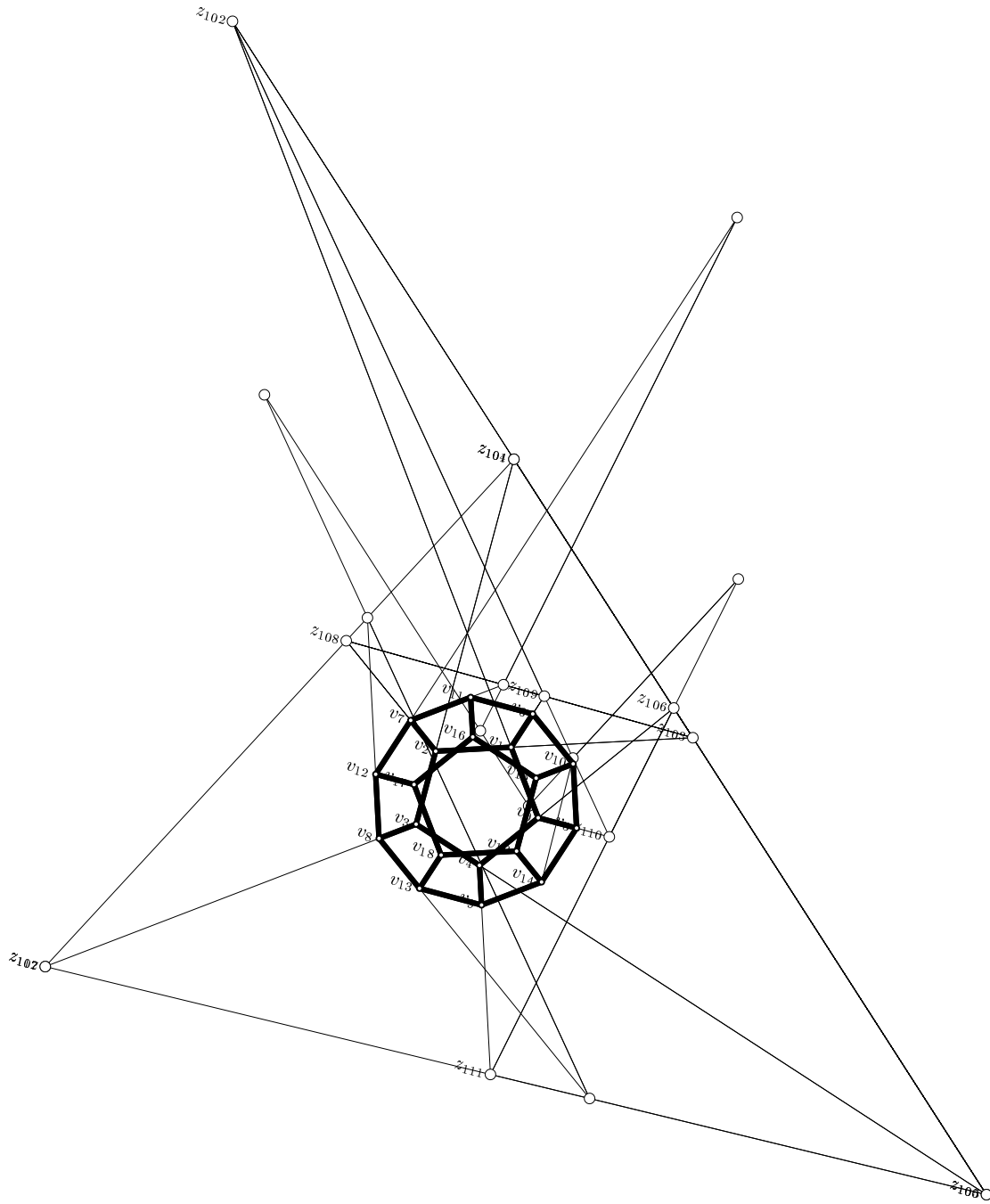


Figure 4.9: A cross-section of a dodecahedron.

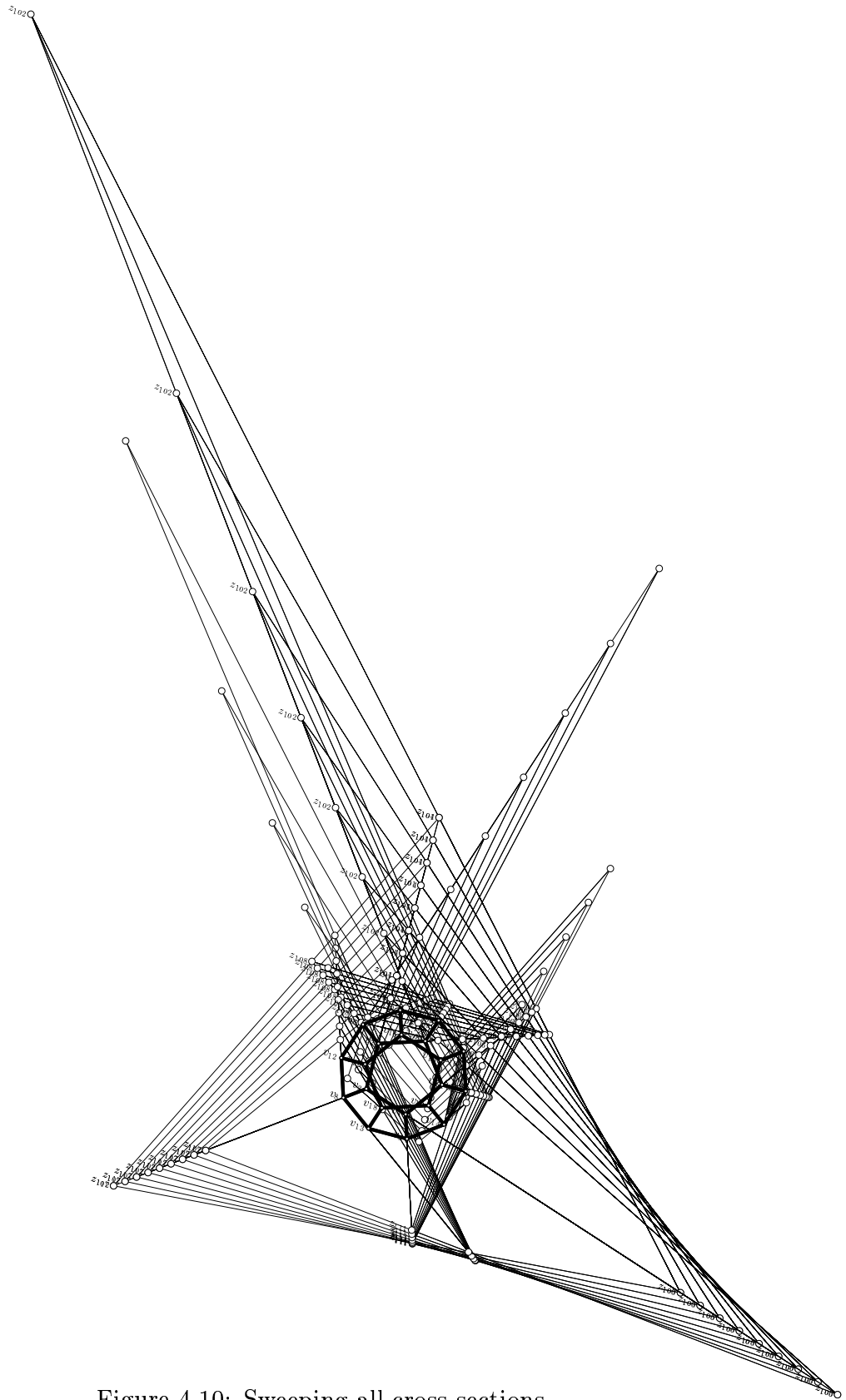


Figure 4.10: Sweeping all cross-sections.

4.3.3 General Drawings

We say that a line drawing has a *determined cross-section* when, after the initial choice of two lines, the remaining ones are fully determined. The incremental construction is possible on drawings of trihedral polydisks and polyspheres precisely because they yield this type of cross-sections. Actually, it would be useful to characterize the whole class of drawings with determined cross-sections, to delimit the full range of applicability of the incremental construction. Note that this class is not only restricted to drawings of trihedral polydisks and polyspheres. The reader can check, for example, that the drawings in Figure 4.15 also exhibit this property, but have vertices with more than three incident faces. Up to our knowledge, this characterization has not been pursued yet and remains an interesting open problem.

If we try to apply the incremental construction on a general drawing, it may happen that an intermediate line remains undetermined. If this is the case, we can always choose one among all possible positions for this line, and continue with the construction. If we are able to complete the cross-section, the drawing is certainly correct. If we do not succeed though, we cannot reject it as incorrect, since other locations for the undetermined lines might still yield a compatible cross-section. Obviously, we could check all possibilities in a generate-and-test fashion with backtracking, but this is clearly non-viable.

Unfortunately, no general method has been devised to generate all compatible cross-sections, or otherwise show that none exists, using pencil and straightedge alone. This constitutes another important and challenging open problem (if solvable at all). On the contrary, if algebraic tools are allowed an algorithm does exist. Before we present it in Chapter 6, we need the tools of Grassmann-Cayley algebra and Instantaneous Kinematics, which are next introduced.

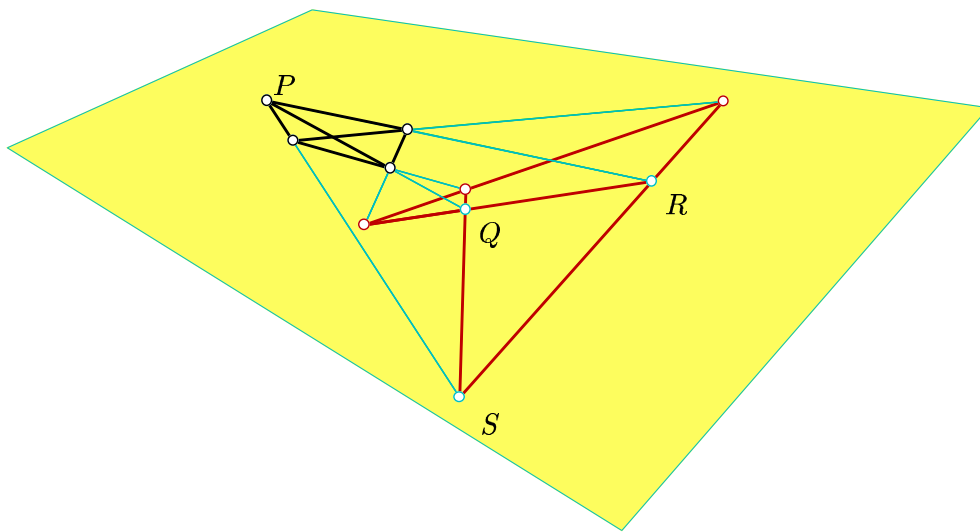


Figure 4.11: A tetrahedron with a compatible cross-section, lying on the XY -plane.

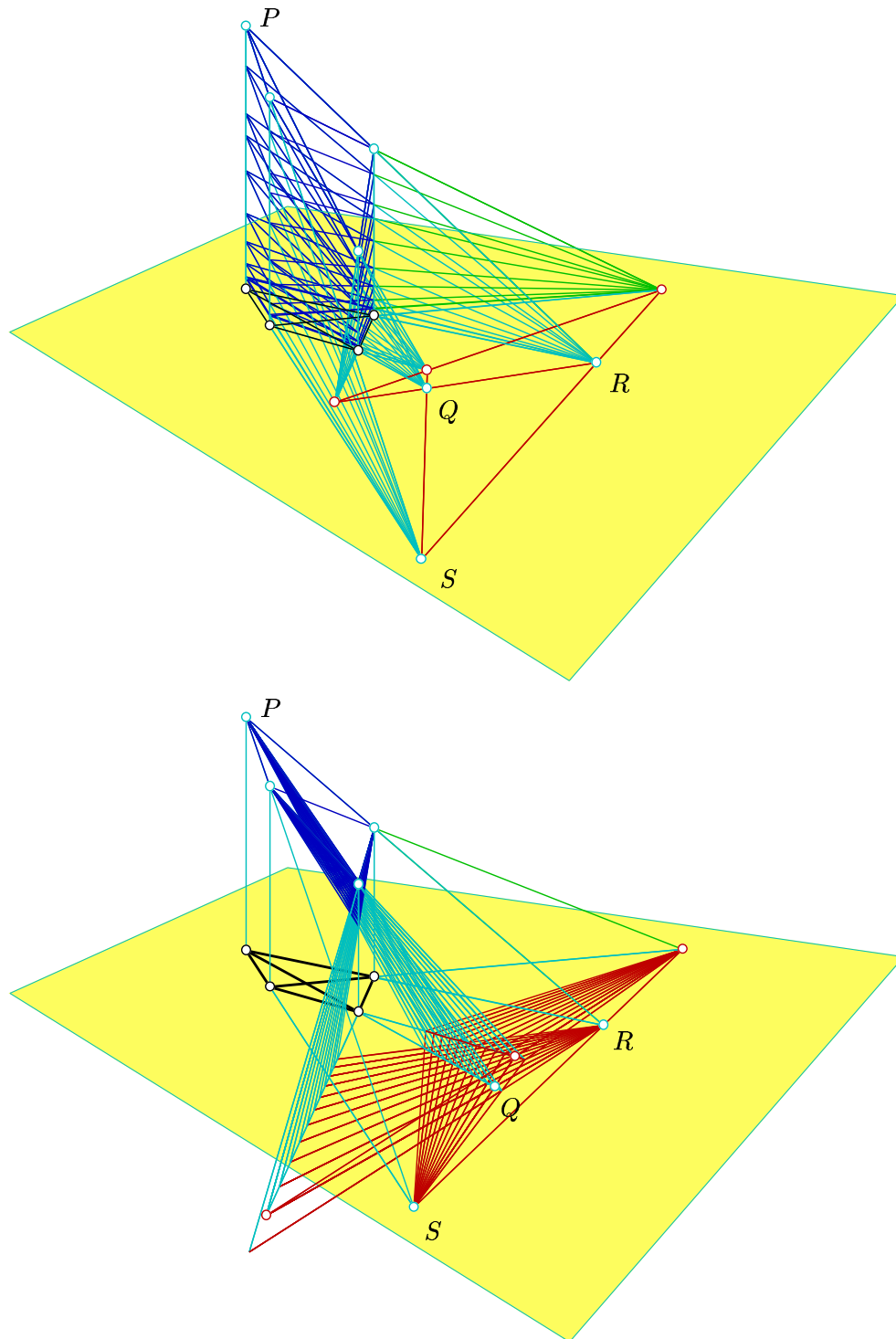


Figure 4.12: Top: moving P to generate the liftings with a fixed cross-section. Bottom: liftings with P fixed but a variable cross-section obtained by moving Q .

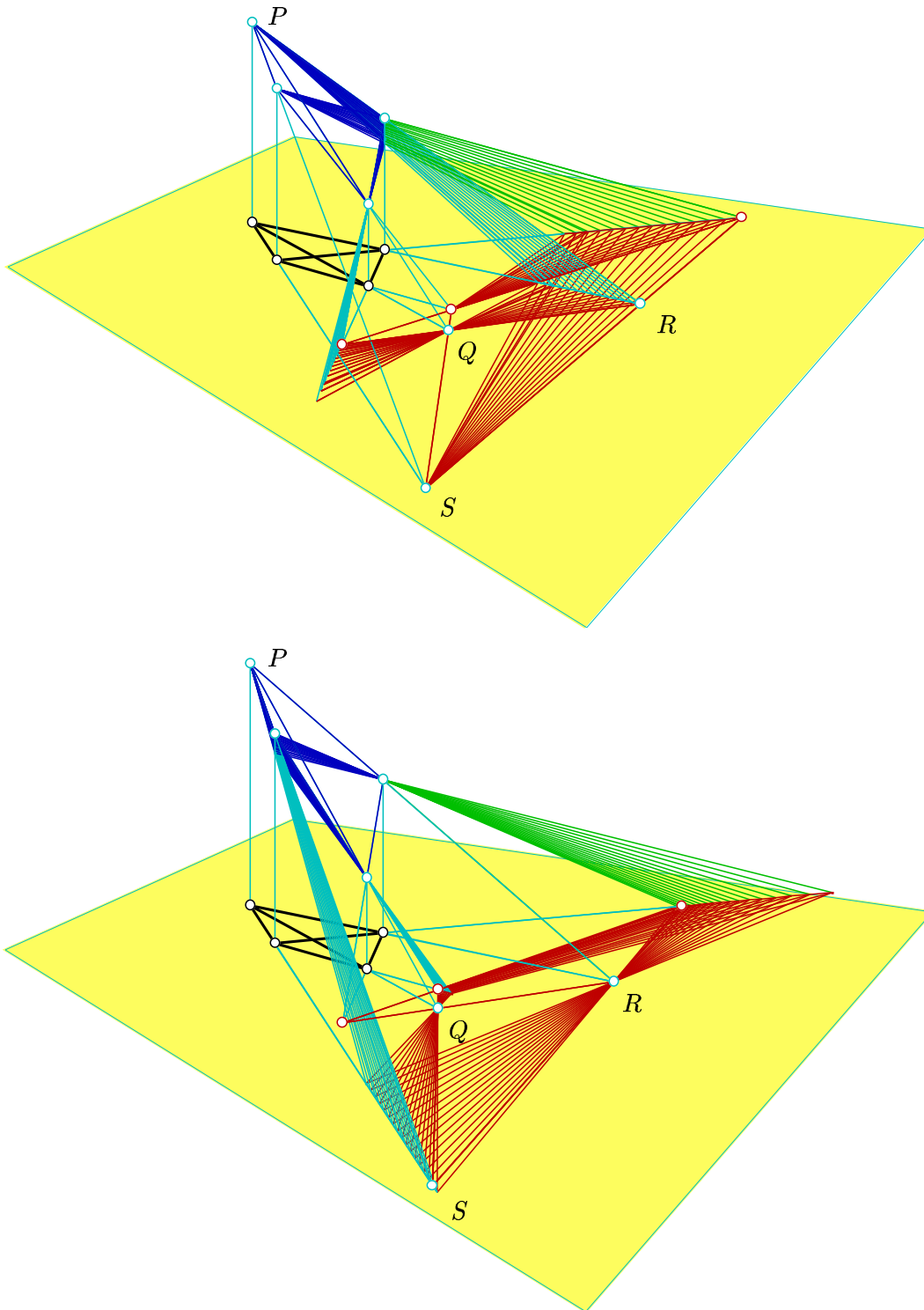


Figure 4.13: Sweeping R (top) and S (bottom) with P fixed.

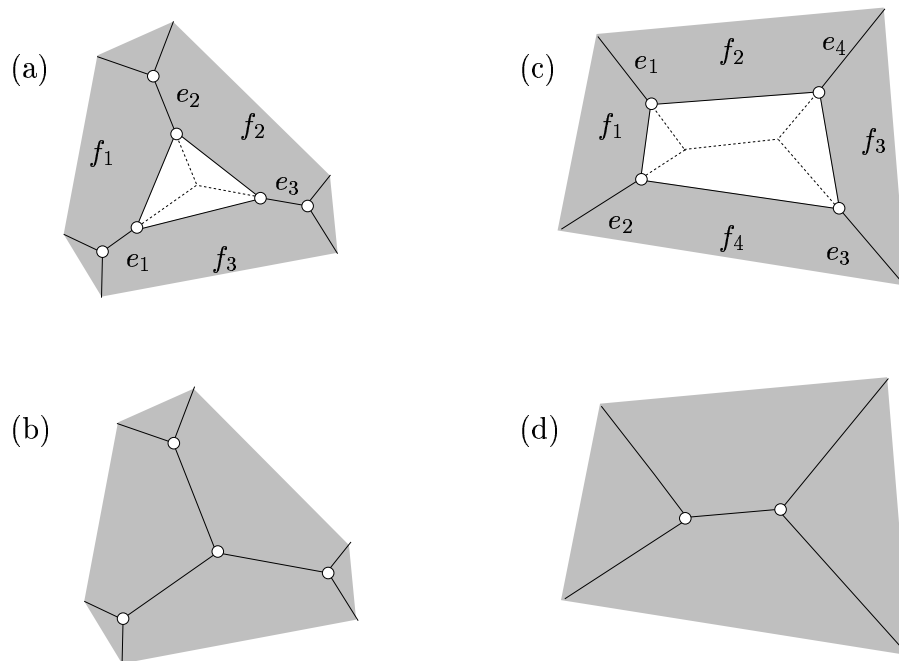


Figure 4.14: Testing polydisks with holes.

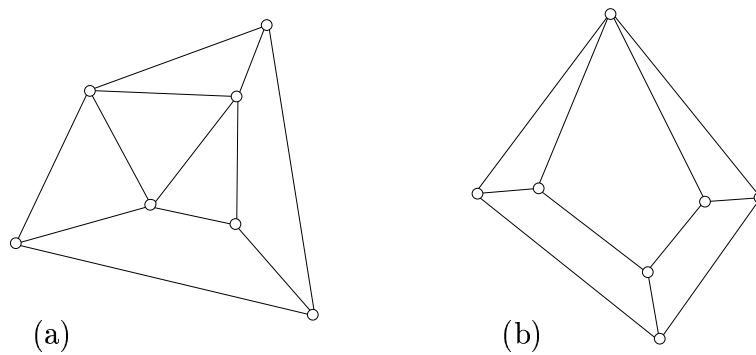


Figure 4.15: Non-trihedral drawings with determined cross-sections. (a) is generically realizable, while (b) is not.

Chapter 5

Kinematics in the Grassmann-Cayley Algebra

*“As I tell my students, algebra is cosmetics, not surgery:
change the appearance but not the substance”.*

*Walter Whiteley —The Decline and Rise of Geometry
in 20th Century North America, 1999 [126].*

To properly understand the connections of line drawing realizability with the kinematics of frameworks we need some algebraic tools. As already seen in the previous chapter, the realizability of a drawing is a projective property: if a drawing is realizable, then any non-singular projective transformation of it provides another realizable drawing. In a similar fashion, although it is not generally emphasized in mechanical engineering courses and textbooks, many kinematic properties of a mechanism are also projective. There is a suitable modern language to algebraically analyse these properties, the Grassmann-Cayley Algebra, developed during the seventies by Rota, Doubilet and Stein [23], and later popularized by the Structural Topology Group in Canada [19, 119, 120, 121].

In this chapter, we will see how to represent velocities using this algebra, translating their usual Euclidean treatment to a projective setting. To begin with, we recall several basic concepts about projective spaces and Plücker coordinates, then we review a concrete version of the Grassmann-Cayley Algebra, based on this coordinates. The reader is referred to [23, 115] for a more exhaustive introduction to the topic, and to [119, 120, 114, 116, 95] for several applications of the algebra, including statics and kinematics of mechanisms and robotic systems. For the sake of conciseness we will often present the theorems without a full proof. This will be done when the proof does not provide essential insight for further developments. In such cases, however, we provide references to the literature for further details.

5.1 Projective Spaces

Although we will primarily work in the 3-dimensional space, it is useful to consider an arbitrary m -dimensional space \mathbb{R}^m . For our application, a projective space can be basically viewed as a means of representing (coordinatizing) *all* points of \mathbb{R}^m , even those that are infinitely distant from the origin. To this end, if (x_1, x_2, \dots, x_m) are the *Cartesian coordinates* of a point in \mathbb{R}^m , the vector $(x_1, x_2, \dots, x_m, 1)$ is defined as its *homogeneous coordinates*. If λ is a non-zero scalar, the vector $\lambda(x_1, x_2, \dots, x_m, 1) = (\lambda x_1, \lambda x_2, \dots, \lambda x_m, \lambda)$ is said to define the same point. If we also allow $(m + 1)$ -tuples with the last coordinate equal to 0, which we regard as representing *points at infinity*, then we have the standard construction of the m -dimensional projective space \mathcal{P}^m . With this, we have actually represented \mathbb{R}^m with some extra points thrown in by an $(m + 1)$ -dimensional vector space \mathbb{V} , in such a way that we have the following correspondences:

points of \mathbb{R}^m	\leftrightarrow	1-dimensional subspaces of \mathbb{V}
lines of \mathbb{R}^m	\leftrightarrow	2-dimensional subspaces of \mathbb{V}
planes of \mathbb{R}^m	\leftrightarrow	3-dimensional subspaces of \mathbb{V}
$(k - 1)$ -dimensional affine subspaces of \mathbb{R}^m	\leftrightarrow	k -dimensional subspaces of \mathbb{V}

We call \mathbb{V} the *associated vector-space* of the m -dimensional projective space \mathcal{P}^m . The points of \mathcal{P}^m whose last homogeneous coordinate is different from zero are called *proper points*. The points at infinity are also called *improper points*.

One can easily see the relationship between \mathcal{P}^m and \mathbb{V} for the case $m = 2$ using the construction in Figure 5.1. We think of \mathcal{P}^2 as the plane $z = 1$ of \mathbb{R}^3 . A finite point of \mathcal{P}^2 is represented by a straight line l through the origin O , not contained in the XY plane of \mathbb{R}^3 . All points on this line represent the same Euclidean point: the point where

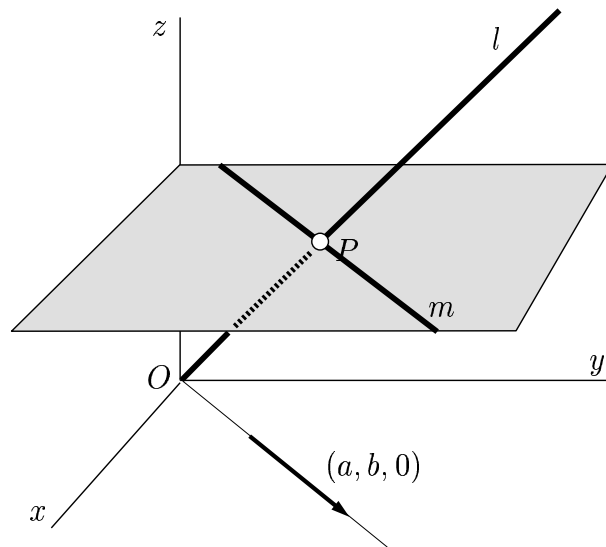


Figure 5.1: The relationship of \mathcal{P}^2 and $\mathbb{V} = \mathbb{R}^3$.

l intersects with $z = 1$. Assume we move l , keeping its incidence with O and with a line m of $z = 1$: we get different Euclidean points P over m . As l approaches the XY plane, the corresponding Euclidean point moves to infinity towards the direction of the vector $(a, b, 0)$. When, eventually, l lies on the plane $z = 0$, l is said to represent the point at infinity in the direction (a, b) .

The following theorem provides another way of “visualizing” a projective space: we can think of it as a sphere that has been folded upon itself by identifying antipodal points.

Theorem 5.1. *The space \mathcal{P}^m is topologically equivalent to the unit sphere \mathcal{S}^m of \mathbb{R}^{m+1} in which antipodal points have been identified.*

The proof can be found in many textbooks on Algebraic Topology (see for example [32]), but we can develop an intuition about what is going on as follows. A point \mathbf{x} of \mathcal{S}^m is represented by a vector $(x_1, x_2, \dots, x_{m+1})$ such that $\sum_{i=1}^{m+1} x_i^2 = 1$. This also represents a point of \mathcal{P}^m . The vector $-\mathbf{x}$ represents the antipodal point of x , which is also on \mathcal{S}^m , but represents the *same* point of \mathcal{P}^m .

5.2 Plücker Coordinates

Let U be a $(k - 1)$ -dimensional affine subspace of \mathbb{R}^m . U corresponds to a k -dimensional vector subspace \mathbb{U} of \mathbb{V} , the associated vector-space of \mathcal{P}^m . Let $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$ be a basis of \mathbb{U} . When these vectors are arranged as rows of a matrix, we obtain:

$$\underline{\mathbf{P}} = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,m} & 1 \\ u_{2,1} & u_{2,2} & \cdots & u_{2,m} & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{k,1} & u_{k,2} & \cdots & u_{k,m} & 1 \end{pmatrix}.$$

The (j_1, j_2, \dots, j_k) -th *Plücker coordinate* of the subspace \mathbb{U} , denoted by $\underline{\mathbf{P}}_{j_1, j_2, \dots, j_k}$, is the $k \times k$ determinant obtained from the k columns of the matrix $\underline{\mathbf{P}}$ with indices j_1, j_2, \dots, j_k . Since we have a Plücker coordinate for each combination of the k columns, the total number of Plücker coordinates is $\binom{m+1}{k}$.

Definition 5.1 (Plücker coordinate vector). The *Plücker coordinate vector* $\mathbf{P}_{\mathbb{U}}$ of \mathbb{U} is the $\binom{m+1}{k}$ -tuple obtained by listing all Plücker coordinates of \mathbb{U} in some predetermined order. Since \mathbb{U} is associated with the affine subspace U , we feel free to write \mathbf{P}_U instead of $\mathbf{P}_{\mathbb{U}}$, when needed.

We next give three examples of Plücker coordinate vectors (or *Plücker vectors*, for short) that are thoroughly used in what follows. The Plücker vector of a proper line l in 3-space will be used to represent the rotational motion of a rigid body \mathcal{B} around l . The Plücker vector of an improper line will represent a pure translational motion of \mathcal{B} and, finally, the Plücker vector of a plane will represent the velocity of a specific point of \mathcal{B} . Details are given in Sections 5.4 and 5.5.

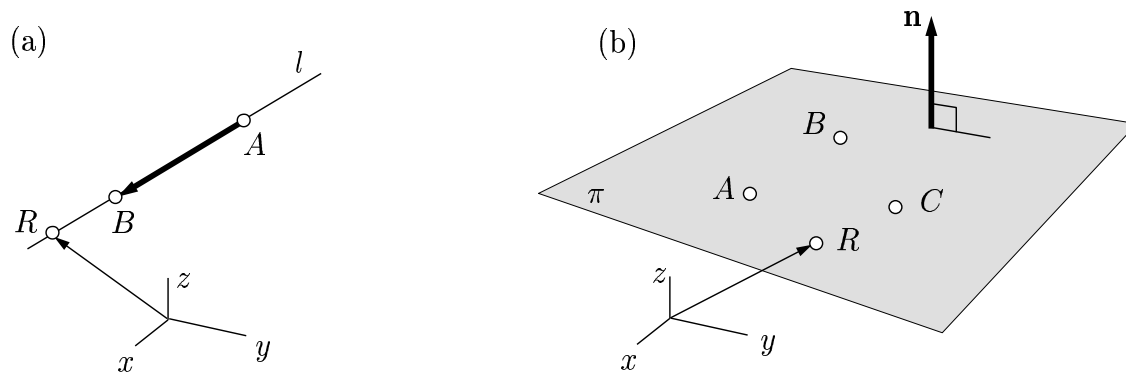


Figure 5.2: Interpretation of the Plücker vectors of a line (a) and a plane (b).

5.2.1 Proper Lines in 3-Space

The Plücker coordinate vector of the line l along two proper points \mathbf{a} and \mathbf{b} , represented in homogeneous coordinates by the rows of

$$\underline{\mathbf{P}} = \begin{pmatrix} a_1 & a_2 & a_3 & 1 \\ b_1 & b_2 & b_3 & 1 \end{pmatrix},$$

is defined as:

$$\begin{aligned} \underline{\mathbf{P}}_l &= (\underline{\mathbf{P}}_{1,4}, \underline{\mathbf{P}}_{2,4}, \underline{\mathbf{P}}_{3,4}, \underline{\mathbf{P}}_{2,3}, \underline{\mathbf{P}}_{3,1}, \underline{\mathbf{P}}_{1,2}) = \\ &= (b_1 - a_1, b_2 - a_2, b_3 - a_3, a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1). \end{aligned}$$

Note that the first three components of $\underline{\mathbf{P}}_l$ are the coordinates of the vector $\mathbf{s} = \mathbf{b} - \mathbf{a}$, and that the three last ones represent the moment¹ of this vector with respect to the origin, $\mathbf{a} \times \mathbf{s}$. See Figure 5.2a.

The point at infinity on l is the homogeneous coordinate vector $(b_1 - a_1, b_2 - a_2, b_3 - a_3, 0)$ and its scalar multiples. It can be thought of as infinitely far away in the direction given by \mathbf{s} . The *same* point at infinity lies on every line parallel to l , but non-parallel lines have distinct points at infinity. Notice that $\underline{\mathbf{P}}_l$ may be computed from any two distinct points on l , including the point at infinity.

5.2.2 Lines at Infinity

A line at infinity is determined by two distinct points at infinity, say c and d , represented by the rows of

$$\underline{\mathbf{P}} = \begin{pmatrix} c_1 & c_2 & c_3 & 0 \\ d_1 & d_2 & d_3 & 0 \end{pmatrix},$$

¹The engineering convention is to compute the moment of a line-bound vector \mathbf{s} about the origin as $\mathbf{s} \times \mathbf{r}$, where \mathbf{r} is any point on the line. We here take the opposite convention, computing the moment as $\mathbf{r} \times \mathbf{s}$.

and its Plücker coordinate vector is

$$\underline{\mathbf{P}}_l = (0, 0, 0, c_2d_3 - c_3d_2, c_3d_1 - c_1d_3, c_1d_2 - c_2d_1).$$

These lines will be used to model translational velocities of a rigid body as rotations about an axis at infinity.

5.2.3 Planes in 3-Space

The Plücker coordinate vector of a plane π , determined by three finite points whose homogeneous coordinates are the rows of

$$\underline{\mathbf{P}} = \begin{pmatrix} a_1 & a_2 & a_3 & 1 \\ b_1 & b_2 & b_3 & 1 \\ p_1 & p_2 & p_3 & 1 \end{pmatrix},$$

is defined as

$$\underline{\mathbf{P}}_\pi = (\underline{\mathbf{P}}_{2,3,4}, -\underline{\mathbf{P}}_{1,3,4}, \underline{\mathbf{P}}_{1,2,4}, \underline{\mathbf{P}}_{1,2,3}) = (\mathbf{n}, \mathbf{r} \cdot \mathbf{n}),$$

where \mathbf{n} is a vector orthogonal to π and \mathbf{r} is the position vector of any point on π . See Figure 5.2b.

5.2.4 Coordinatizations of Affine Subspaces

The following theorem guarantees that the Plücker vector of a subspace totally characterizes this subspace [114]. Thus, Plücker coordinates really offer a way to coordinatize vector spaces of \mathbb{V} , and their associated counterparts, the affine subspaces of \mathbb{R}^m .

Theorem 5.2. *The vector subspace \mathbb{U} uniquely determines $\mathbf{P}_\mathbb{U}$ up to a scalar multiple. Moreover, if \mathbf{P} is a $\binom{m+1}{k}$ -tuple of real numbers and if $\mathbf{P} = \mathbf{P}_\mathbb{U}$ for some subspace \mathbb{U} , then \mathbf{P} uniquely determines \mathbb{U} .*

Proof. To verify that \mathbb{U} uniquely determines $\mathbf{P}_\mathbb{U}$ up to a scalar multiple we can proceed as follows. Instead of $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ we choose a different basis $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ and construct all the minors for the Plücker coordinates induced by $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$. Then we can write every vector \mathbf{v}_i in these minors as a linear combination of the original basis $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$. But, using the multilinearity property of determinants, we can develop every such minor into a sum of other determinants. The reader can check that many determinants in this sum are zero, the result is the original determinant multiplied by a factor, and that this same factor affects all the other minors.

A proof that a $\binom{m+1}{k}$ -tuple of real numbers, made up of the $k \times k$ minors of a real matrix containing the basis of a subspace \mathbb{U} uniquely determines \mathbb{U} can be found in [41,

Chapter VII]. However, we check this for a simple example. If $\mathbf{P}_U = (\underline{\mathbf{P}}_{2,3}, \underline{\mathbf{P}}_{3,1}, \underline{\mathbf{P}}_{1,2})$ is made out of the 2×2 indicated minors of

$$\underline{\mathbf{P}} = \begin{pmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \end{pmatrix},$$

then \mathbf{P}_U uniquely determines the line along (a_1, a_2) and (b_1, b_2) , since the equation of this line can be written as

$$\begin{vmatrix} x & y & 1 \\ a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \end{vmatrix} = 0,$$

or, equivalently, $\underline{\mathbf{P}}_{2,3}x + \underline{\mathbf{P}}_{3,1}y + \underline{\mathbf{P}}_{1,2} = 0$. \square

A closely related result follows. If we have a k -dimensional vector space U the *signed k -volume* of k vectors $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k$ of U is defined as $\det(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k)$. Then, [23]:

Theorem 5.3. *Two Plücker vectors are equal if and only if they are constructed from bases of the same subspace and define equal signed k -volumes in this subspace.*

5.3 Grassmann-Cayley Algebra

The Grassmann-Cayley algebra is defined as a vector-space $\Lambda(\mathbb{V})$ with two operations on the elements of $\Lambda(\mathbb{V})$: *join* and *meet*. Informally speaking, $\Lambda(\mathbb{V})$ can be defined in the following way. If \mathbb{V} is the $(m+1)$ -dimensional vector space \mathbb{R}^{m+1} (which we regard as the vector space associated with \mathcal{P}^m), then we can construct a collection of spaces $\mathbb{V}^{(k)}$, $k = 0, \dots, m+1$, where $\mathbb{V}^{(k)}$ is the space generated by all possible Plücker coordinate vectors of k -dimensional subspaces of \mathbb{V} , and their linear combinations. Then $\Lambda(\mathbb{V})$ is the ‘union’ of all $\mathbb{V}^{(k)}$, $k = 0, \dots, m+1$. This union will be defined as the usual direct sum of linear vector spaces.

Also roughly speaking, the join operation takes two Plücker coordinate vectors $\underline{\mathbf{P}}_{U_1}$ and $\underline{\mathbf{P}}_{U_2}$, representing the subspaces U_1 and U_2 of \mathbb{V} , and derives a Plücker coordinate vector for the smallest subspace of \mathbb{V} containing *both* U_1 and U_2 . The meet operator takes the same input, but produces the largest subspace of \mathbb{V} contained *both* in U_1 and U_2 . Let us formalise this a bit more.

Let \mathbb{V} be an $(m+1)$ -dimensional vector space over the field \mathbb{R} . Let U be a k -dimensional subspace of \mathbb{V} , and \mathbf{A} be the Plücker coordinate vector of U , obtained from a basis $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$ of U . Then, we symbolically write the vector \mathbf{A} as

$$\mathbf{A} = \mathbf{u}_1 \vee \mathbf{u}_2 \vee \dots \vee \mathbf{u}_k,$$

and refer to $\mathbf{u}_1 \vee \mathbf{u}_2 \vee \dots \vee \mathbf{u}_k$ as a *k -extensor*. The number k is called the *step* of the extensor. We say that U is the *support* of \mathbf{A} , which we write as $U = \overline{\mathbf{A}}$. Moreover, $\mathbf{u}_1 \vee \mathbf{u}_2 \vee \dots \vee \mathbf{u}_k$ is an element of a $\binom{m+1}{k}$ -dimensional vector space, hereafter referred to

as $\mathbb{V}^{(k)}$. $\mathbb{V}^{(k)}$ is defined as the space containing all k -extensors of k -dimensional subspaces of \mathbb{V} and their linear combinations.

In general, a sum of k -extensors is not another k -extensor, since the resulting vector may not be expressible as the Plücker coordinate vector of some subspace. An $\binom{m+1}{k}$ -tuple of numbers is decomposable as a vector of $\binom{m+1}{k}$ $k \times k$ minors of a $k \times (m+1)$ matrix if, and only if, it satisfies the so-called *quadratic p -relations* [41, pag. 309]. For example, if $m = 3$ and $\mathbb{V} = \mathbb{R}^4$, the p -relation that a sextuple $\mathbf{P} = (\underline{\mathbf{P}}_{1,4}, \underline{\mathbf{P}}_{2,4}, \underline{\mathbf{P}}_{3,4}, \underline{\mathbf{P}}_{2,3}, \underline{\mathbf{P}}_{3,1}, \underline{\mathbf{P}}_{1,2})$ must accomplish is

$$\underline{\mathbf{P}}_{1,4}\underline{\mathbf{P}}_{2,3} + \underline{\mathbf{P}}_{2,4}\underline{\mathbf{P}}_{3,1} + \underline{\mathbf{P}}_{3,4}\underline{\mathbf{P}}_{1,2} = 0.$$

The elements of $\mathbb{V}^{(k)}$ that are not k -extensors are called *indecomposable k -tensors* and they can always be written as a sum of k -extensors. The general term *antisymmetric k -tensor* refers either to k -extensors or indecomposable k -tensors.

Before we formally define $\Lambda(\mathbb{V})$, we recall the definition of the direct sum operator ‘ \oplus ’ between vector spaces. If \mathbb{E} and \mathbb{F} are two vector spaces over a same field K , the *direct sum* of \mathbb{E} and \mathbb{F} is the set $\mathbb{E} \times \mathbb{F}$ with the operations

$$\begin{aligned} (\mathbf{u}, \mathbf{v}) + (\mathbf{u}_1, \mathbf{v}_1) &= (\mathbf{u} + \mathbf{u}_1, \mathbf{v} + \mathbf{v}_1) \\ k(\mathbf{u}, \mathbf{v}) &= (k\mathbf{u}, k\mathbf{v}), \end{aligned}$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{E}$, $\mathbf{u}_1, \mathbf{v}_1 \in \mathbb{F}$ and $k \in K$. With these operations $\mathbb{E} \times \mathbb{F}$ is a vector space, which we denote as $\mathbb{E} \oplus \mathbb{F}$. It is a basic result of linear algebra [15] that $\dim(\mathbb{E} \oplus \mathbb{F}) = \dim(\mathbb{E}) + \dim(\mathbb{F})$.

Definition 5.2. The vector space $\Lambda(\mathbb{V})$ is the direct sum of all $\mathbb{V}^{(k)}$:

$$\Lambda(\mathbb{V}) = \mathbb{V}^{(0)} \oplus \mathbb{V}^{(1)} \oplus \dots \oplus \mathbb{V}^{(n)}.$$

We note that $\dim(\Lambda(\mathbb{V})) = \sum_{k=0}^{m+1} \binom{m+1}{k} = 2^{m+1}$.

5.3.1 Join

Let $\mathbf{A} = a_1 \vee a_2 \vee \dots \vee a_k$ and $\mathbf{B} = b_1 \vee b_2 \vee \dots \vee b_j$ be two extensors. The *join product* (or *join*, for short) of \mathbf{A} and \mathbf{B} is defined as the $(j+k)$ -extensor

$$\mathbf{A} \vee \mathbf{B} = a_1 \vee a_2 \vee \dots \vee a_k \vee b_1 \vee b_2 \vee \dots \vee b_j. \quad (5.1)$$

Theorem 5.4. *If the vectors $a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_j$ are linearly dependent, then $\mathbf{A} \vee \mathbf{B} = 0$. If they are independent, then the support of $\mathbf{A} \vee \mathbf{B}$ is the union of the supports of \mathbf{A} and \mathbf{B} : $\overline{\mathbf{A} \vee \mathbf{B}} = \overline{\mathbf{A}} + \overline{\mathbf{B}} = \text{span}(\overline{\mathbf{A}} \cup \overline{\mathbf{B}})$.*

Proposition 5.1. *The join operator has the following properties.*

- **Associativity:** $\mathbf{A} \vee (\mathbf{B} \vee \mathbf{C}) = (\mathbf{A} \vee \mathbf{B}) \vee \mathbf{C}$, where \mathbf{A} and \mathbf{B} are extensors of any step $k \in \{0, \dots, m+1\}$.

- **Distributivity over addition:** $\mathbf{A} \vee (\mathbf{B} + \mathbf{C}) = \mathbf{A} \vee \mathbf{B} + \mathbf{A} \vee \mathbf{C}$, where \mathbf{A} and \mathbf{B} are extensors of equal step.
- **Anticommutativity:** $\mathbf{A} \vee \mathbf{B} = (-1)^{kj} \mathbf{B} \vee \mathbf{A}$, where \mathbf{A} and \mathbf{B} are extensors of step k and j , respectively.

If we are given the Plücker coordinate vectors, say \mathbf{A} and \mathbf{B} , of two k -extensors and want to compute their join product $\mathbf{A} \vee \mathbf{B}$, then we can decompose them as the join of several projective points and use Formula 5.1. However, such a decomposition is not necessary and there is a direct formula working with the Plücker coordinates of \mathbf{A} and \mathbf{B} . See [18]. We give the formula for the case $m = 3$ and the join product of a 2-extensor (a line) and a 1-extensor (a point), since this product is needed in next chapter. If the line is $\mathbf{L} = (\underline{\mathbf{Q}}_{1,4}, \underline{\mathbf{Q}}_{2,4}, \underline{\mathbf{Q}}_{3,4}, \underline{\mathbf{Q}}_{2,3}, \underline{\mathbf{Q}}_{3,1}, \underline{\mathbf{Q}}_{1,2})$ and the point is $\mathbf{p} = (p_1, p_2, p_3, p_4)$, then $\mathbf{L} \vee \mathbf{p} = (\underline{\mathbf{P}}_{2,3,4}, \underline{\mathbf{P}}_{3,4,1}, \underline{\mathbf{P}}_{1,2,4}, \underline{\mathbf{P}}_{2,3,1})$, where

$$\begin{aligned} \underline{\mathbf{P}}_{2,3,4} &= p_2 \underline{\mathbf{Q}}_{3,4} - p_3 \underline{\mathbf{Q}}_{2,4} + p_4 \underline{\mathbf{Q}}_{2,3}, & \underline{\mathbf{P}}_{3,4,1} &= p_3 \underline{\mathbf{Q}}_{4,1} - p_4 \underline{\mathbf{Q}}_{3,1} + p_1 \underline{\mathbf{Q}}_{3,4}, \\ \underline{\mathbf{P}}_{1,2,4} &= p_1 \underline{\mathbf{Q}}_{2,4} - p_2 \underline{\mathbf{Q}}_{1,4} + p_4 \underline{\mathbf{Q}}_{1,2}, & \underline{\mathbf{P}}_{2,3,1} &= p_2 \underline{\mathbf{Q}}_{3,1} - p_3 \underline{\mathbf{Q}}_{2,1} + p_1 \underline{\mathbf{Q}}_{2,3}. \end{aligned}$$

5.3.2 Meet

Although the meet operator is not strictly necessary for the purposes of this Thesis, we briefly introduce it here for the sake of completeness. Let $\mathbf{A} = a_1 \vee a_2 \vee \dots \vee a_k$ and $\mathbf{B} = b_1 \vee b_2 \vee \dots \vee b_j$ be two extensors, with $k + j > m + 1$. The *meet* of \mathbf{A} and \mathbf{B} is defined as:

$$\mathbf{A} \wedge \mathbf{B} = \sum_{\sigma} \text{sgn}(\sigma) [a_{\sigma(1)}, \dots, a_{\sigma(m-j+1)}, b_1, \dots, b_j] a_{\sigma(m-j+2)} \dots a_{\sigma(k)},$$

where the sum is over all the permutations σ of $\{1, 2, \dots, k\}$ such that

$$\sigma(1) < \sigma(2) < \dots < \sigma(m - j + 1),$$

and

$$\sigma(m - j + 2) < \sigma(m - j + 3) < \dots < \sigma(k).$$

Theorem 5.5. *If $\mathbf{A} \neq 0$ and $\mathbf{B} \neq 0$ and $\overline{\mathbf{A}} \cup \overline{\mathbf{B}}$ spans \mathbb{V} , then $\overline{\mathbf{A} \wedge \mathbf{B}} = \overline{\mathbf{A}} \cap \overline{\mathbf{B}}$, otherwise $\mathbf{A} \wedge \mathbf{B} = 0$.*

Proposition 5.2. *The meet operator has the following properties:*

- **Associativity:** $\mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{C}$, where \mathbf{A} and \mathbf{B} are extensors of any step $k \in \{0, \dots, m + 1\}$.

- **Distributivity over addition:** $\mathbf{A} \wedge (\mathbf{B} + \mathbf{C}) = \mathbf{A} \wedge \mathbf{B} + \mathbf{A} \wedge \mathbf{C}$, where \mathbf{A} and \mathbf{B} are extensors of equal step.
- **Anticommutativity:** $\mathbf{A} \wedge \mathbf{B} = (-1)^{(m+1-k)(m-1-j)} \mathbf{B} \wedge \mathbf{A}$, where \mathbf{A} and \mathbf{B} are extensors of step k and j , respectively.

5.4 Kinematics in the Plane

The goal of this section is to give a simplified projective notation for the motion of a point. We will see that it is possible to treat both translations and rotations using exactly the same algebraic representation. This is due to the fact that a translation can actually be interpreted as a rotation about a center at infinity. We assume that the reader is familiar with the basic concepts of plane and spatial kinematics. In particular, the notions of *instantaneous center of rotation*, for motions in the plane, and *instantaneous screw axis*, for motions in 3-space, will play an important role. For the sake of conciseness, the adjective *instantaneous* will often be omitted and we will simply talk about the *center of rotation* and the *screw axis*, respectively. Detailed definitions of these concepts, as well as applications, can be found in [96, 95] and the books by Agulló [1], Hunt [45] or Bottema and Roth [12], to name a few. We will first introduce the kinematics of motions on a plane to ease the development of the same concepts for general motions in 3-space.

5.4.1 Instantaneous Motion of a Point

We start with a projective definition for the *instantaneous motion* of a point (that is, its velocity) and then see that the concept is transparent to the type of motion: it can either be used to represent rotational or translational velocities. Since we only deal with velocities the adjective ‘instantaneous’ will be omitted for conciseness.

Definition 5.3 (Motion of a point). For a point P on a body \mathcal{B} , undergoing a velocity $\mathbf{v}(P) = (v_1, v_2)$, we define the *motion of \mathcal{B} at P* , $\mathbf{M}_{\mathcal{B}}(P)$, as the vector

$$\mathbf{M}_{\mathcal{B}}(P) = \begin{pmatrix} v_1 \\ v_2 \\ -\mathbf{v}(P) \cdot P \end{pmatrix}. \quad (5.2)$$

Unless otherwise stated, the velocity $\mathbf{v}(P)$ and the motion $\mathbf{M}_{\mathcal{B}}(P)$ are assumed to be relative to an absolute frame. If we work with several frames, we usually designate them with numbers, reserving frame 0 for the absolute frame. In such cases, we will write $\mathbf{M}_{\mathcal{B},i}(P)$ to indicate the motion of P relative to the i th-frame, assuming that P is rigidly attached to body \mathcal{B} .

5.4.2 Pure Rotations

Suppose that \mathcal{B} is undergoing a rotation about its instantaneous center of rotation, say the point C , at an angular velocity ω (Figure 5.3). Then, the induced velocity at $P = (p_1, p_2)$, $\mathbf{v}(P)$, is a vector of length $\omega\|P - C\|$ turned 90 degrees counterclockwise from $P - C$. Thus, $\mathbf{v}(P) = (v_1, v_2) = \omega(c_2 - p_2, p_1 - c_1)$, $-\mathbf{v}(P) \cdot P = \omega(c_1 p_2 - c_2 p_1)$ and the motion at P can be written as

$$\mathbf{M}_{\mathcal{B}}(P) = \begin{pmatrix} v_1 \\ v_2 \\ -\mathbf{v}(P) \cdot P \end{pmatrix} = \omega \begin{pmatrix} c_2 - p_2 \\ p_1 - c_1 \\ c_1 p_2 - c_2 p_1 \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{P}}_{23} \\ \underline{\mathbf{P}}_{31} \\ \underline{\mathbf{P}}_{12} \end{pmatrix},$$

where

$$\underline{\mathbf{P}} = \begin{pmatrix} \omega c_1 & \omega c_2 & \omega \\ p_1 & p_2 & 1 \end{pmatrix}.$$

As indicated, the coordinates of $\mathbf{M}_{\mathcal{B}}(P)$ can be seen as the three 2×2 minors of a 2×3 matrix whose two rows are the homogeneous coordinates of the center of rotation, weighted by ω , and the point P , respectively. These three minors are Plücker coordinates for the line CP , and by Theorem 5.3 they define a segment on this line, with the same length and orientation as $P \rightarrow \omega C$ has (see Figure 5.3a). Thus, in Grassmann-Cayley algebra we can represent the motion of P with the 2-extensor

$$\mathbf{M}_{\mathcal{B}}(P) = \omega \mathbf{c} \vee \mathbf{p},$$

and the instantaneous center of rotation is represented by the weighted projective point $\omega \mathbf{c}$. The 1-extensor $\omega \mathbf{c}$ is simply called the *center* of the motion. Usually, we will assume that the angular velocity is absorbed inside a weighted point $\mathbf{c}_{\mathcal{B}}$, letting $\mathbf{c}_{\mathcal{B}} = \omega \mathbf{c}$, and we will simply write

$$\mathbf{M}_{\mathcal{B}}(P) = \mathbf{c}_{\mathcal{B}} \vee \mathbf{p},$$

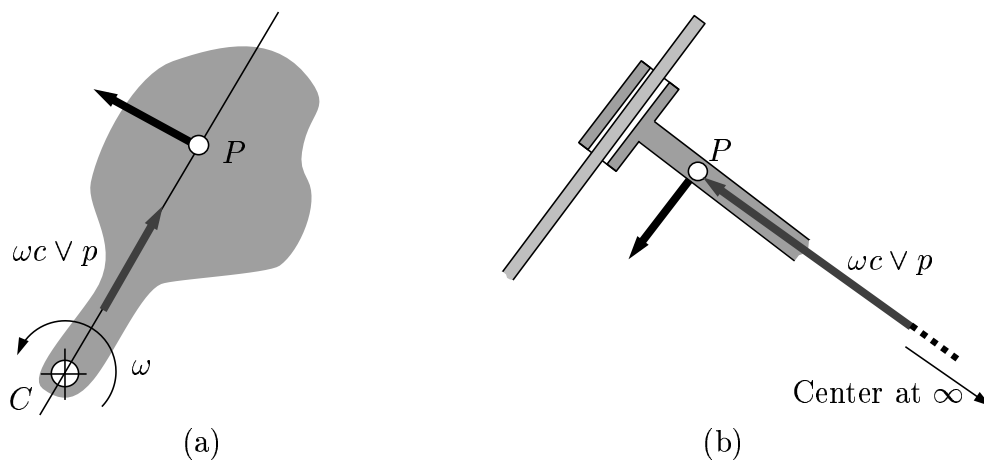


Figure 5.3: Rotations (a) and translations (b) seen as 2-extensors.

assuming that the third coordinate of \mathbf{c}_B represents ω .

In Kinematics, the concept of instantaneous center of rotation is always relative to a reference frame and \mathbf{c}_B is meant with reference to the absolute frame. When we need to specify the frame for a center we will write $\mathbf{c}_{B,i}$ to indicate the center of rotation of body B with respect to the i th frame. We can speak of the center of rotation of a given reference frame j too, assuming that the points in this frame move rigidly together. The center for such a frame, relative to the i th frame, will be written as $\mathbf{c}_{j,i}$.

5.4.3 Pure Translations

When B is undergoing a pure translation, with all its points at a velocity $v = (t_1, t_2)$, it is also possible to see the motion of one of its points as a 2-extensor, since we can write

$$\mathbf{M}_B(P) = \begin{pmatrix} v_1 \\ v_2 \\ -v \cdot P \end{pmatrix} = \omega \begin{pmatrix} t_1 \\ t_2 \\ -t_1 p_1 - t_2 p_2 \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{P}}_{23} \\ \underline{\mathbf{P}}_{31} \\ \underline{\mathbf{P}}_{12} \end{pmatrix},$$

where

$$\underline{\mathbf{P}} = \begin{pmatrix} -t_2 & t_1 & 0 \\ p_1 & p_2 & 1 \end{pmatrix},$$

and hence

$$\mathbf{M}_B(P) = (-t_2, t_1, 0) \vee (p_1, p_2, 1).$$

By analogy to the rotational case, we can view the translation of P as a rotation with an infinitesimal angular velocity about the center $\mathbf{c}_B = (-t_2, t_1, 0)$, a point at infinity in the direction $(-t_2, t_1)$ of the plane. Here, the components of $(-t_2, t_1, 0) \vee (p_1, p_2, 1)$ can also be regarded as representing an oriented line segment from the point P to the center at infinity (Figure 5.3b).

5.4.4 Composition of Motions

Assume we have two reference frames from where we can measure the velocity of P : the “absolute” frame, and the “relative” frame. From the law of composition of velocities of Kinematics we know that the absolute velocity of P equals the sum of its relative velocity plus the absolute velocity that P would have if it was rigidly attached to the relative frame. If we work with projective motions instead of Euclidean velocities, we have an analogous relationship:

Proposition 5.3. *Let $\mathbf{M}_{B,0}(P)$ and $\mathbf{M}_{B,1}(P)$ be the motions of a point P on a body B with respect to the absolute frame 0 and the relative frame 1, respectively. Let $\mathbf{M}_{1,0}(P)$ be the motion that P would have if P was rigidly attached to frame 1, as measured from frame 0. Then,*

$$\mathbf{M}_{B,0}(P) = \mathbf{M}_{B,1}(P) + \mathbf{M}_{1,0}(P). \quad (5.3)$$

Proof. It is straightforward to verify this by substituting every motion in Equation 5.3 by its definition in terms of the point P and its velocity, as in Equation 5.2. The law of composition of velocities confirms the equality, row by row, in the resulting vector equation. \square

When we work with points in the 2-dimensional projective space \mathcal{P}^2 , the sum of 2-extensors is another 2-extensor. That is, it is a 3-tuple that can be decomposed as the join product of two points of \mathcal{P}^2 . This can be seen if we let $\mathbf{M}_{\mathcal{B},1}(P) = \mathbf{c}_{\mathcal{B},1} \vee \mathbf{p}$ and $\mathbf{M}_{1,0}(P) = \mathbf{c}_{1,0} \vee \mathbf{p}$ and write Equation 5.3 in terms of these 2-extensors:

$$\mathbf{M}_{\mathcal{B},0}(P) = \mathbf{c}_{\mathcal{B},1} \vee \mathbf{p} + \mathbf{c}_{1,0} \vee \mathbf{p},$$

or, equivalently,

$$\mathbf{M}_{\mathcal{B},0}(P) = (\mathbf{c}_{\mathcal{B},1} + \mathbf{c}_{1,0}) \vee \mathbf{p},$$

and since $\mathbf{c}_{\mathcal{B},1} + \mathbf{c}_{1,0}$ is a new projective point of \mathcal{P}^2 , $\mathbf{M}_{\mathcal{B},0}(P)$ is a 2-extensor. This also proves the following well-known theorem of Kinematics.

Theorem 5.6. *Every composition of translational and/or rotational velocities in the plane is equivalent either to a pure rotation or a pure translation. The former corresponds to a projective rotation about a proper point, and the latter to a projective rotation about a point at infinity.*

In general, the sum of k -extensors made up with points in \mathcal{P}^n , $n > 2$, is not again a k -extensor and, as seen below, for motions in 3-space it will not be possible to rewrite a composition of motions as a pure rotation about an axis or a pure translation along it.

5.4.5 Centers of Two Hinged Bodies

If we have two rigid bodies \mathcal{B}_1 and \mathcal{B}_2 in the plane, hinged at a point P with a rotational joint, then the centers of their motions are related as follows.

Proposition 5.4. *Let $\mathbf{c}_{1,0}$ and $\mathbf{c}_{2,0}$ be the centers of \mathcal{B}_1 and \mathcal{B}_2 , respectively, relative to the absolute frame 0. The center of \mathcal{B}_2 with respect to a relative frame 1 on \mathcal{B}_1 is $\mathbf{c}_{\mathcal{B}_2,1} = \omega_{\mathcal{B}_2,1} \mathbf{p}$, where $\omega_{\mathcal{B}_2,1}$ is the angular velocity of \mathcal{B}_2 measured from frame 1. Then,*

$$\mathbf{c}_{\mathcal{B}_2,0} = \mathbf{c}_{\mathcal{B}_2,1} + \mathbf{c}_{1,0},$$

Proof. For a point Q on \mathcal{B}_2 :

$$\mathbf{M}_{\mathcal{B}_2,0}(Q) = \mathbf{M}_{\mathcal{B}_2,1}(Q) + \mathbf{M}_{1,0}(Q),$$

or, equivalently

$$\mathbf{c}_{\mathcal{B}_2,0} \vee \mathbf{q} = \mathbf{c}_{\mathcal{B}_2,1} \vee \mathbf{q} + \mathbf{c}_{1,0} \vee \mathbf{q}.$$

And eliminating \mathbf{q} , we see that

$$\mathbf{c}_{\mathcal{B}_2,0} = \mathbf{c}_{\mathcal{B}_2,1} + \mathbf{c}_{1,0}.$$

\square

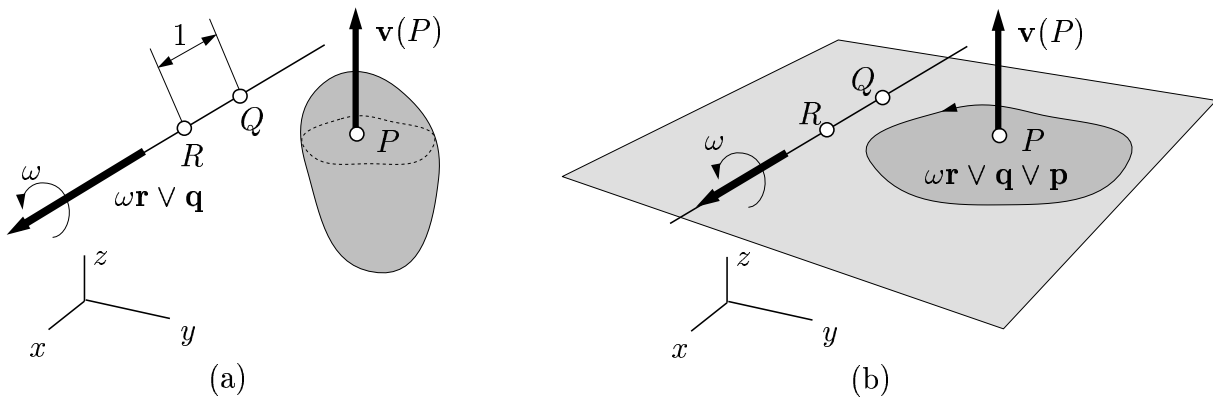


Figure 5.4: (a) The instantaneous center of rotation is represented by a 2-extensor $\boldsymbol{\omega} \mathbf{r} \vee \mathbf{q}$ (a), and the motion at a point by a 3-extensor $\boldsymbol{\omega} \mathbf{r} \vee \mathbf{q} \vee \mathbf{p}$ (b).

This leads to a well-known result of planar Kinematics, the theorem of the three centers by Kennedy and Aronhold.

Theorem 5.7 (Kennedy and Aronhold). *For two moving bodies \mathcal{B}_1 and \mathcal{B}_2 , linked by a rotational joint P , their two instantaneous centers of rotation must be aligned with P .*

Proof. The equation $\mathbf{c}_{\mathcal{B}_2,0} = \mathbf{c}_{\mathcal{B}_2,1} + \mathbf{c}_{1,0}$ says that the homogeneous points $\mathbf{c}_{\mathcal{B}_2,0}$, $\mathbf{c}_{\mathcal{B}_2,1}$ and $\mathbf{c}_{1,0}$ are linearly dependent, or equivalently, that their Euclidean versions are aligned. \square

5.5 Kinematics in 3-Space

5.5.1 Instantaneous Motion of a Point

As for the 2-dimensional case, the following definition for the instantaneous motion of a point will allow to express its velocity as an appropriate extensor in the Grassman-Cayley algebra, independently of whether it is a rotation or a translation.

Definition 5.4 (Motion of a point). Given a rigid body \mathcal{B} and a point P on it, undergoing a velocity $\mathbf{v}(P) = (v_1, v_2, v_3)$, we define the motion of \mathcal{B} at P , $\mathbf{M}_{\mathcal{B}}(P)$, as the vector

$$\mathbf{M}_{\mathcal{B}}(P) = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ -\mathbf{v}(P) \cdot P \end{pmatrix}. \quad (5.4)$$

5.5.2 Pure Rotations

Suppose that \mathcal{B} is instantaneously rotating about an axis defined by two points, say Q and R , with angular velocity ω , as indicated in Figure 5.4a. The line QR is the instantaneous screw axis of \mathcal{B} under this motion. From Kinematics we know that the induced velocity at a point P of \mathcal{B} can be computed as the cross-product $\mathbf{v}(P) = \omega(R - Q) \times (P - Q)$, where P and Q have been properly selected so that the vector $R - Q$ has a unit length. Thus:

$$\mathbf{v}(P) = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \omega \begin{vmatrix} r_1 - q_1 & p_1 - q_1 & i \\ r_2 - q_2 & p_2 - q_2 & j \\ r_3 - q_3 & p_3 - q_3 & k \end{vmatrix} = \omega \begin{vmatrix} q_1 & r_1 & p_1 & i \\ q_2 & r_2 & p_2 & j \\ q_3 & r_3 & p_3 & k \\ 1 & 1 & 1 & 0 \end{vmatrix},$$

and the three components of the velocity are the minors associated with i , j and k in the last determinant. Also, note that $-\mathbf{v}(P) \cdot P$ is actually the determinant

$$-\mathbf{v}(P) \cdot P = -\omega \begin{vmatrix} q_1 & r_1 & p_1 & p_1 \\ q_2 & r_2 & p_2 & p_2 \\ q_3 & r_3 & p_3 & p_3 \\ 1 & 1 & 1 & 0 \end{vmatrix} = -\omega \begin{vmatrix} q_1 & r_1 & p_1 \\ q_2 & r_2 & p_2 \\ q_3 & r_3 & p_3 \end{vmatrix},$$

and, hence, the motion of P can be written in terms of Plücker coordinates as,

$$M_{\mathcal{B}}(P) = (\underline{\mathbf{P}}_{2,3,4}, \underline{\mathbf{P}}_{3,4,1}, \underline{\mathbf{P}}_{1,2,4}, \underline{\mathbf{P}}_{2,3,1})^T,$$

where

$$\underline{\mathbf{P}} = \begin{pmatrix} \omega q_1 & \omega q_2 & \omega q_3 & \omega \\ r_1 & r_2 & r_3 & 1 \\ p_1 & p_2 & p_3 & 1 \end{pmatrix}.$$

Thus, in Grassmann-Cayley algebra we can represent $M_{\mathcal{B}}(P)$, as the 3-extensor

$$\mathbf{M}_{\mathcal{B}}(P) = \omega \mathbf{q} \vee \mathbf{r} \vee \mathbf{p}.$$

The “first half” of this motion ($\omega \mathbf{q} \vee \mathbf{r}$) is a 2-extensor representing the instantaneous screw axis of the rotation, as it is a 6-tuple of Plücker coordinates representing the line QR . However, like in the planar case, it is more than this. By Theorem 5.3 $\omega \mathbf{q} \vee \mathbf{r}$ uniquely determines an oriented line segment on this axis, with the same length and orientation as the segment $R \rightarrow \omega Q$ and, thus, it also determines the direction of the rotation and its angular velocity. Following Forder [28, pag. 14] we call $\omega \mathbf{q} \vee \mathbf{r}$ a *rotor*, although other classical texts call it a *dyname* [51, pag. 32]. This rotor will be compactly written as $\mathbf{S}_{\mathcal{B}}$ and, hereafter, we will often write the motion of P compactly as

$$\mathbf{M}_{\mathcal{B}}(P) = \mathbf{S}_{\mathcal{B}} \vee \mathbf{p}.$$

If no other subindex is specified, $\mathbf{S}_{\mathcal{B}}$ is taken to represent the instantaneous screw axis of \mathcal{B} relative to the absolute frame. When we need to specify the frame we will write $\mathbf{S}_{\mathcal{B},i}$ to indicate the screw axis of \mathcal{B} with respect to the i th frame and let $\mathbf{S}_{i,j}$ denote the screw axis of the i th frame relative to the j th frame.

Finally, we note that the 4-tuple $M_{\mathcal{B}}(P) = (\underline{\mathbf{P}}_{2,3,4}, \underline{\mathbf{P}}_{3,4,1}, \underline{\mathbf{P}}_{1,2,4}, \underline{\mathbf{P}}_{2,3,1})^T$ can also be represented geometrically as an oriented section of the plane through P , perpendicular to $\mathbf{v}(P)$, with an area equal to the length of $\mathbf{v}(P)$ (Figure 5.4b).

5.5.3 Pure Translations

If the velocity of \mathcal{B} is translational, say $\mathbf{v}(P) = (t_1, t_2, t_3)$ for all points P of \mathcal{B} , we can also express the motion as a 3-extensor. Indeed, note that we can always find two vectors $\mathbf{a}' = (a_1, b_2, b_3)$ and $\mathbf{b}' = (b_1, b_2, b_3)$ of \mathbb{R}^3 , such that

$$\mathbf{a}' \times \mathbf{b}' = (t_1, t_2, t_3).$$

Then, if $\mathbf{a} = (a_1, a_2, a_3, 0)$, $\mathbf{b} = (b_1, b_2, b_3, 0)$ and $\mathbf{p} = (p_1, p_2, p_3, 1)$, the motion of P can be written as

$$\mathbf{M}_{\mathcal{B}}(P) = \mathbf{a} \vee \mathbf{b} \vee \mathbf{p},$$

since if

$$\underline{\mathbf{P}} = \begin{pmatrix} a_1 & a_2 & a_3 & 0 \\ b_1 & b_2 & b_3 & 0 \\ p_1 & p_2 & p_3 & 1 \end{pmatrix},$$

then,

$$\mathbf{a} \vee \mathbf{b} \vee \mathbf{p} = \begin{pmatrix} \underline{\mathbf{P}}_{234} \\ \underline{\mathbf{P}}_{341} \\ \underline{\mathbf{P}}_{124} \\ \underline{\mathbf{P}}_{231} \end{pmatrix} = \begin{pmatrix} a_2 b_3 - b_2 a_3 \\ a_3 b_1 - b_3 a_1 \\ a_1 b_2 - b_1 a_2 \\ \begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ p_1 & p_2 & p_3 \end{vmatrix} \end{pmatrix} = \begin{pmatrix} \mathbf{a}' \times \mathbf{b}' \\ -(\mathbf{a}' \times \mathbf{b}') \cdot P \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ - \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \end{pmatrix},$$

and we see that the velocity of P can be seen as a rotation about an axis at infinity, defined by the two projective points \mathbf{a} and \mathbf{b} . By analogy to the translational case $\mathbf{S}_{\mathcal{B}} = \mathbf{a} \vee \mathbf{b}$ is called the *rotor* of this motion and we write

$$\mathbf{M}_{\mathcal{B}}(P) = \mathbf{S}_{\mathcal{B}} \vee \mathbf{p}.$$

5.5.4 Composition of Motions

As for motions in the plane, Proposition 5.3 is also valid in 3-space, and the composition of motions corresponds to the addition of the associated 3-extensors. Thus, using the same notation, we can write

$$\mathbf{M}_{\mathcal{B},0}(P) = \mathbf{M}_{\mathcal{B},1}(P) + \mathbf{M}_{1,0}(P).$$

If $\mathbf{M}_{\mathcal{B},1}(P)$ and $\mathbf{M}_{1,0}(P)$ are pure rotations or translations, then there are rotors $\mathbf{S}_{\mathcal{B},1}$ and $\mathbf{S}_{1,0}$ for which $\mathbf{M}_{\mathcal{B},1}(P) = \mathbf{S}_{\mathcal{B},1} \vee \mathbf{p}$ and $\mathbf{M}_{1,0}(P) = \mathbf{S}_{1,0} \vee \mathbf{p}$, which yields

$$\mathbf{M}_{\mathcal{B},0}(P) = \mathbf{S}_{\mathcal{B},1} \vee \mathbf{p} + \mathbf{S}_{1,0} \vee \mathbf{p} = (\mathbf{S}_{\mathcal{B},1} + \mathbf{S}_{1,0}) \vee \mathbf{p}.$$

In the planar case a composition of rotations or translations did always reduce to a single rotation (possibly about a center at infinity). However, this is not possible in 3-space

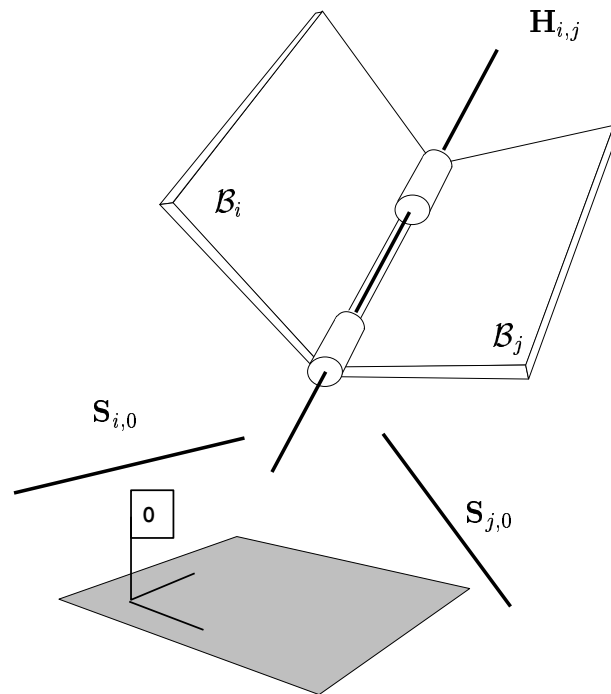


Figure 5.5: Two hinged bodies.

because the sum of rotors $\mathbf{S}_{\mathcal{B}_1} + \mathbf{S}_{1,0}$ is not in general a new rotor. (Recall that, in general, a sum of k -extensors is not decomposable as the join of k points.) Such a composition of rotors is called a *screw* and it can be proved that any instantaneous rigid motion of a body can be represented by such a 6-tuple [119, Section 4.2]. Moreover, every screw motion can be decomposed as a rotation about an axis and a translation parallel to it, i.e. as the sum of two rotors. This is known as Poincot's Central Axis Theorem in Kinematics and an explicit process for such a decomposition is given by Whiteley in [119, Section 3.3].

5.5.5 Screws of Two Hinged Bodies

Let \mathcal{B}_i and \mathcal{B}_j be two moving bodies articulated by a hinge on a line l (Figure 5.5). Let $\mathbf{H}_{i,j}$ be the Plücker coordinate vector of l computed as $\mathbf{H}_{i,j} = \mathbf{a} \vee \mathbf{b}$, where \mathbf{a} and \mathbf{b} are two points on l , separated by a unit distance. Also, let $\mathbf{S}_{\mathcal{B}_i,0}$ and $\mathbf{S}_{\mathcal{B}_j,0}$ be, respectively, the screws of \mathcal{B}_i and \mathcal{B}_j relative to the absolute frame. $\mathbf{S}_{\mathcal{B}_i,0}$ and $\mathbf{S}_{\mathcal{B}_j,0}$ cannot be arbitrary and the following proposition (the spatial analogue of Proposition 5.4) shows that they are restricted by $\mathbf{H}_{i,j}$ and the law of composition of velocities.

Proposition 5.5. *There is a scalar $\omega_{i,j}$ for which*

$$\mathbf{S}_{\mathcal{B}_i,0} = \omega_{i,j} \mathbf{H}_{i,j} + \mathbf{S}_{\mathcal{B}_j,0}. \quad (5.5)$$

Proof. If we take a point P on \mathcal{B}_i and fix a reference frame j on \mathcal{B}_j , we can write

$$\mathbf{M}_{\mathcal{B}_i,0}(P) = \mathbf{M}_{\mathcal{B}_i,j}(P) + \mathbf{M}_{j,0}(P),$$

or

$$\mathbf{S}_{\mathcal{B}_i,0} \vee \mathbf{p} = \mathbf{S}_{\mathcal{B}_i,j} \vee \mathbf{p} + \mathbf{S}_{\mathcal{B}_j,0} \vee \mathbf{p},$$

where $\mathbf{S}_{\mathcal{B}_i,j}$ is the screw of \mathcal{B}_i relative to frame j . But $\mathbf{S}_{\mathcal{B}_i,j} = \omega_{i,j} \mathbf{H}_{i,j}$, for some scalar $\omega_{i,j}$, and thus $\mathbf{S}_{\mathcal{B}_i,0} = \omega_{i,j} \mathbf{H}_{i,j} + \mathbf{S}_{\mathcal{B}_j,0}$. \square

Also, there is the following spatial analogue of Kennedy's theorem of the three centers (Theorem 5.7 above):

Theorem 5.8. *For two bodies articulated along a hinge, each of them is instantaneously following a pure rotational motion with respect to a reference frame if and only if the instantaneous axes of rotation of the two bodies are coplanar and copunctual with the axis of the hinge.*

Proof. Equation 5.5 expresses a linear dependence between the two screws of the two bodies and the extensor associated with the line of the hinge. When all motions are pure rotations, these screws are rotors (2-extensors). It can be seen that this happens if and only if the three lines of these rotors are coplanar and copunctual [20]. \square

Chapter 6

Realizability and Reconstruction through Instantaneous Kinematics

“Illusion consists in the conviction that there is only one way to interpret the visual pattern in front of us. We are blind to other possible configurations because we literally cannot imagine them”.

E. H. Gombridge —Art and Illusion, 1960.

There is a bijective mapping between the liftings of a line drawing \mathcal{D} and the instantaneous motions of an associated articulated mechanism: a panel-and-hinge framework with a panel for every face of \mathcal{D} and a hinge for every edge. This was discovered by Whiteley, who used it to prove the converse of Maxwell’s theorem in [121]. In this chapter we will see that this mapping permits a concise and efficient solution for the realizability problem; namely, a drawing will have a sharp lifting if and only if the kernel of a related matrix contains a vector with nonzero components. Despite its importance, we believe that the correspondence between liftings and motions has never been exploited by the Machine Vision community investigating the problem, probably because it came to light in the context of Structural Geometry [119, 120] and Rigidity Theory [31], a usually unnoticed source of results for our problems.

We develop further on this mapping and see that it allows us to define a linear parameterization of all liftings of a correct drawing, thus solving the reconstruction problem in a simple way. As a by-product we also get a linear parameterization of the compatible cross-sections, a tool we demanded in Chapter 4. Also, the sign patterns of the vectors in the mentioned kernel directly tell which edges are convex and which concave in the spatial interpretation. This will provide a nice mathematical model of the Necker reversal phenomenon and will allow a way to compute all $\{+, -\}$ -labellings of the drawing, without resorting to the use of a junction dictionary. Overall, the solutions we describe have several advantages over the best ones provided so far [104].

6.1 Panel-and-Hinge Frameworks

A *panel-and-hinge framework* is a pair $\mathcal{F}^{ph} = (P, H)$, where P is a collection of planar polygonal *panels* $\{P_1, P_2, \dots, P_n\}$, all rigid, and H is a collection of *hinges* $\{\dots, \mathbf{H}_{i,j}, \dots\}$, each articulating a pair of panels. $\mathbf{H}_{i,j}$ denotes the hinge articulating P_i and P_j . We represent a hinge by a 2-extensor of its supporting line, that is, $\mathbf{H}_{i,j} = \mathbf{a} \vee \mathbf{b}$, where a and b are two points on the rotation axis of the hinge. It will be advantageous to redundantly represent each hinge and we will include both $\mathbf{H}_{i,j} = \mathbf{a} \vee \mathbf{b}$ and $\mathbf{H}_{j,i} = \mathbf{b} \vee \mathbf{a}$ in H . Note that $\mathbf{H}_{i,j} = -\mathbf{H}_{j,i}$.

A *path* between two panels of \mathcal{F}^{ph} , say P_{i_1} and P_{i_n} , is an alternate sequence of panels and hinges $\{P_{i_1}, \mathbf{H}_{i_1,i_2}, P_{i_2}, \mathbf{H}_{i_2,i_3}, \dots, P_{i_{n-1}}, \mathbf{H}_{i_{n-1},i_n}, P_{i_n}\}$, such that no panel is repeated, and every panel P_{i_k} is adjacent to the next one $P_{i_{k+1}}$ through the hinge $\mathbf{H}_{i_k,i_{k+1}}$. A *cycle* of \mathcal{F}^{ph} is a closed path that begins and ends in the same panel, i.e., $P_{i_1} = P_{i_n}$. Figure 6.1 shows a panel-and-hinge framework with two cycles.

As in any mechanism, a usual question is to ask what are the possible instantaneous motions undergone by the panels of \mathcal{F}^{ph} , more precisely, which are the possible instantaneous screw axes $\mathbf{S}_{i,0}$ that every panel P_i can have, with respect to an absolute reference frame, here labelled as frame 0. This motivates the following definitions.

An *instantaneous motion* for \mathcal{F}^{ph} is an assignment of a screw axis $\mathbf{S}_{i,0}$ to each panel $P_i \in P$ such that for each $\mathbf{H}_{i,j} \in H$, $\mathbf{S}_{i,0} - \mathbf{S}_{j,0} = \omega_{i,j} \mathbf{H}_{i,j}$ for some choice of scalars $\omega_{i,j}$. Such an assignment induces velocities on the points of the panels that are actually satisfying the kinematic constraints imposed by the hinges. In other words, the fact that $\mathbf{S}_{i,0} - \mathbf{S}_{j,0} = \omega_{i,j} \mathbf{H}_{i,j}$ guarantees that the induced motions on points \mathbf{p} of the hinge $\mathbf{H}_{i,j}$ will be the same either if we compute them as $\mathbf{S}_{i,0} \vee \mathbf{p}$, or as $\mathbf{S}_{j,0} \vee \mathbf{p}$ (see Section 5.5.5). We note that, since

$$\omega_{j,i} \mathbf{H}_{j,i} = \mathbf{S}_{j,0} - \mathbf{S}_{i,0} = -\omega_{i,j} \mathbf{H}_{i,j} = \omega_{i,j} \mathbf{H}_{j,i},$$

then $\omega_{i,j} = \omega_{j,i}$ and there is a single scalar assigned to each hinge.

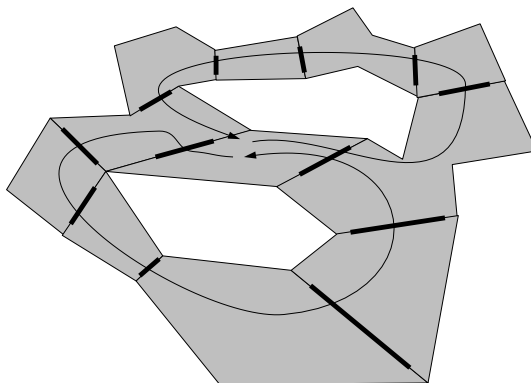


Figure 6.1: A panel-and-hinge framework with two cycles. Hinges are indicated as thick black segments.

A *motion assignment* for \mathcal{F}^{ph} is an assignment of a scalar $\omega_{i,j}$ to each hinge $\mathbf{H}_{i,j} \in H$, such that $\omega_{i,j} = \omega_{j,i}$ and $\sum_{\mathbf{H}_{i,j} \in C} \omega_{i,j} \cdot \mathbf{H}_{i,j} = 0$ for every cycle C of panels and hinges in \mathcal{F}^{ph} . If every hinge is a 2-extensor of two points on the axis, separated by a unitary distance, then $\omega_{i,j}$ can be interpreted as the angular velocity between P_i and P_j (See Section 5.5.2), and the following theorem guarantees that a motion assignment is equivalent to an instantaneous motion for \mathcal{F}^{ph} .

Theorem 6.1. *For a panel-and-hinge framework $\mathcal{F}^{ph} = (P, H)$ with a selected panel P_0 designated as the absolute reference frame, there exists a one-to-one correspondence between instantaneous motions relative to this panel and motion assignments.*

Proof. If we are given an instantaneous motion, then the scalars $\omega_{i,j}$ satisfying the equations $\mathbf{S}_{i,0} - \mathbf{S}_{j,0} = \omega_{i,j} \mathbf{H}_{i,j}$ already define a motion assignment. Certainly, if we consider any cycle C of panels and hinges, write down the equations $\mathbf{S}_{i,0} - \mathbf{S}_{j,0} = \omega_{i,j} \mathbf{H}_{i,j}$ for all hinges in C , and sum them all, we conclude that $\sum \omega_{i,j} \cdot \mathbf{H}_{i,j} = 0$. This gives one half of the correspondence.

If we are given a motion assignment, we can define a corresponding instantaneous motion as follows. We let $\mathbf{S}_{0,0} = 0$, as the panel P_0 is not moving with respect to itself. Then, to compute $\mathbf{S}_{i,0}$ for any other panel P_i we select an arbitrary path T of panels and hinges from P_0 to P_i and let:

$$\mathbf{S}_{i,0} = - \sum_{\mathbf{H}_{i,j} \in T} \omega_{i,j} \mathbf{H}_{i,j}, \quad (6.1)$$

where the sum is along all hinges of T . We note that the value of $\mathbf{S}_{i,0}$ is independent from the chosen path T , because any two different paths, say T_1 and T_2 , from P_0 to P_i will form a closed cycle C on which the sum is zero. That is, taking into account that a hinge $\mathbf{H}_{i,j}$ in T_2 corresponds to the hinge $\mathbf{H}_{j,i}$ of C we can write

$$0 = \sum_{\mathbf{H}_{i,j} \in C} \omega_{i,j} \mathbf{H}_{i,j} = \sum_{\mathbf{H}_{i,j} \in T_1} \omega_{i,j} \mathbf{H}_{i,j} - \sum_{\mathbf{H}_{i,j} \in T_2} \omega_{i,j} \mathbf{H}_{i,j},$$

and hence,

$$\sum_{\mathbf{H}_{i,j} \in T_1} \omega_{i,j} \mathbf{H}_{i,j} = \sum_{\mathbf{H}_{i,j} \in T_2} \omega_{i,j} \mathbf{H}_{i,j},$$

which completes the other half of the correspondence. \square

6.2 A Kinematic Test of Realizability

6.2.1 A Correspondence between Liftings and Motions

If \mathcal{D} is a line drawing with incidence structure $S = (V, F, I)$, we can associate \mathcal{D} with a panel-and-hinge framework $\mathcal{F}^{ph}(\mathcal{D})$ by putting a panel for each face $f \in F$, and a hinge

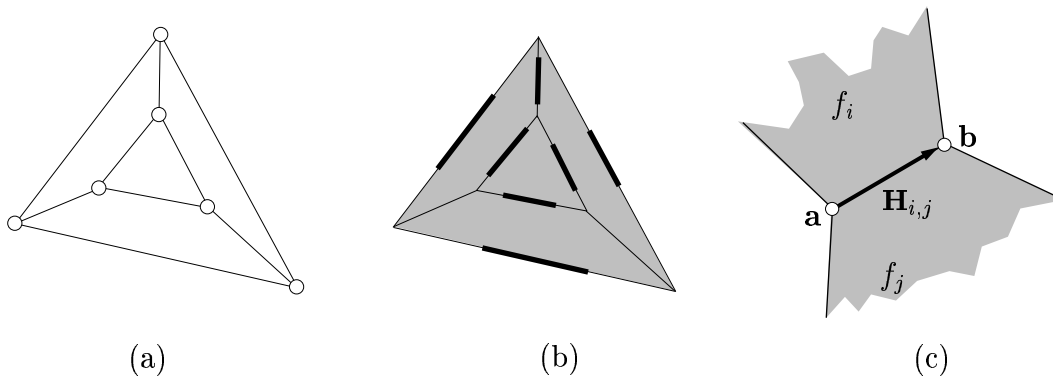


Figure 6.2: A line drawing (a) and its associated panel-and-hinge framework (b). (c) Orientation of the hinges.

articulating two panels if their associated faces are adjacent in \mathcal{D} . Figure 6.2 shows a projected truncated tetrahedron (a) and its associated framework (b). It will be helpful to give a common orientation to the hinges as follows. Assuming that the incidence structure has the topology of an orientable surface, we can orient the faces in F , designating an *outer* and *inner* side for each of them. Let the outer side be the one facing an observer placed at the center of projection. Then, for the hinge $\mathbf{H}_{i,j} = a \vee b$, we select a and b so that the vector $b - a$ turned 90 degrees clockwise points from face f_i to face f_j . See Figure 6.2c.

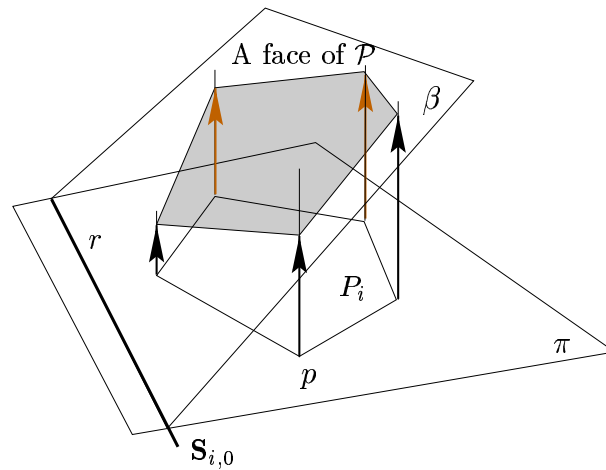
The following theorem reveals one sense of the announced relationship between the liftings of \mathcal{D} and the motion assignments of $\mathcal{F}^{ph}(\mathcal{D})$.

Theorem 6.2 (Whiteley, 1982). *If \mathcal{D} is the correct projection of a polysurface \mathcal{P} without vertical faces (but possibly with holes and handles), then $\mathcal{F}^{ph}(\mathcal{D})$ has a motion assignment that assigns a non-null angular velocity $\omega_{i,j}$ to every pair of adjacent panels whose corresponding faces of \mathcal{P} lie on different planes.*

Proof. Let π denote the plane where \mathcal{D} lies. An instantaneous motion can be found as follows. First, we take every panel P_i of $\mathcal{F}^{ph}(\mathcal{D})$ and assign a velocity vector to every point p of P_i : the vector from p to the lifted position of p on the polysurface \mathcal{P} in 3-space (see Figure 6.3). Since all these velocities are orthogonal to π , the instantaneous screw axis of P_i must be a straight line on π . Moreover, let β be the plane of the lifted face of P_i . Actually, the velocities of the points on P_i , as defined, are all proportional to the distance between p and the line r of intersection of π with β . Thus, r is the instantaneous screw axis of P_i . Let $\mathbf{S}_{i,0}$ be a Plücker coordinate vector for P_i . For any panel P_j adjacent to P_i through a hinge $\mathbf{H}_{i,j}$ it will be

$$\mathbf{S}_{i,0} - \mathbf{S}_{j,0} = \omega_{i,j} \mathbf{H}_{i,j} \quad (6.2)$$

for some scalar $\omega_{i,j}$, since the three lines of $\mathbf{S}_{i,0}$, $\mathbf{S}_{j,0}$ and $\mathbf{H}_{i,j}$ are concurrent to a point (see Theorem 5.8). This means that, as defined, the assigned screw axes $\mathbf{S}_{i,0}$ are an instantaneous motion of $\mathcal{F}^{ph}(\mathcal{D})$. By Theorem 6.1 the scalars $\omega_{i,j}$ directly give a motion assignment. If f_i and f_j are two adjacent faces of \mathcal{P} , and β_i , β_j are their respective planes,

Figure 6.3: A motion assignment on $\mathcal{F}^{ph}(\mathcal{D})$.

we observe that:

- If $\beta_i = \beta_j$, then the lines of $\mathbf{S}_{i,0}$ and $\mathbf{S}_{j,0}$ coincide, meaning that the only scalar $\omega_{i,j}$ that satisfies Equation 6.2 is zero.
- If $\beta_i \neq \beta_j$, then the lines $\mathbf{S}_{i,0}$ and $\mathbf{S}_{j,0}$ are different and $\mathbf{S}_{i,0} - \mathbf{S}_{j,0}$ is non-null in Equation 6.2, meaning that $\omega_{i,j} \neq 0$,

which proves the theorem. □

The converse of this theorem is also true:

Theorem 6.3 (Whiteley, 1982). *For a line drawing \mathcal{D} with the incidence structure of a polysurface, and its associated panel-and-hinge framework $\mathcal{F}^{ph}(\mathcal{D})$, if $\mathcal{F}^{ph}(\mathcal{D})$ has a motion assignment, then \mathcal{D} is a correct projection of a polysurface \mathcal{P} , with different planes for adjacent faces on each edge where $\omega_{i,j} \neq 0$.*

Proof. We verify the theorem by constructing a lifted polysurface \mathcal{P} of \mathcal{D} . If $\mathcal{F}^{ph}(\mathcal{D})$ has a motion assignment, then there are scalars $\omega_{i,j}$ assigned to the hinges of $\mathcal{F}^{ph}(\mathcal{D})$ such that $\sum \omega_{i,j} \cdot \mathbf{H}_{i,j} = 0$ for every cycle of panels and hinges in $\mathcal{F}^{ph}(\mathcal{D})$. Thus, by Theorem 6.1 we can find an instantaneous motion of $\mathcal{F}^{ph}(\mathcal{D})$ that assigns a screw axis $\mathbf{S}_{i,0}$ to every panel P_i . These axes will all lie on the plane π of the drawing, since their 2-extensors are all linear combinations of the 2-extensors $\mathbf{H}_{i,j}$ of the hinges. Hence, every vertex of the framework is instantaneously moving with a velocity vector that is orthogonal to π . The tips of these vectors are now taken as the liftings of the corresponding vertices of \mathcal{D} . As defined, for each face of \mathcal{D} its lifted vertices will all be coplanar, since their height is proportional to the distance between the vertex and the corresponding screw axis. Also, along a hinge $\mathbf{H}_{i,j}$ articulating two panels P_i and P_j the velocity vectors induced by $\mathbf{S}_{i,0}$ and $\mathbf{S}_{j,0}$ will be exactly the same, meaning that the lifted faces for P_i and P_j coincide along their common edge. We conclude that the tips of the velocity vectors provide a lifting of \mathcal{D} .

The defined lifting will be sharp if every two adjacent panels, say P_i and P_j , receive non-coincident planes. Since $\mathbf{S}_{i,0} - \mathbf{S}_{j,0} = \omega_{i,j} \mathbf{H}_{i,j}$, this happens when $\omega_{i,j} \neq 0$. \square

Now, we can gather all scalars $\omega_{i,j}$ for all edges of \mathcal{D} into a tuple $\omega = \{\dots, \omega_{i,j}, \dots\} \in \mathbb{R}^e$, where e is the number of edges of the drawing. If we collect all vector equations $\sum \omega_{i,j} \cdot \mathbf{H}_{i,j} = 0$ for all cycles of $\mathcal{F}^{ph}(\mathcal{D})$, we can express them in matrix form as

$$\underline{\mathbf{R}}_{\mathcal{D}} \cdot \omega = 0, \quad (6.3)$$

where $\underline{\mathbf{R}}_{\mathcal{D}}$ is called the *rigidity matrix* and contains as many rows as cycles in $\mathcal{F}^{ph}(\mathcal{D})$, and as many columns as there are hinges in $\mathcal{F}^{ph}(\mathcal{D})$. Let $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ denote the kernel of $\underline{\mathbf{R}}_{\mathcal{D}}$. With these definitions and Theorems 6.2 and 6.3 we finally have a set of necessary and sufficient conditions for \mathcal{D} to be realizable.

Theorem 6.4 (Realizability of \mathcal{D}). *A line drawing \mathcal{D} with the incidence structure of a polysurface has a sharp lifting if, and only if, there is some vector $\omega \in Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ all of whose coordinates are different from zero.*

Proof. (\Rightarrow) If \mathcal{D} is realizable, there exists a sharp lifting \mathcal{P} with all pairs of adjacent faces lying on different planes. By Theorem 6.2 \mathcal{P} defines a motion assignment with non-null scalars $\omega_{i,j}$ on all hinges of $\mathcal{F}^{ph}(\mathcal{D})$, and these scalars are the required non-null coordinates of a vector ω of $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$.

(\Leftarrow) If there is a vector $\omega \in Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ with all coordinates $\omega_{i,j}$ different from zero, by Theorem 6.3 every pair of adjacent faces will lie on different planes of the spatial lifting. \square

This theorem is of central importance, as it directly solves the realizability problem in an efficient way. Indeed, one can check that there exists a vector $\omega \in Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ with all its coordinates different from zero by first computing a basis $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ of $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$, arranging these vectors as columns of a matrix

$$\begin{pmatrix} u_{11} & u_{21} & \dots & u_{p1} \\ u_{12} & u_{22} & \dots & u_{p2} \\ \dots & \dots & \dots & \dots \\ u_{1e} & u_{2e} & \dots & u_{pe} \end{pmatrix}$$

and then checking that there are no null rows in it.

6.2.2 Topologic Considerations

It is not necessary to put an equation in $\underline{\mathbf{R}}_{\mathcal{D}} \cdot \omega = 0$, for every cycle of panels of $\mathcal{F}^{ph}(\mathcal{D})$. Actually, if we regard $\mathcal{F}^{ph}(\mathcal{D})$ as a graph G where the vertex and edge set are the panels and hinges of $\mathcal{F}^{ph}(\mathcal{D})$, respectively, it suffices to put a cycle equation for every independent cycle of G . It is a well-known result of Graph Theory that the set of all cycles of a graph has the structure of a vector space. Then, a set of *independent cycles* is just defined as

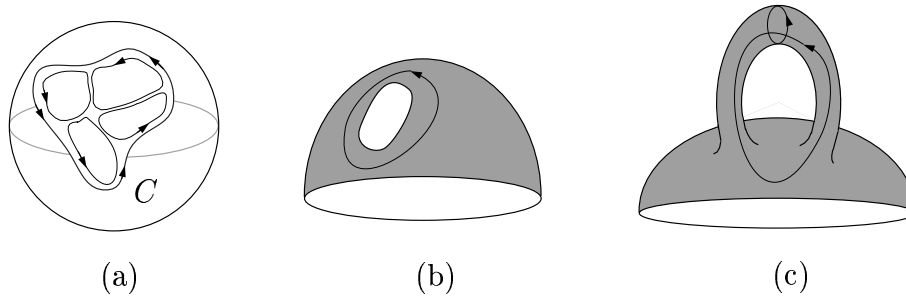


Figure 6.4: Independent cycle equations.

a basis of this vector space [9, 8]. For example, suppose that the line drawing has the incidence structure of a polysphere. Then, we only need an equation $\sum \omega_{i,j} \cdot \mathbf{H}_{i,j} = 0$ for every cycle of panels and hinges around a vertex. The equations of other larger cycles are dependent on these. This can be seen with the aid of Figure 6.4a. The equation of a large cycle C on the polysphere can be obtained as a sum of cycle equations around individual vertices inside C , because the cycles are oriented and the hinges not in C appear two times in the sum, with opposite signs, and their terms $\omega_{i,j} \mathbf{H}_{i,j}$ are cancelled. The same reasoning justifies that if \mathcal{D} has the structure of a polydisk, cycle equations around the vertices will suffice for the test.

For other topologies, though, extra cycle equations are required. Clearly, if \mathcal{D} depicts a polydisk with holes, one cycle equation will be needed around each hole (Figure 6.4b). If the depicted polysurface has handles, then two extra equations per handle are needed, involving the cycles in Figure 6.4c.

6.3 Reconstruction by Sweeping the Kernel

An important corollary of the previous results which directly solves the reconstruction problem, is the following.

Corollary 6.1 (Reconstruction of \mathcal{D}). *There is a one-to-one correspondence between vectors $\omega \in Ker(\mathbf{R}_{\mathcal{D}})$ and the liftings of \mathcal{D} .*

Proof. By Theorem 6.1 there is a one-to-one correspondence between a vector ω of motion assignments and instantaneous motions $\mathbf{S}_{i,0}$ of $\mathcal{F}^{ph}(\mathcal{D})$. Different screw axes $\mathbf{S}_{i,0}$ induce different velocity vectors on the panel P_i and hence different liftings of its vertices. \square

Consequently, we need only to sweep $Ker(\mathbf{R}_{\mathcal{D}})$ to generate all liftings of \mathcal{D} : from a motion assignment we compute its instantaneous motion, which gives a screw for each panel. The join product of this screw and a point P on the panel gives the motion of P . This motion will be a velocity orthogonal to the plane of the drawing, and its Z -coordinate will provide the height of P in the lifting. If the starting motion assignment is written as a linear combination of a basis of $Ker(\mathbf{R}_{\mathcal{D}})$, then this process provides a linear parameterization

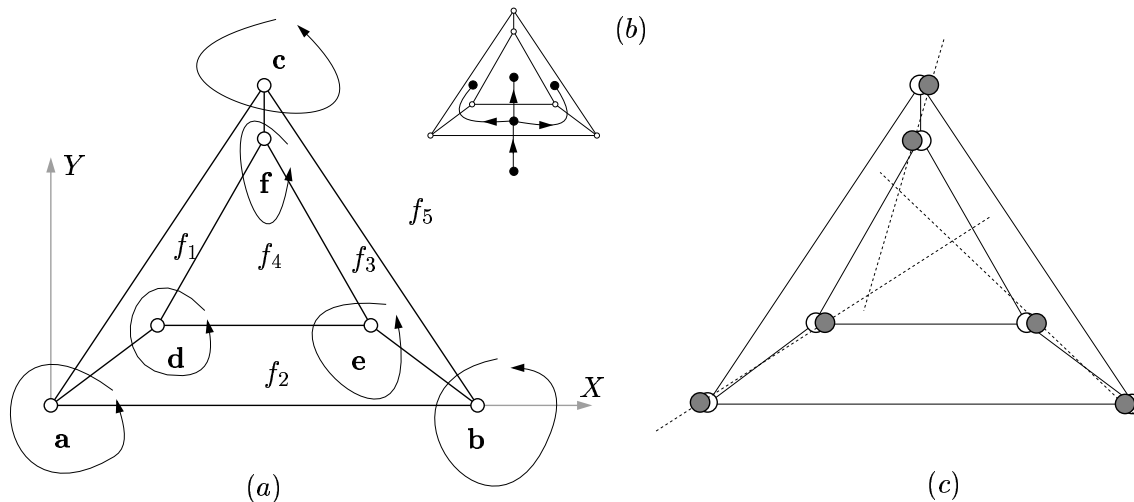


Figure 6.5: Testing a truncated tetrahedron. (a) Cycles involved. (b) Paths to compute the instantaneous motions. (c) A perturbation that slightly moves all vertices takes the drawing out of this realizable configuration. The proximity to a correct drawing, though, can be detected with the singular value decomposition of \mathbf{R}_D .

of all liftings of \mathcal{D} . Next section applies this process to an example, and Section 6.5 shows how it can be implemented in floating-point arithmetic, avoiding superstrictness problems via the singular value decomposition.

6.4 An Example

The coordinates of the truncated tetrahedron in Figure 6.5 are

$$\begin{aligned} \mathbf{a} &= (0, 0, 0, 1), & \mathbf{d} &= (4, 3, 0, 1), \\ \mathbf{b} &= (16, 0, 0, 1), & \mathbf{e} &= (12, 3, 0, 1), \\ \mathbf{c} &= (8, 12, 0, 1), & \mathbf{f} &= (8, 10, 0, 1). \end{aligned}$$

The 2-extensors of the hinges $\mathbf{H}_{i,j}$ between each pair of faces f_i and f_j are:

$$\begin{aligned} \mathbf{H}_{1,2} = \mathbf{a} \vee \mathbf{d} &= \begin{pmatrix} -4 \\ -3 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & \mathbf{H}_{2,3} = \mathbf{b} \vee \mathbf{e} &= \begin{pmatrix} 4 \\ -3 \\ 0 \\ 0 \\ 0 \\ 48 \end{pmatrix}, & \mathbf{H}_{3,1} = \mathbf{c} \vee \mathbf{f} &= \begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ -16 \end{pmatrix}, \\ \mathbf{H}_{1,4} = \mathbf{d} \vee \mathbf{f} &= \begin{pmatrix} -4 \\ -7 \\ 0 \\ 0 \\ 0 \\ 16 \end{pmatrix}, & \mathbf{H}_{2,4} = \mathbf{e} \vee \mathbf{d} &= \begin{pmatrix} 8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 24 \end{pmatrix}, & \mathbf{H}_{3,4} = \mathbf{f} \vee \mathbf{e} &= \begin{pmatrix} -4 \\ 7 \\ 0 \\ 0 \\ 0 \\ -96 \end{pmatrix}, \end{aligned}$$

$$\mathbf{H}_{1,5} = \mathbf{c} \vee \mathbf{a} = \begin{pmatrix} 8 \\ 12 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{H}_{2,5} = \mathbf{a} \vee \mathbf{b} = \begin{pmatrix} -16 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{H}_{3,5} = \mathbf{b} \vee \mathbf{c} = \begin{pmatrix} 8 \\ -12 \\ 0 \\ 0 \\ 0 \\ 192 \end{pmatrix}.$$

The rigidity matrix will have the following block form, where each zero entry is a column sextuple of zeroes:

$$\underline{\mathbf{R}}_{\mathcal{D}} = \begin{pmatrix} -\mathbf{H}_{1,2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_{1,5} & -\mathbf{H}_{2,5} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_{2,3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_{2,5} & -\mathbf{H}_{3,5} \\ \mathbf{0} & \mathbf{0} & -\mathbf{H}_{3,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{H}_{1,5} & \mathbf{0} & \mathbf{H}_{3,5} \\ \mathbf{H}_{1,2} & \mathbf{0} & \mathbf{0} & -\mathbf{H}_{1,4} & \mathbf{H}_{2,4} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{2,3} & \mathbf{0} & \mathbf{0} & -\mathbf{H}_{2,4} & \mathbf{H}_{3,4} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_{3,1} & \mathbf{H}_{1,4} & \mathbf{0} & -\mathbf{H}_{3,4} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

If we substitute the hinges, we get the following matrix. Here, we omit the third, fourth and fifth components of the sextuples $\mathbf{H}_{i,j}$ since they are null when the hinges lie on the XY plane:

$$\underline{\mathbf{R}}_{\mathcal{D}} = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 8 & 16 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 & 0 & 0 & -16 & -8 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 12 \\ 0 & -48 & 0 & 0 & 0 & 0 & 0 & 0 & -192 \\ 0 & 0 & 0 & 0 & 0 & 0 & -8 & 0 & 8 \\ 0 & 0 & -2 & 0 & 0 & 0 & -12 & 0 & -12 \\ 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 192 \\ -4 & 0 & 0 & 4 & 8 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -16 & 24 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & -8 & -4 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 48 & 0 & 0 & -24 & -96 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & -7 & 0 & -7 & 0 & 0 & 0 \\ 0 & 0 & -16 & 16 & 0 & 96 & 0 & 0 & 0 \end{pmatrix}.$$

Using, for example, the singular value decomposition of $\underline{\mathbf{R}}_{\mathcal{D}}$ (see the next section) we can see that its rank is 8, and that a basis of its 1-dimensional kernel is the vector

$$\omega = (-8, -8, -24, \frac{-24}{7}, \frac{-16}{7}, \frac{-24}{7}, 2, 1, 2). \quad (6.4)$$

Clearly, no component of ω is null and, consequently, \mathcal{D} has a sharp lifting. If μ is a free parameter, the motion assignments have the form

$$\begin{aligned}\omega &= (\omega_{1,2}, \omega_{2,3}, \omega_{3,1}, \omega_{1,4}, \omega_{2,4}, \omega_{3,4}, \omega_{1,5}, \omega_{2,5}, \omega_{3,5}) \\ &= \mu \left(-8, -8, -24, \frac{-24}{7}, \frac{-16}{7}, \frac{-24}{7}, 2, 1, 2\right).\end{aligned}$$

According to the proof of Theorem 6.1, we compute the motion of a Panel P_i by choosing a path from a reference panel P_k to P_i and then applying Equation 6.1. Choosing the paths in Figure 6.5b, where the background panel is taken as the reference frame, we get:

$$\begin{aligned}\mathbf{S}_{2,5} &= \omega_{2,5} \mathbf{H}_{2,5}, \\ \mathbf{S}_{1,5} &= \omega_{2,5} \mathbf{H}_{2,5} + \omega_{1,2} \mathbf{H}_{1,2}, \\ \mathbf{S}_{3,5} &= \omega_{2,5} \mathbf{H}_{2,5} - \omega_{2,3} \mathbf{H}_{2,3}, \\ \mathbf{S}_{4,5} &= \omega_{2,5} \mathbf{H}_{2,5} - \omega_{2,4} \mathbf{H}_{2,4}.\end{aligned}$$

That is,

$$\begin{aligned}\mathbf{S}_{2,5} &= \begin{pmatrix} -16 \mu \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & \mathbf{S}_{1,5} &= \begin{pmatrix} 16 \mu \\ 24 \mu \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \\ \mathbf{S}_{3,5} &= \begin{pmatrix} 16 \mu \\ -24 \mu \\ 0 \\ 0 \\ 0 \\ 384 \end{pmatrix}, & \mathbf{S}_{4,5} &= \begin{pmatrix} \frac{16}{7} \mu \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{384}{7} \mu \end{pmatrix}.\end{aligned}$$

Then, the motion of each vertex \mathbf{v} of \mathcal{D} is computed from the screw of a panel P_l incident with \mathbf{v} as $M_{l,5}(\mathbf{v}) = \mathbf{S}_{l,5} \vee \mathbf{v}$. The third coordinate of $M_{l,5}(\mathbf{v})$ gives the Z coordinate of this vertex on the lifted polysurface.

$$\begin{aligned}Z_a &= 0, \\ Z_b &= 0, \\ Z_c &= 0, \\ Z_d &= 48 \mu, \\ Z_e &= 48 \mu, \\ Z_f &= 32 \mu.\end{aligned}$$

Finally, we can check that every quadrilateral face lifts coplanarly to 3-space verifying that the determinant of its four lifted vertices is identically zero. Also, every edge must have non coplanar incident planes and the determinant of the two end-vertices of the edge

and two other points, each on an adjacent face, must not be identically null. For example, for face $abcd$ and edge fc :

$$\det_{abcd} = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 4 & 3 & 48 & \mu \\ 8 & 10 & 32 & \mu \\ 8 & 12 & 0 & 1 \end{vmatrix} = 0, \quad \det_{fcab} = \begin{vmatrix} 8 & 10 & 32 & \mu \\ 8 & 12 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 16 & 0 & 0 & 1 \end{vmatrix} = 6144 \mu.$$

6.5 Testing Realizability in Floating-Point Arithmetic

Given that line drawings are seldom generically realizable, a common issue that any realizability test must face is that of superstrictness (see Sections 3.1.3 and 3.1.4). That is, even when a drawing is correct, any small perturbations introduced in the coordinates will take the vertices out from a realizable configuration. In practice, this means that it is difficult to implement a realizability test on a computer using floating-point arithmetic, because small round-off errors will produce such perturbations, and there is low probability that a correct drawing would be properly classified. However, the fact that it suffices to analyse the kernel of a matrix to determine a drawing's correctness, allows an easy way around this problem, using the singular value decomposition of $\mathbf{R}_{\mathcal{D}}$. We briefly introduce the concept and follow its application with an example. Additional material on the singular value decomposition can be found in [29, Chapter 9] and [127, Chapter I.10].

The singular value decomposition (or SVD for short) provides a powerful technique for dealing with sets of equations or matrices that are either singular or else numerically very close to singular. SVD methods are based on the following theorem of linear algebra.

Theorem 6.5. *Any $m \times n$ matrix \mathbf{A} whose number of rows m is greater than or equal to its number of columns n , can be written as the product of an $m \times n$ column-orthogonal matrix \mathbf{U} , an $n \times n$ diagonal matrix \mathbf{W} with positive or zero elements w_1, \dots, w_n , the singular values, and the transpose of an $n \times n$ orthogonal matrix \mathbf{V} ,*

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times n} \mathbf{W}_{n \times n} \mathbf{V}_{n \times n}^t,$$

such that \mathbf{U} and \mathbf{V} have orthonormal columns. That is, if \mathbf{I} is the identity matrix, $\mathbf{U}^t \mathbf{U} = \mathbf{I}$ and $\mathbf{V}^t \mathbf{V} = \mathbf{I}$.

The SVD can also be carried out when $m < n$. In this case, the singular values w_{m+1}, \dots, w_n , are all zero, and the corresponding columns of \mathbf{U} are also zero.

The SVD explicitly constructs orthonormal bases for the kernel and the image space of a matrix. Specifically, the columns of \mathbf{U} whose same-numbered elements w_j are nonzero are an orthonormal basis spanning the image space. The columns of \mathbf{V} whose same-numbered elements w_j are zero are an orthonormal basis for the kernel. Hence, the rank of the matrix is equal to the number of non-zero singular values in \mathbf{W} . These properties can be used to tell whether the rigidity matrix $\mathbf{R}_{\mathcal{D}}$ is singular or close to singular, indicating the proximity of the vertices of \mathcal{D} to those of a realizable drawing, say \mathcal{D}^* . The number

of close-to-zero singular values will indicate the dimension of $Ker(\mathbf{R}_{\mathcal{D}^*})$, and the columns of \mathbf{V} corresponding to these small singular values will be a reasonable approximation of a basis of $Ker(\mathbf{R}_{\mathcal{D}^*})$. Let us see this with an example.

Suppose we perturb the X -coordinates of the correct line drawing in Figure 6.5a, as shown in Figure 6.5c, letting the vertices be as follows:

$$\begin{aligned} \mathbf{a} &= (-0.01, 0, 0, 1), & \mathbf{d} &= (4.01, 3, 0, 1), \\ \mathbf{b} &= (15.99, 0, 0, 1), & \mathbf{e} &= (12.01, 3, 0, 1), \\ \mathbf{c} &= (8.01, 12, 0, 1), & \mathbf{f} &= (7.99, 10, 0, 1). \end{aligned}$$

We get an incorrect drawing with the hinges:

$$\mathbf{H}_{1,2} = \mathbf{a} \vee \mathbf{d} = \begin{pmatrix} -4.02 \\ -3 \\ 0 \\ 0 \\ 0 \\ -0.03 \end{pmatrix}, \quad \mathbf{H}_{2,3} = \mathbf{b} \vee \mathbf{e} = \begin{pmatrix} 3.98 \\ -3 \\ 0 \\ 0 \\ 0 \\ 47.97 \end{pmatrix}, \quad \mathbf{H}_{3,1} = \mathbf{c} \vee \mathbf{f} = \begin{pmatrix} 0.02 \\ 2 \\ 0 \\ 0 \\ 0 \\ -15.78 \end{pmatrix},$$

$$\mathbf{H}_{1,4} = \mathbf{d} \vee \mathbf{f} = \begin{pmatrix} -3.98 \\ -7 \\ 0 \\ 0 \\ 0 \\ 16.13 \end{pmatrix}, \quad \mathbf{H}_{2,4} = \mathbf{e} \vee \mathbf{d} = \begin{pmatrix} 8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 24 \end{pmatrix}, \quad \mathbf{H}_{3,4} = \mathbf{f} \vee \mathbf{e} = \begin{pmatrix} -4.02 \\ 7 \\ 0 \\ 0 \\ 0 \\ -96.13 \end{pmatrix},$$

$$\mathbf{H}_{1,5} = \mathbf{c} \vee \mathbf{a} = \begin{pmatrix} 8.02 \\ 12 \\ 0 \\ 0 \\ 0 \\ 0.12 \end{pmatrix}, \quad \mathbf{H}_{2,5} = \mathbf{a} \vee \mathbf{b} = \begin{pmatrix} -16 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{H}_{3,5} = \mathbf{b} \vee \mathbf{c} = \begin{pmatrix} 7.98 \\ -12 \\ 0 \\ 0 \\ 0 \\ 191.88 \end{pmatrix},$$

which yield the following rigidity matrix (again, zero elements of $\mathbf{H}_{i,j}$ are omitted):

$$\mathbf{R}_{\mathcal{D}} = \begin{pmatrix} 4.02 & 0 & 0 & 0 & 0 & 0 & 8.02 & 16.00 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & 0 \\ 0.03 & 0 & 0 & 0 & 0 & 0 & 0.12 & 0 & 0 \\ 0 & -3.98 & 0 & 0 & 0 & 0 & 0 & -16.00 & -7.98 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 12 \\ 0 & -47.97 & 0 & 0 & 0 & 0 & 0 & 0 & -191.88 \\ 0 & 0 & -0.02 & 0 & 0 & 0 & -8.02 & 0 & 7.98 \\ 0 & 0 & -2 & 0 & 0 & 0 & -12 & 0 & -12 \\ 0 & 0 & 15.78 & 0 & 0 & 0 & -0.12 & 0 & 191.88 \\ -4.02 & 0 & 0 & 3.98 & 8.00 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ -0.03 & 0 & 0 & -16.13 & 24.00 & 0 & 0 & 0 & 0 \\ 0 & 3.98 & 0 & 0 & -8.00 & -4.02 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 47.97 & 0 & 0 & -24.00 & -96.13 & 0 & 0 & 0 \\ 0 & 0 & 0.02 & -3.98 & 0 & 4.02 & 0 & 0 & 0 \\ 0 & 0 & 2 & -7 & 0 & -7 & 0 & 0 & 0 \\ 0 & 0 & -15.78 & 16.13 & 0 & 96.13 & 0 & 0 & 0 \end{pmatrix}.$$

After a SVD of $\mathbf{R}_{\mathcal{D}}$, we obtain a decomposition $\mathbf{R}_{\mathcal{D}} = \mathbf{U}_{18 \times 9} \mathbf{W}_{9 \times 9} \mathbf{V}_{9 \times 9}^t$, and the singular values:

$$\begin{aligned} \mathbf{W}(1, 1) &= 274.5898, & \mathbf{W}(4, 4) &= 31.4712, & \mathbf{W}(7, 7) &= 12.7619, \\ \mathbf{W}(2, 2) &= 142.9735, & \mathbf{W}(5, 5) &= 24.8156, & \mathbf{W}(8, 8) &= 4.3301, \\ \mathbf{W}(3, 3) &= 52.6653, & \mathbf{W}(6, 6) &= 18.1010, & \mathbf{W}(9, 9) &= 0.03207. \end{aligned}$$

We observe that $\mathbf{W}(9, 9)$ is quite small, revealing that $\mathbf{R}_{\mathcal{D}}$ is close to singular and near a configuration with a 1-dimensional kernel. The ninth column of \mathbf{V} is a reasonable approximation of a vector spanning this kernel:

$$(-0.2948, -0.2903, -0.8821, -0.1265, -0.0852, -0.1235, 0.07428, 0.03642, 0.07258).$$

Since no component of this vector is close to zero, compared to the rest, we conclude that the input drawing, though incorrect, is near a configuration with sharp liftings. The reader can check that this vector and that of Equation 6.4 are almost aligned.

6.6 Enumerating Concavities and Convexities

Traditionally, there has been a remarkable interest not only in getting a parameterization of all liftings, but also in determining the possible shapes of the edges around a lifted vertex, telling whether they are concave or convex, as seen from the center of projection. See Section 3.1.1 for an introduction to this topic. From the following result, we will see

that all $\{+, -\}$ -labellings can be found by simply enumerating the sign combinations that the coordinates of the vectors $\omega \in Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ can have, and we will provide a constructive method to this end. In addition, this will yield a simple model for the Necker reversal phenomenon .

Proposition 6.1. *For a given motion assignment of $\mathcal{F}^{ph}(\mathcal{D})$, if $\omega_{i,j} > 0$ then the edge between faces f_i and f_j is concave in the corresponding lifting of \mathcal{D} . If $\omega_{i,j} < 0$ this edge is convex. If $\omega_{i,j} = 0$, then f_i and f_j are coplanar.*

Proof. Let R be a point on the drawing, on the edge e between faces f_i and f_j . Let P and Q be two points, the first on f_i and the second on f_j , both lying on a line through R , orthogonal to e . Suppose that the distance from R to P is the same that from R to Q . The lifted positions for P , Q , and R , are given by the velocities that P , Q and R have under the motion assignment (Figure 6.6). If these velocities are respectively $v(P)$, $v(Q)$ and $v(R)$, then e is concave if, and only if, $v(R) < \frac{v(P)+v(Q)}{2}$, or $2v(R) < v(P) + v(Q)$. Since the velocities are the first three coordinates of the motion extensor, this is equivalent to

$$2\mathbf{M}_{j,0}(R)]_3 < (\mathbf{M}_{i,0}(P) + \mathbf{M}_{j,0}(Q))]_3,$$

where the notation “ $]_3$ ” indicates the third coordinate of the 4-tuple to which it applies. Expliciting the screws $\mathbf{M}_{j,0}(R)$, $\mathbf{M}_{i,0}(P)$ and $\mathbf{M}_{j,0}(Q)$, this is equivalent to

$$(2\mathbf{S}_{j,0} \vee \mathbf{r})]_3 < (\mathbf{S}_{i,0} \vee \mathbf{p} + \mathbf{S}_{j,0} \vee \mathbf{q})]_3,$$

but, by the composition of motions, we have $\mathbf{M}_{i,0}(P) = \omega_{i,j}\mathbf{H}_{i,j} \vee \mathbf{p} + \mathbf{S}_{j,0} \vee \mathbf{p}$, and thus

$$(2\mathbf{S}_{j,0} \vee \mathbf{r})]_3 < (\omega_{i,j}\mathbf{H}_{i,j} \vee \mathbf{p} + \mathbf{S}_{j,0} \vee \mathbf{p} + \mathbf{S}_{j,0} \vee \mathbf{q})]_3,$$

and reordering the terms:

$$(\mathbf{S}_{j,0} \vee (2\mathbf{r} - \mathbf{p} - \mathbf{q}))]_3 < (\omega_{i,j}\mathbf{H}_{i,j} \vee \mathbf{p})]_3.$$

Now, since $\mathbf{r} = \frac{\mathbf{p}+\mathbf{q}}{2}$, the left hand side of this inequality is zero:

$$0 < (\omega_{i,j}\mathbf{H}_{i,j} \vee \mathbf{p})]_3.$$

This means that the edge e is concave if, and only if, the relative velocity of P with respect to panel P_j has a positive third coordinate. This happens if, and only if, the line bound vector of $\omega_{i,j}\mathbf{H}_{i,j}$ has the same direction as the line bound vector of $\mathbf{H}_{i,j}$ (see Figure 6.6), which happens whenever $\omega_{i,j}$ is positive. With an analogous reasoning, e is convex if, and only if, $\omega_{i,j}$ is negative. \square

It is clear now why $\{+, -\}$ -labellings come in pairs. As the signs of a motion assignment directly indicate the edge types, whenever a lifting exists corresponding to a given vector $\omega \in Ker(\mathcal{R}_{\mathcal{D}})$, a mirror lifting exists with the edge types reversed, since $-\omega$ also lies in $Ker(\mathcal{R}_{\mathcal{D}})$. We give an algorithm to enumerate these pairs in next section.

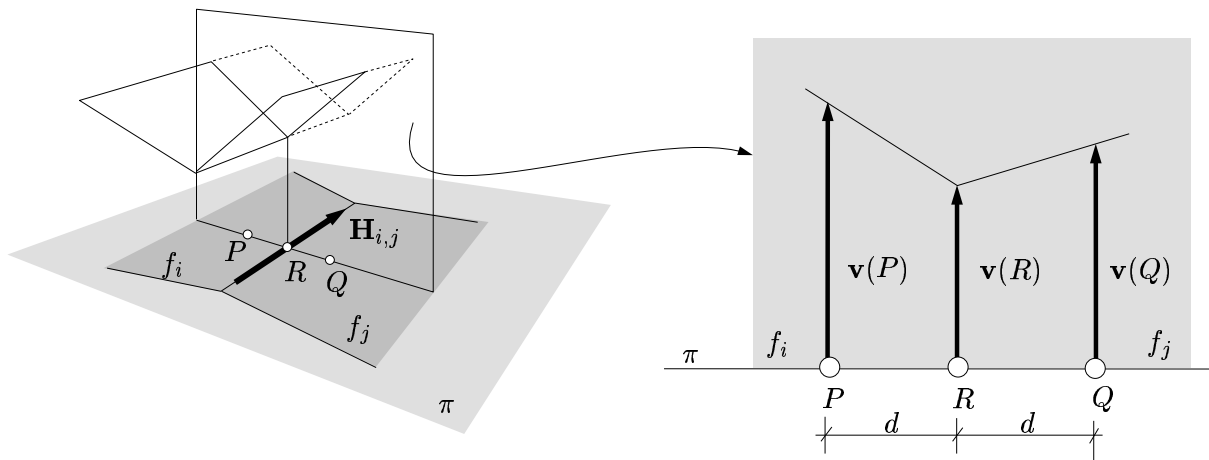


Figure 6.6: $\omega_{i,j} > 0$ implies that the edge between f_i and f_j is concave. If $\omega_{i,j} < 0$ the edge is convex.

6.6.1 Feasible Sign Vectors

Let α_i denote the i -th coordinate hyperplane of \mathbb{R}^e . That is, $\alpha_i = \{\omega \in \mathbb{R}^e : \omega_i = 0\}$. We can orient each of these hyperplanes and define the sets

$$\begin{aligned}\alpha_i^+ &= \{\omega \in \mathbb{R}^e : \omega_i > 0\} \\ \alpha_i^- &= \{\omega \in \mathbb{R}^e : \omega_i < 0\}\end{aligned}$$

to be the *positive* and *negative* side of α_i , respectively. If $e = 3$, there are three coordinate planes and \mathbb{R}^3 is partitioned into eight connected regions called octants. In general, \mathbb{R}^e is divided by e coordinate hyperplanes and, by an abuse of language, we will also refer to the 2^e connected regions in this partition as *octants*. For convenience it is useful to regard the octants as open sets, not containing the points on their limiting hyperplanes. Each octant O of \mathbb{R}^e can be characterized with a sign vector $s(O) \in \{+1, -1\}^e$. Namely, the i -th component of $s(O)$ will be “+1” if the points of O lie on the positive side of α_i , or “-1” otherwise. Clearly, the sign vectors of all octants containing points of $\text{Ker}(\underline{\mathbf{R}}_{\mathcal{D}})$ provide all consistent $\{+, -\}$ -labellings of \mathcal{D} . Before going into details, let us give a rough overview of the method employed to enumerate them.

Let p be the dimension of $\text{Ker}(\underline{\mathbf{R}}_{\mathcal{D}})$ and, to simplify, assume that $\text{Ker}(\underline{\mathbf{R}}_{\mathcal{D}})$ is not contained in any hyperplane α_i . Then, the intersection of $\text{Ker}(\underline{\mathbf{R}}_{\mathcal{D}})$ with each α_i is a linear subspace β_i of dimension $p - 1$ (Figure 6.7, left). We can regard $\text{Ker}(\underline{\mathbf{R}}_{\mathcal{D}})$ as the p -dimensional vector space \mathbb{R}^p , instead of as a subspace of \mathbb{R}^e , which can be done by constructing a bijective linear mapping \mathbf{s} between points of $\text{Ker}(\underline{\mathbf{R}}_{\mathcal{D}})$ and points of \mathbb{R}^p :

$$\mathbf{s} : \text{Ker}(\underline{\mathbf{R}}_{\mathcal{D}}) \longrightarrow \mathbb{R}^p.$$

This mapping converts the subspaces β_i of \mathbb{R}^e into $(p - 1)$ -dimensional hyperplanes γ_i of \mathbb{R}^p . The collection of all γ_i defines a partition of \mathbb{R}^p into several convex cones, with their apex at the origin (Figure 6.7, right). Clearly, each cone corresponds to an octant

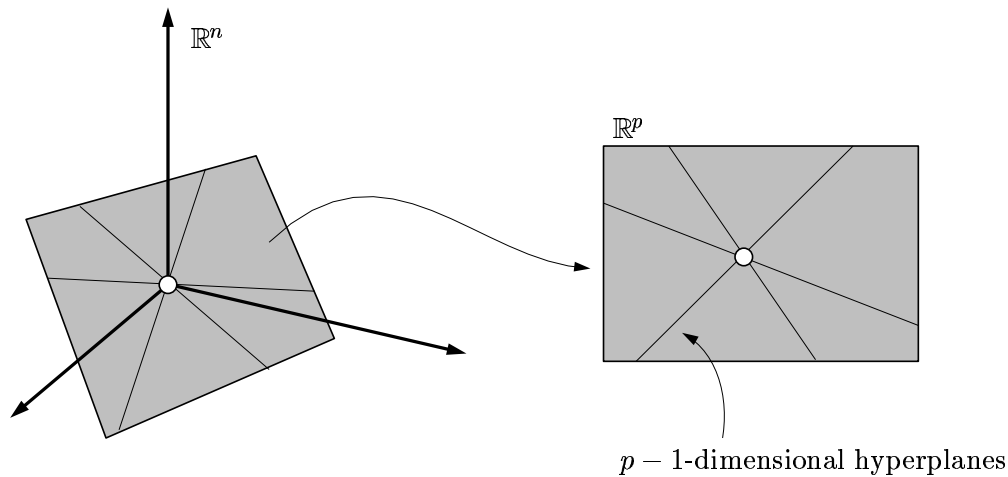


Figure 6.7: Feasible $\{+, -\}$ -labellings correspond to cells of an arrangement \mathcal{A}^* of hyperplanes in \mathbb{R}^p .

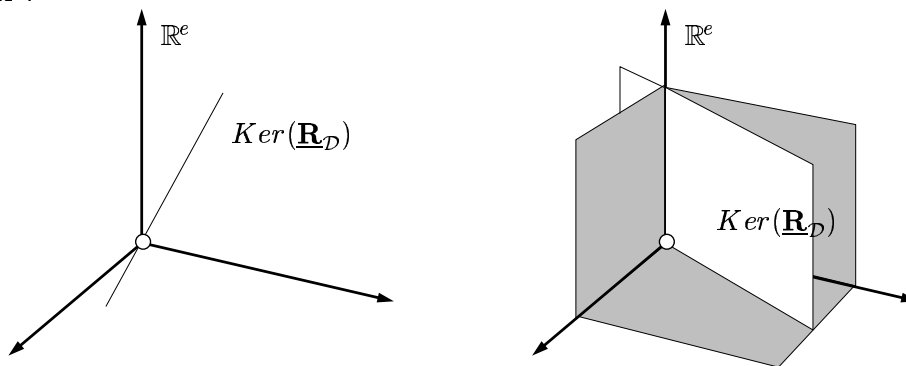


Figure 6.8: Two different coordinate hyperplanes can yield the same intersection with $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$.

of \mathbb{R}^e intersected by $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ and, hence, after this transformation the problem reduces to the enumeration of all these cones. This is an instance of a well-known problem in Computational Geometry : the enumeration of all cells of the spatial partition induced by a collection of hyperplanes. Several efficient algorithms are known to this end, but before we give one in next section we examine the previous steps with further detail.

Without loss of generality, we can assume that $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ is not contained in any coordinate hyperplane $\omega_i = 0$ and, thus, that $\beta_i = Ker(\underline{\mathbf{R}}_{\mathcal{D}}) \cap \alpha_i$ is a linear subspace of dimension $p-1$. (If $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ was contained in k coordinate hyperplanes, say $\alpha_{i_1}, \dots, \alpha_{i_k}$, then the scalars $\omega_{i_1}, \dots, \omega_{i_k}$ would be zero in all motion assignments ω , and the corresponding pairs of faces would lift coplanarly to 3-space in all liftings of \mathcal{D} .) In this case, we can always apply the remaining analysis considering $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$ as part of the lower dimensional space \mathbb{R}^{e-k} (instead of \mathbb{R}^e) made up with all coordinates of \mathbb{R}^e , except $\omega_{i_1}, \dots, \omega_{i_k}$.

Note that in general several coordinate hyperplanes can give rise to the same subspace β_i , when intersected with $Ker(\underline{\mathbf{R}}_{\mathcal{D}})$. This can be seen with the examples in Figure 6.8: if

$e = 3$ and $p = 1$, $Ker(\mathbf{R}_{\mathcal{D}})$ is a line through the origin of \mathbb{R}^3 and all subspaces β_i are a single point; if $e = 3$ and the kernel is a plane containing the Z axis, α_1 and α_2 intersect with this plane over the same line. However, let us assume for the moment that no two coordinate hyperplanes give the same intersection with $Ker(\mathbf{R}_{\mathcal{D}})$.

To compute the equation of each hyperplane $\gamma_i \subset \mathbb{R}^p$, corresponding to every $\beta_i \subset \mathbb{R}^e$, we will employ the following two vector bases:

- A basis $\{v_1^i, v_2^i, \dots, v_{p-1}^i\}$ of β_i . Since β_i is the solution of the following homogeneous linear system, such a basis can be derived, for example, from the singular value decomposition of its matrix:

$$\left. \begin{array}{l} \mathbf{R}_{\mathcal{D}} \cdot \omega = 0 \\ \omega_i = 0 \end{array} \right\}.$$

- An orthogonal basis $\{u_1, \dots, u_p, \dots, u_e\}$ of \mathbb{R}^e where the first p vectors $\{u_1, \dots, u_p\}$ are an orthogonal basis of $Ker(\mathbf{R}_{\mathcal{D}})$. $\{u_1, \dots, u_p\}$ can be obtained from the singular value decomposition of $Ker(\mathbf{R}_{\mathcal{D}})$, while we easily get $\{u_{p+1}, \dots, u_e\}$ by completing the basis $\{u_1, \dots, u_p\}$ to one of \mathbb{R}^e using the Gram-Schmidt orthogonalization algorithm.

Moreover, we will use the following two matrices:

$$\mathbf{S} = \begin{pmatrix} u_{11} & u_{21} & \dots & u_{e1} \\ u_{12} & u_{22} & \dots & u_{e2} \\ \dots & \dots & \dots & \dots \\ u_{1e} & u_{2e} & \dots & u_{ee} \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \\ 0 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{pmatrix}.$$

Since \mathbf{S} has the vectors $\{u_1, \dots, u_e\}$ as columns, its inverse \mathbf{S}^{-1} converts a vector in the canonical basis of \mathbb{R}^e to the same vector in the basis $\{u_1, \dots, u_e\}$. \mathbf{P} is an $e \times p$ matrix where the upper $p \times p$ block is the identity, and the remaining entries are all zero.

Now, using \mathbf{S}^{-1} we can express the vectors $\{v_1^i, v_2^i, \dots, v_{p-1}^i\}$ in the basis $\{u_1, \dots, u_e\}$. After such a transformation, the last $e - p$ coordinates of the resulting vectors will be clearly zero, since the $\{u_{p+1}, \dots, u_e\}$ are all orthogonal to $Ker(\mathbf{R}_{\mathcal{D}})$. If for each such vector we drop out these last $e - p$ zero coordinates, we get a basis $\{w_1^i, \dots, w_{p-1}^i\}$ of the hyperplane $\gamma_i \subset \mathbb{R}^p$. In sum, the matrix $\mathbf{S}^{-1} \cdot \mathbf{P}^t$ defines the desired mapping $\mathbf{s} : Ker(\mathbf{R}_{\mathcal{D}}) \rightarrow \mathbb{R}^p$, and each vector w_j^i is obtained from the v_j^i as

$$w_j^i = \mathbf{S}^{-1} \mathbf{P}^t v_j^i.$$

Finally, with the Gram-Schmidt method we can find a vector $w_p^i \in \mathbb{R}^p$ that is simultaneously orthogonal to all vectors $\{w_1^i, \dots, w_{p-1}^i\}$, which provides the desired equation for γ_i , since

$$\gamma_i = \{x \in \mathbb{R}^p : w_p^i \cdot x = 0\}.$$

Each γ_i divides \mathbb{R}^p in two half-spaces, γ_i^+ and γ_i^- , defined as follows:

$$\begin{aligned}\gamma_i^+ &= \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}_p^i \cdot \mathbf{x} > 0\}, \\ \gamma_i^- &= \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}_p^i \cdot \mathbf{x} < 0\}.\end{aligned}$$

The collection \mathcal{H} of all hyperplanes γ_i induces a decomposition of \mathbb{R}^p into several p -dimensional cells, called the *hyperplane arrangement* of \mathcal{H} , denoted $\mathcal{A}(\mathcal{H})$. Each cell is a convex cone through the origin and, clearly, every octant intersected by $\text{Ker}(\mathbf{R}_{\mathcal{D}})$ is in one-to-one correspondence with one such cone. Like the octants of \mathbb{R}^e , every cone C of $\mathcal{A}(\mathcal{H})$ can be characterized by a sign vector $s(C) \in \{+1, -1\}^q$, where q is the number of hyperplanes in \mathcal{H} . The i -th coordinate of $s(C)$ will be “+1” if the points of C lie in γ_i^+ , or “-1” if they lie in γ_i^- . Thus, the labelling problem reduces to the enumeration of all sign vectors of the cones in $\mathcal{A}(\mathcal{H})$ for which a standard algorithm to construct hyperplane arrangements will be used. Before we give it, we need to consider the following issues.

All algorithms for enumerating the cells in $\mathcal{A}(\mathcal{H})$ require that all hyperplanes in \mathcal{H} be different. But we have seen that several hyperplanes α_i can yield the same intersection with $\text{Ker}(\mathbf{R}_{\mathcal{D}})$. In general, there will be $q \leq e$ different subspaces β_i , which we will index from 1 to q as β_1, \dots, β_q . To these subspaces, there will correspond also a collection \mathcal{H} of q different hyperplanes γ_i , indexed as $\gamma_1, \dots, \gamma_q$. Then, while the sign vector of an octant of \mathbb{R}^e has e signs, the sign vector of a cone of $\mathcal{A}(\mathcal{H})$ will have $q \leq e$ signs, and we need a way to tell the signs of the first given the signs of the second. This is given by the following.

Proposition 6.2. *Let C be a cone of $\mathcal{A}(\mathcal{H})$ and O its corresponding octant of \mathbb{R}^e . Let γ_k be the hyperplane of \mathcal{H} that corresponds to α_j , i.e. γ_k is the transformed subspace of $\beta_k = \alpha_j \cap \text{Ker}(\mathbf{R}_{\mathcal{D}})$. Then, the j -th coordinate of $s(O)$ is*

$$s_j(O) = \begin{cases} +1 & \text{if } \mathbf{S} \mathbf{P} s_k(C) \mathbf{w}_p^k \in \alpha_j^+, \\ -1 & \text{if } \mathbf{S} \mathbf{P} s_k(C) \mathbf{w}_p^k \in \alpha_j^-, \end{cases}$$

where $s_k(C)$ is the k -th coordinate of $s(C)$.

Proof. $s_j(O)$ is positive (respectively negative) if the points on C correspond to points on the positive (resp. negative) side of α_j . This can be checked by selecting a point \mathbf{p} of C , transforming it back to \mathbb{R}^e by computing $\mathbf{S} \mathbf{P} \mathbf{p}$, and testing whether the j -th coordinate of the resulting vector is positive (resp. negative). One such point is $\mathbf{p} = s_k(C) \mathbf{w}_p^k$. \square

6.6.2 Constructing a Hyperplane Arrangement

This section describes the *incremental construction*, a standard method for constructing an arrangement. See [38] for a recent survey on this and other techniques related to construction of arrangements and their applications.

We begin with some basic background. A p -cell in an arrangement $\mathcal{A}(\mathcal{H})$, is a maximal p -dimensional connected region of \mathbb{R}^p , not intersected by any hyperplane in \mathcal{H} . A k -cell in

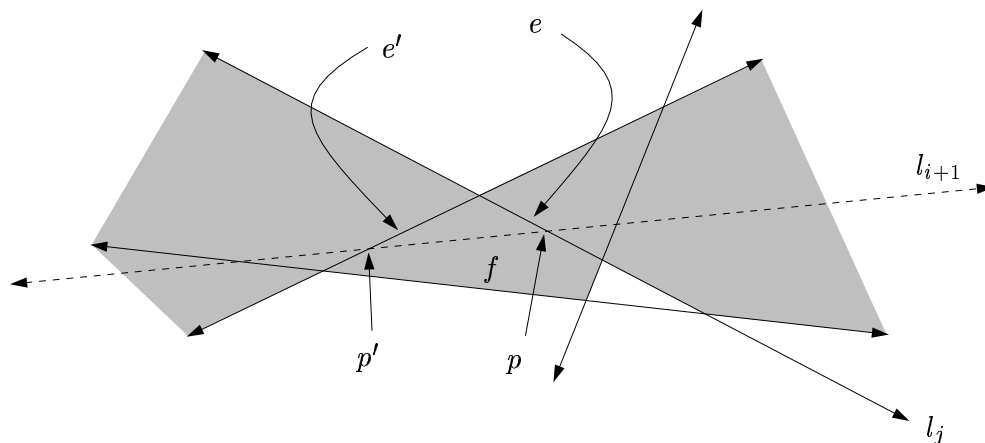


Figure 6.9: Adding the line l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$. The shaded region is the zone of l_{i+1} in the arrangement of the other four lines.

$\mathcal{A}(\mathcal{H})$, for $0 \leq k \leq p-1$, is a maximal k -dimensional connected region in the intersection of a subset of the hyperplanes in \mathcal{H} that is not intersected by any other hyperplane in \mathcal{H} . It follows that any cell in an arrangement is convex. An arrangement is called *simple* if every p hyperplanes in \mathcal{H} meet in a single point and if any $p+1$ hyperplanes have no point in common. There are cells with special names: a *vertex*, an *edge*, a *face* and a *facet* are the 0, 1, 2 and $(p-1)$ -dimensional cells of $\mathcal{A}(\mathcal{H})$, respectively. If C is a curve of \mathbb{R}^p , the *zone of C* is the collection of p -cells of $\mathcal{A}(\mathcal{H})$ intersected by C .

To construct $\mathcal{A}(\mathcal{H})$ we need a way to represent the cells in it. A usual tool is the *cell-tuple structure*, which consists of an *incidence graph* plus some order information between the cells. The incidence graph (sometimes called the *facial lattice*) of $\mathcal{A}(\mathcal{H})$ is a graph $G = (V, E)$ where there is a node in V for every k -cell of $\mathcal{A}(\mathcal{H})$, and an arc between two nodes if the corresponding cells are incident to one another. While the incidence graph captures all the cells in the arrangement, and their connectivity, it misses *order* information between cells. For example, there is a natural ordering among the edges that appear along the boundary of a face in a planar arrangement. This ordering can be generalized to any k -dimensional convex cell G . This information together with G define the cell-tuple structure [13], which is a generalization of the *quad-edge structure* of Guibas and Stolfi [35] for the planar case.

The incremental construction starts with an empty space \mathbb{R}^p and proceeds by adding one hyperplane after the other to the arrangement, while maintaining (a representation of) the arrangement of the objects added so far. This approach yields an optimal-time algorithm for arrangements of hyperplanes. We describe it next for a collection $\mathcal{L} = \{l_1, l_2, \dots, l_n\}$ of n lines in the plane, assuming that the arrangement is simple. The same approach extends to higher dimensions. See [26, Chapter 7] for the details.

Let \mathcal{L}_i denote the set $\{l_1, \dots, l_i\}$. At stage $i+1$ we add l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$. We maintain the quad-edge representation for $\mathcal{A}(\mathcal{L}_i)$, so that in addition to the

incidence information, we also have the order of edges along the boundary of each face. The addition of l_{i+1} is carried in two steps: (i) we find a point p of intersection between l_{i+1} and an edge of $\mathcal{A}(\mathcal{L}_i)$, and split that edge into two, and (ii) we walk along l_{i+1} from p to the left (assuming l_{i+1} is not vertical) updating $\mathcal{A}(\mathcal{L}_i)$ as we go; we then walk along l_{i+1} from p to the right completing the construction of $\mathcal{A}(\mathcal{L}_{i+1})$. See Figure 6.9.

Finding an edge of $\mathcal{A}(\mathcal{L}_i)$ that l_{i+1} intersects can be done in $O(i)$ time by choosing one line l_j from \mathcal{L}_i and checking all the edges of $\mathcal{A}(\mathcal{L}_i)$ that lie on l_j for intersection with l_{i+1} . This intersection point p lies on an edge e that borders two faces of $\mathcal{A}(\mathcal{L}_i)$. We split e into two edges at p . Next, consider the face f intersected by the part of l_{i+1} to the left of p . Using the order information, we walk along the edges of f away from p and we check for another intersection p' of l_{i+1} with an edge e' on the boundary of f . At the intersection we split e' into two edges, we add an edge to the arrangement for the portion pp' of l_{i+1} , and we move to the face on the other (left) side of e' . Once we are done with the faces of $\mathcal{A}(\mathcal{L}_i)$ crossed by l_{i+1} to the left of p , we go back to p and walk to the other side. This way we visit all the faces of the zone of l_{i+1} in $\mathcal{A}(\mathcal{L}_i)$, as well as some of its edges. The amount of time spent is proportional to the number of edges we visit, and hence bounded by the complexity of the zone. The following theorem gives the complexity of the algorithm (see [26, Chapter 7] for a proof).

Theorem 6.6. *If \mathcal{H} is a set of n hyperplanes in \mathbb{R}^d such that $\mathcal{A}(\mathcal{H})$ is a simple arrangement, then $\mathcal{A}(\mathcal{H})$ can be constructed in $\Theta(n^d)$ time and space.*

Part III

Correction of Drawings

Chapter 7

Modifying Vertex Positions

“Geometry is the science of correct reasoning on incorrect figures”.

George Pólya —How to Solve It, 1945 [76].

7.1 Introduction

This chapter presents a new algorithm for correcting incorrect line drawings —incorrect projections of a polyhedral scene. There are at least two situations where these incorrect drawings may arise:

- When a single camera is used to generate a 3D map of a polyhedral world. The camera takes an image, on which an edge and vertex detection process is applied, the straight lines are detected and, finally, a raw line drawing is obtained [104, pag. 179] (Figure 7.1).
- When a designer sketches a polyhedral part or scene on a sheet of paper, and he wants to reconstruct it with the aid of a computer. This can be achieved by first scanning the sketch, then processing the image to detect its vertices and edges to produce a raw line drawing (Figure 7.2). There are several past and on-going works pursuing such a man-machine interface for 3D Solid Modelling. See for example the works by Marti et al. [61, 62], Lipson et al. [53, 93], Grimstead et al. [33, 34] and Markowsky et al. [59, 110, 111].

In both cases, we may use the realizability and reconstruction algorithms given in Chapter 6 to recover all 3D shapes that project onto the synthesized drawing. However, before doing so, there is a problem that we must overcome. Note that even when the extracted drawing has the appropriate incidence structure, the coordinates of its vertices will seldom correspond to those of a correct 2D projection. The correct positions are invariably perturbed by the roughness of the hand-sketching process, the digitalization errors, the edge-detection algorithms and so on, and we end up with a drawing whose vertices are usually in generic position (see Section 3.1.3). As most drawings are not generically realizable

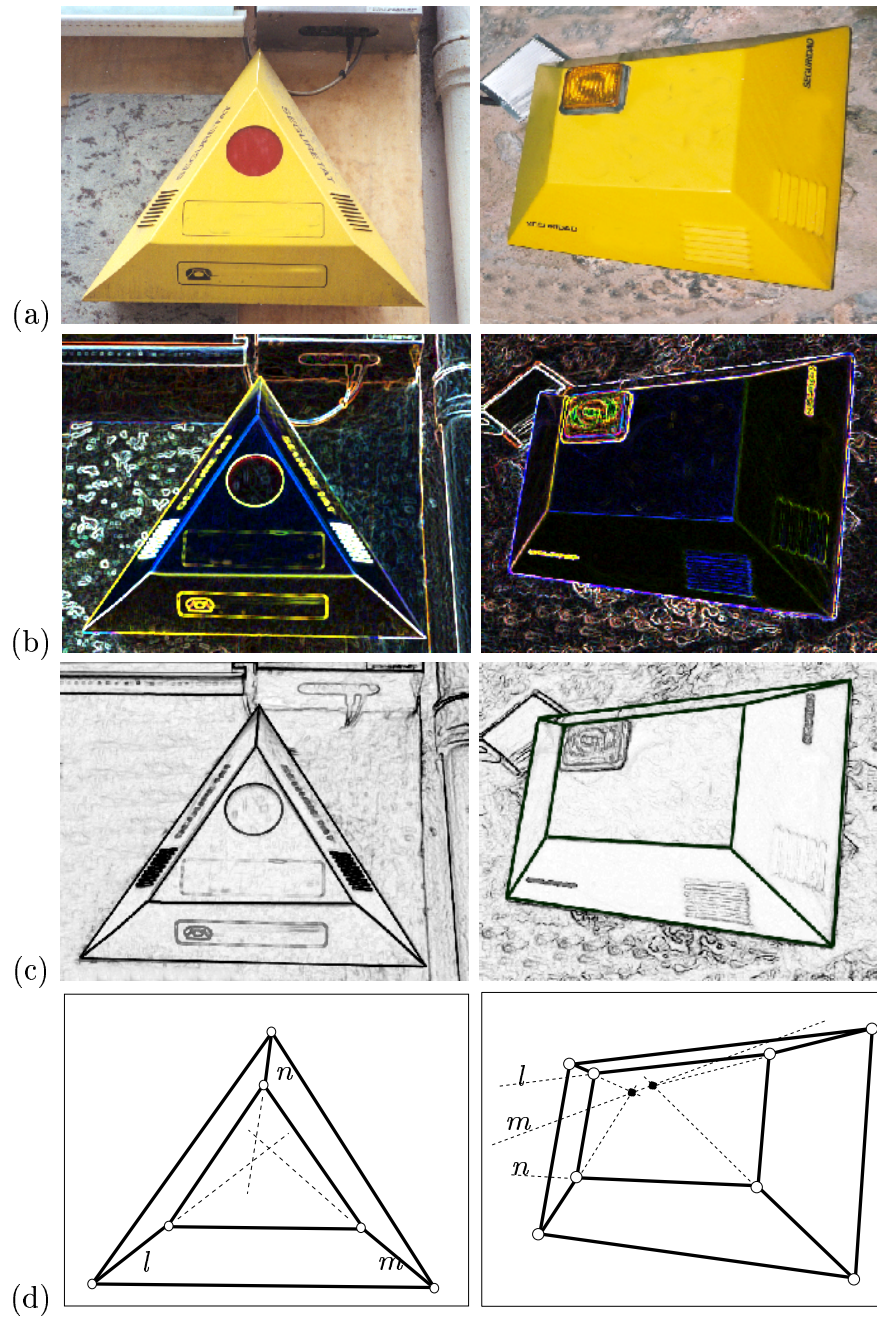


Figure 7.1: A picture of an alarm device (a) can be processed to detect the sharp edges (b), extract the straight lines (c), and derive a line drawing from it (d), which is usually incorrect (the lines l , m and n should be concurrent, according to the conditions in Chapter 4).

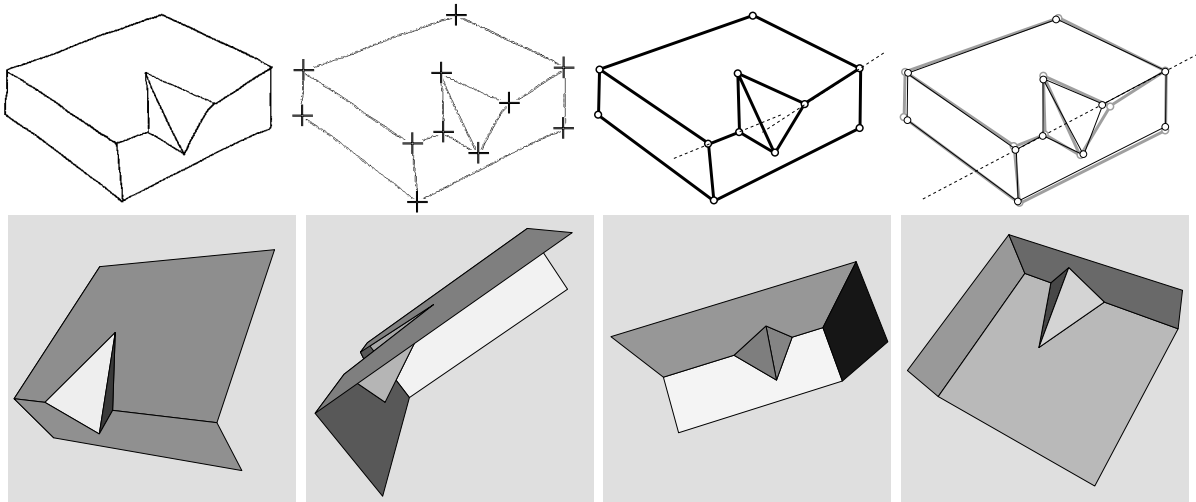


Figure 7.2: A line drawing extracted from a hand-made sketch is seldom correct (top row). Using the correction algorithm, though, we can find the closest correct drawing (top right) and apply the algorithms of Chapter 6 to reconstruct it. The bottom row shows several views of a possible lifting.

and most available algorithms for interpreting them are “superstrict” (see Sections 3.1.3 and 3.1.4), they will judge these noisy diagrams as incorrect and fail to reconstruct a 3-dimensional scene from them. We get around this problem with a new procedure that keeps the incidence structure but moves the positions of all vertices until the closest correct drawing is found. This overcomes the superstrictness problem in a simple way: in order to apply a superstrict method on an incorrect drawing \mathcal{D}^{inc} , the closest correct drawing to it, say \mathcal{D}^{cor} , is first computed. If the vertices on \mathcal{D}^{cor} are too far from those in \mathcal{D}^{inc} (according to a well-defined distance and a given tolerance), \mathcal{D}^{inc} is judged as incorrect, otherwise we accept it as “practically correct” and we can start the reconstruction process from \mathcal{D}^{cor} . With this approach, any superstrict classic method for line drawing interpretation is now practical, as it can be applied to the corrected version of the input drawing (Figure 7.2, top right).

The key contribution of this chapter is a rational parameterization of the class of correct drawings for a given polyhedron (Sections 7.3 and 7.4), which allows to rephrase the correction problem as an *unconstrained* minimization of a rational function (Section 7.2). This minimization can be tackled using a conjugate gradient method which has proven to be very effective in all experiments of our implementation (Section 7.5). Moreover, to avoid falling into local minima, Section 7.6 presents a heuristic algorithm to find a good starting point for the search. Finally, Section 7.7 points out how the technique can be applied to line drawings of polydisks and Section 7.8 summarizes some points for further attention.

7.2 The Overall Algorithm

To simplify, we will deal with drawings produced by orthogonally projecting a *single spherical* polyhedron, onto the XY plane, showing all edges (even the hidden ones). This is not too restrictive, and Section 7.7 explains how to extend the results to drawings of polydisks, possibly with holes on them.

Given an incorrect drawing \mathcal{D}^{inc} , our goal is to obtain a correct one \mathcal{D}^{cor} , with the same incidence structure as \mathcal{D}^{inc} , which is as close to \mathcal{D}^{inc} as possible. As a function measuring the distance between the two drawings, we have chosen the sum of the squared Euclidean distances between pairs of corresponding vertices in \mathcal{D}^{inc} and \mathcal{D}^{cor} .

The problem can be stated as follows. If v_i^{inc} and v_i^{cor} denote, respectively, the 2D coordinates of the i th vertex of \mathcal{D}^{inc} and \mathcal{D}^{cor} , we want to minimize

$$\sum_{i=1}^n \|v_i^{inc} - v_i^{cor}\|^2,$$

subject to the constraint that the vertices v_i^{cor} define a correct drawing \mathcal{D}^{cor} with the same incidence structure as \mathcal{D}^{inc} .

However, we will show that it is possible to parameterize the 2D coordinates of the vertices of all correct drawings with a given incidence structure. More precisely, given an incidence structure $I^{cor} = (V, F, R)$, it is possible to write the coordinates (x_i^{cor}, y_i^{cor}) of every vertex $v_i^{cor} \in V$ as functions

$$\begin{aligned} x_i^{cor} &= \chi_i(p_1, p_2, \dots, p_n), \\ y_i^{cor} &= \psi_i(p_1, p_2, \dots, p_n), \end{aligned}$$

in such a way that any tuple of parameters (p_1, p_2, \dots, p_n) , $p_i \in \mathbb{R}^n - \mathcal{F}$, $\forall i$, where \mathcal{F} is a zero-measure subset of \mathbb{R}^n , fixes a correct drawing. Since χ_i and ψ_i are rational functions, our problem reduces to the unconstrained minimization of the rational function

$$\sum_{i=1}^n \|v_i^{inc} - (\chi_i(p_1, p_2, \dots, p_n), \psi_i(p_1, p_2, \dots, p_n))\|^2,$$

which can be solved numerically by a gradient search from an initial correct drawing that estimates the final solution. Next section presents the *resolvable sequence*, the key concept that leads to this parameterization.

7.3 Resolvable Sequences

Is there a set of independent choices that can be made to construct a polyhedron in a consistent manner? To illustrate this question, let us focus on the simple example of Figure 7.3a, a pyramid with a quadrilateral base. The shape of this polyhedron can be fixed by, for example, giving the coordinates of all its vertices or the face plane coefficients

of all its faces. But care must be taken in any of the two ways. If we arbitrarily fix all face planes, then f_1, f_2, f_3 and f_4 will not probably have a common point of intersection and vertex v_1 will be inconsistently defined. On the contrary, if we arbitrarily fix all vertices, v_2, v_3, v_4 and v_5 need not be coplanar and face f_5 might be inconsistently defined.

In general, we say that a polyhedron is *resolvable* if it is possible to list its vertices and faces in a sequence $\mathcal{S} = (\dots, v_i, \dots, f_j, \dots)$ in such a way that:

- (C1) when a vertex occurs in \mathcal{S} , it is incident to at most three previous faces;
- (C2) when a face occurs in \mathcal{S} , it is incident to at most three previous vertices;
- (C3) when two faces, f and f' , share three or more vertices (Figure 7.3b), f and f' appear earlier in \mathcal{S} than the third of the common vertices;
- (C4) when two vertices v and v' are incident to three or more common faces (Figure 7.3c), both v and v' appear earlier than the third of the common faces.

\mathcal{S} is called a *resolvable sequence* for the polyhedron. Note that if such a sequence exists, then we can construct the polyhedron in a consistent way. We just need to fix its vertices and faces, one by one, following the order in \mathcal{S} . Along the way, when an element is underconstrained by previous choices, additional choices can be taken arbitrarily.

In 1934, Steinitz proved that all polyhedra whose graph of vertices and edges is planar and 3-connected are resolvable [125]. Actually, for these polyhedra it suffices to find a sequence satisfying conditions (C1) and (C2) above as their 3-connectedness ensures they have no face sharing more than two vertices, nor any pair of vertices sharing more than two faces.

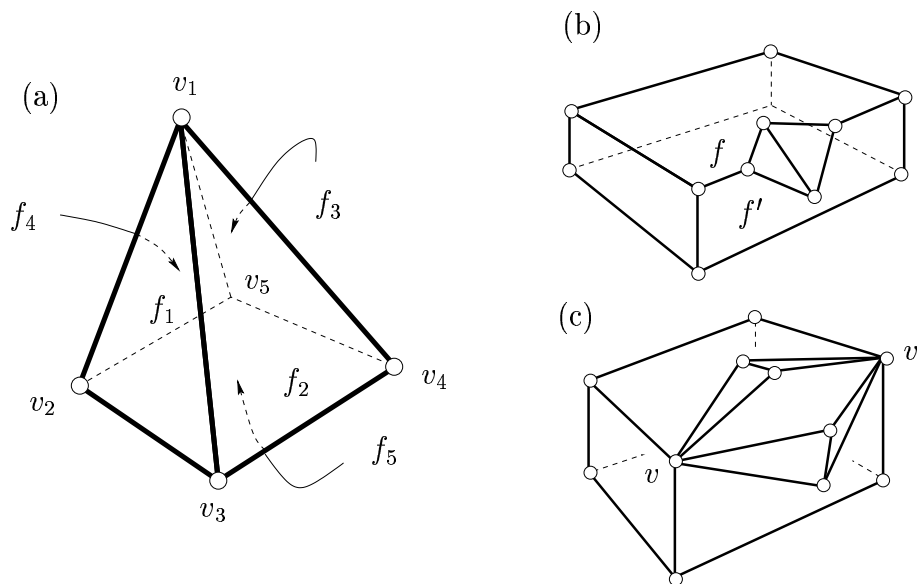


Figure 7.3: (a) A pyramid with a quadrangular base. (b) Two faces sharing more than two vertices. (c) Two vertices sharing more than two faces.

Very recently, in [105] Sugihara has extended Steinitz's results, finding that actually all spherical polyhedra are resolvable. He also gives an algorithm to compute a resolvable sequence in $O(v + f)$ time, where v and f are the number of vertices and faces of the polyhedron, respectively. Moreover the resolvable sequence may not be unique and, naturally, it only depends on the combinatorial structure of the polyhedron at hand.

7.4 Parameterizing Correct Projections

The resolvable sequence induces a parameterization of all polyhedra with a given incidence structure. For example, a trivial resolvable sequence for the truncated tetrahedron in Figure 3.9a is to first list all faces, and then all vertices: $\mathcal{S} = (f_1, \dots, f_5, v_1, \dots, v_6)$. (This is clearly valid for any trihedral polyhedron, one where every vertex has three incident faces.) Thus, here, the coordinates (x_i, y_i, z_i) of every vertex v_i can be written as functions of the coefficients of its three incident planes by, e.g., solving for x_i, y_i and z_i using Cramer's rule. That is, if these planes are

$$\begin{aligned} A_i x + B_i y + C_i z + D_i &= 0 \\ A_j x + B_j y + C_j z + D_j &= 0 \\ A_k x + B_k y + C_k z + D_k &= 0, \end{aligned}$$

then the coordinates of v are

$$v_x = \frac{1}{\Delta} \begin{vmatrix} -D_i & B_i & C_i \\ -D_j & B_j & C_j \\ -D_k & B_k & C_k \end{vmatrix} \quad v_y = \frac{1}{\Delta} \begin{vmatrix} A_i & -D_i & C_i \\ A_j & -D_j & C_j \\ A_k & -D_k & C_k \end{vmatrix}$$

$$v_z = \frac{1}{\Delta} \begin{vmatrix} A_i & B_i & -D_i \\ A_j & B_j & -D_j \\ A_k & B_k & -D_k \end{vmatrix} \quad \Delta = \begin{vmatrix} A_i & B_i & C_i \\ A_j & B_j & C_j \\ A_k & B_k & C_k \end{vmatrix},$$

and varying these parameters we get different liftings of the truncated tetrahedron.

Note that the resolvable sequence also induces a parameterization of all correct drawings with the given incidence structure, as we only need to project the (parameterized) spatial polyhedron onto the XY plane, keeping the parameterization for the X and Y coordinates of every vertex.

In the general case, we can construct a parameterization of all polyhedra with a given incidence structure I as follows. First, we compute a resolvable sequence \mathcal{S} for I . Then, we visit every element of \mathcal{S} , following the order of the sequence. If the element is a face, then:

- If it is not incident to any previous vertex, there is total freedom in choosing its position and the four coefficients of its plane are free parameters.
- If it is incident to three previous vertices, the parameters of the plane are totally fixed and no new parameter is introduced.

- If it is incident to two previous vertices, say p and q , we must select one of all the planes meeting the segment pq . Such a plane can be written as:

$$\begin{vmatrix} p_x & q_x & r_x & x \\ p_y & q_y & r_y & y \\ p_z & q_z & r_z & z \\ 1 & 1 & 1 & 1 \end{vmatrix} = 0,$$

and the three coordinates of a third point $r = (r_x, r_y, r_z)$ are introduced as new parameters.

- If a face is incident to one previous vertex p , its plane can be expressed as

$$(n_x, n_y, n_z) \cdot ((x, y, z) - (p_x, p_y, p_z)) = 0,$$

and the three coordinates of the normal vector (n_x, n_y, n_z) are chosen as parameters.

If the element is a vertex v , then:

- If v is incident to no previous face, there is total freedom in choosing its position and its three coordinates are taken as free parameters.
- If v is incident to three previous faces, the vertex is totally fixed and can be found computing the intersection of the three planes.
- If v is incident to two previous faces, say f_i and f_j , we can write two equations:

$$\begin{aligned} A_i v_x + B_i v_y + C_i v_z + D_i &= 0, \\ A_j v_x + B_j v_y + C_j v_z + D_j &= 0, \end{aligned}$$

and solve them for v_x and v_y in terms of v_z , which is introduced as a new parameter.

- Finally, if v is incident to one previous face, say f , we can freely choose v_x and v_y and get v_z from the equation of f 's plane.

Note that this parameterization is rational, as at each step of its construction we can write a vertex or face coordinate as a quotient of polynomials in the parameters. Although for certain choices of the parameters it may fail to provide a polyhedron (e.g., there is an indetermination when a vertex is incident to three previous faces, and the chosen planes for them are not all distinct), this only happens for a zero-measure subset of the parameter space, posing no problem to the minimization, as the next section explains.

7.5 Implementation and Results

The correction algorithm has been implemented in C for drawings of trihedral polyhedra, as these have the advantage that every vertex position is easily parameterized by the twelve coefficients of its three incident planes.

For the minimization, we use TNPACK, a freely available package specially suited for large-scale problems with possibly thousands of variables [84, 85]. To minimize a function $F(X)$, $X \in \mathbb{R}^n$ TNPACK implements the iterative truncated Newton method, based on minimizing a local quadratic approximation to F at every step. For efficiency, an approximated (truncated) solution of this local minimization is allowed, which is computed through a preconditioned conjugate gradient algorithm.

The user must essentially supply three routines, returning F , its gradient, and the Hessian matrix, evaluated at a given point $X \in \mathbb{R}^n$. For the gradient we directly provide its symbolic expression, as it is easy to derive. For the Hessian matrix, we rely on an (optional) internal TNPACK routine that uses finite differences of the gradient to compute it. To prevent the minimization from falling in a point X of parameter space yielding indetermination (see Section 7.4) the routine computing $F(X)$ is implemented to return a very high value in these configurations.

We have tested the correction process on several drawings of spherical polyhedra: a truncated tetrahedron, a dodecahedron, a truncated icosahedron, a rhombitruncated-cubeoctahedron, and a rhombitruncated-icosidodecahedron (Figure 7.6). The number of optimization variables involved in these examples is 20, 48, 104, 128 and 248, respectively—four times the number of faces. A corrected drawing for them is obtained in less than five seconds of CPU time on a SUN Ultra-80.

For each of these drawings the following experiment has been done. First, a correct

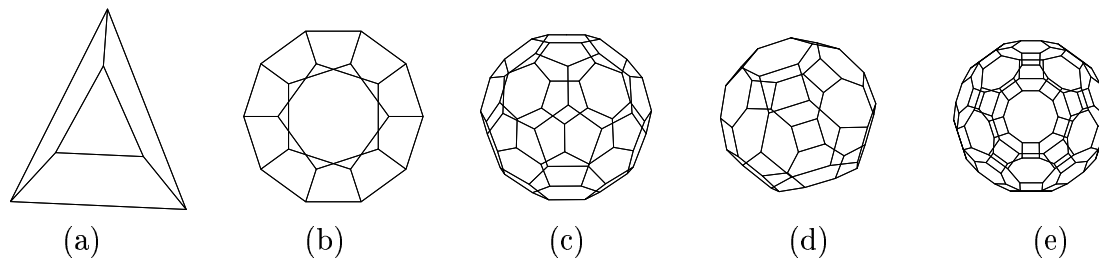


Figure 7.4: Line drawings used as a testbed for the correction algorithm: a truncated tetrahedron (a), a dodecahedron (b), a truncated icosahedron (c), a rhombitruncated cubeoctahedron (d) and a rhombitruncated icosidodecahedron (e).

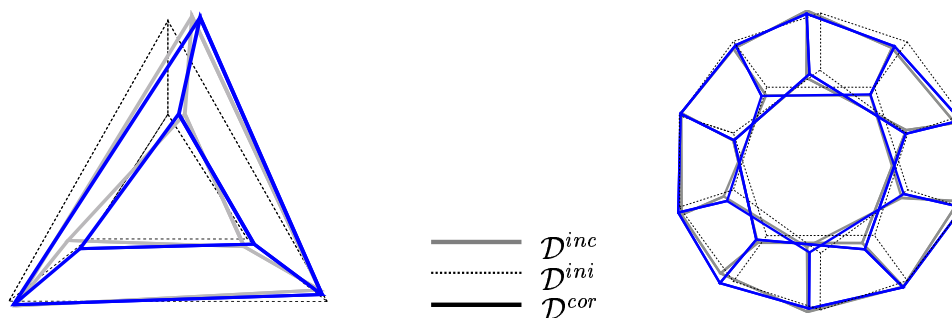


Figure 7.5: Correction of a truncated tetrahedron (left) and of a dodecahedron (right).

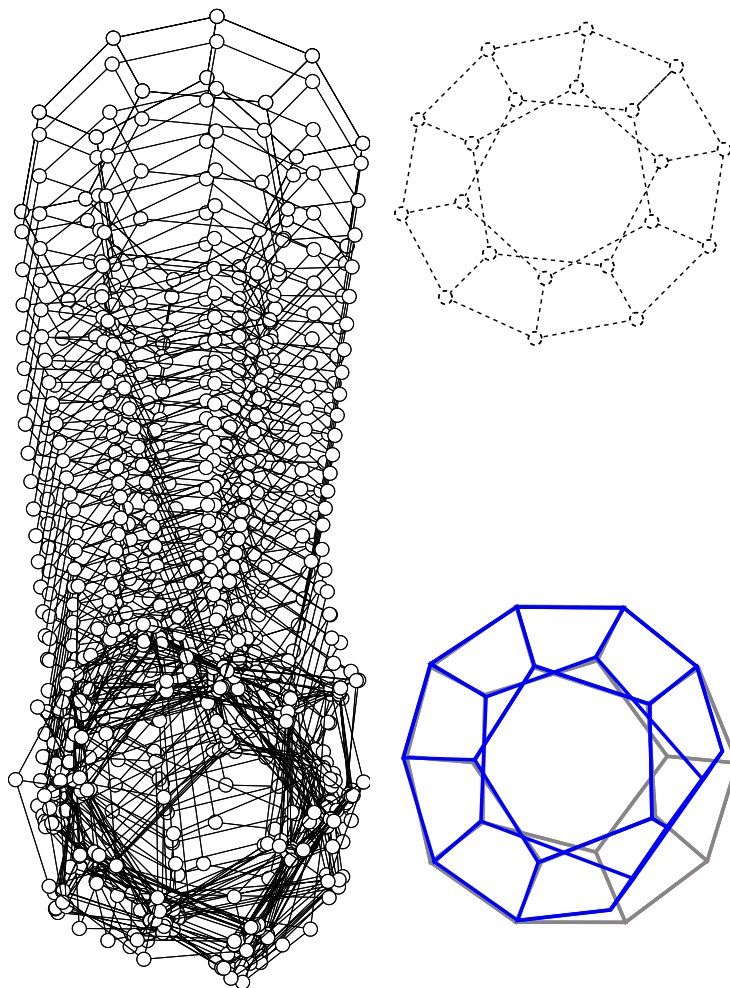


Figure 7.6: A correction sequence of a dodecahedron using a bad starting point (left). The incorrect drawing (in grey lines), the initial estimation (dashed) and the final correction (black) have been singled out to the right. The final correction falls in a bad local minimum because the initial estimation is actually translated and rotated with respect to the input drawing.

drawing \mathcal{D}^{ini} is generated by projecting the spatial polyhedron to a plane. Then, the vertices of \mathcal{D}^{ini} are randomly perturbed to get an incorrect drawing \mathcal{D}^{inc} with the same incidence structure. Finally, the correction algorithm is applied to \mathcal{D}^{inc} , by starting a gradient search from \mathcal{D}^{ini} . The resulting corrected drawing \mathcal{D}^{cor} is shown in Figure 7.5 for the truncated tetrahedron and the dodecahedron.

The gradient search has taken four seconds of CPU time in the toughest case of Figure 7.4, using a SUN Ultra-80. We note that, although simple, these drawings are far more complex than those one can find in the literature [104, 92]. Extensive tests have been done in all cases, starting the gradient search at different initial drawings and the minimization always converged rapidly to a small neighborhood of the incorrect drawing.

However, the used objective function certainly has local minima and one can find

initial drawings \mathcal{D}^{ini} from which the algorithm gets stuck on them. Figure 7.6 gives an example of this undesired behavior. Here, \mathcal{D}^{ini} has been generated by translating the dodecahedron far away from the position of \mathcal{D}^{inc} and rotating it 180 degrees about its center. The sequence to the left shows the path followed by the gradient method. Notice how the vertices follow crossing trajectories, from their origin to the destination, to undo this 180° rotation. As we see, the final correction is a local minimum compared to that of Figure 7.5 right, while the input drawing \mathcal{D}^{inc} is exactly the same in both cases. Clearly, there is a need for feeding the search with a good starting point and next section proposes a method to this end.

7.6 A Good Starting Point

A reasonably good starting drawing can be easily computed using again the resolvable sequence. The idea is to properly place every vertex and face of the sequence, so that the 2D projection is locally close enough to \mathcal{D}^{inc} . Let us see this in detail. We distinguish several situations, depending on whether we are fixing a vertex or a face.

Assume first that we are fixing a vertex v , whose 2D position in the incorrect drawing is v^{inc} . v can be incident to zero, one, two or three previously-fixed faces. In the first case, there is total freedom in choosing v 's spatial position but, to be compliant with the drawing, we choose it to lie in the vertical line over v^{inc} , at any height. If v is incident to just one previous face, then we choose it over this face's plane, in the vertical line at v^{inc} . If v is incident to two faces with planes α and β (respectively), we fix v on the line of intersection of α and β at the place where its 2D projection is at a minimum distance from v^{inc} . Finally, if v is incident with three previous faces, we fix it in the intersection of their respective planes.

On the other hand, if we are fixing a face f , it can be incident to three, two, one or zero previously-fixed vertices. In the first case, there is no choice for the plane of f as it is fully determined by the three vertices. If the face is incident to two fixed vertices, say p and q , we can choose among all the planes meeting the line pq . But, which one? If some of the neighboring faces of f have already been fixed, say faces f_{i_1}, f_{i_2}, \dots , then we would like that the lines of intersection of these faces with f lie reasonably close to the corresponding edges in the incorrect drawing. If we label the common vertices between f and f_{i_1}, f_{i_2}, \dots as w_1, \dots, w_m , and $z(w_i)$ denotes the height of vertex w_i as computed on the plane of its already-fixed face, over its position in the drawing, then a reasonable way to achieve this is to fix f to the plane α that meets the line pq and minimizes the sum of squared residuals:

$$\sum_{i=1}^m (z(w_i) - z_{\alpha}(w_i))^2,$$

where $z_{\alpha}(w_i)$ denotes the z -coordinate of vertex w_i as given by the plane α . Obviously, if no adjacent face of f was previously fixed, we simply fix f at *any* plane meeting pq .

The remaining cases, when f is incident to *one* or to *no* previous vertex are analogous,

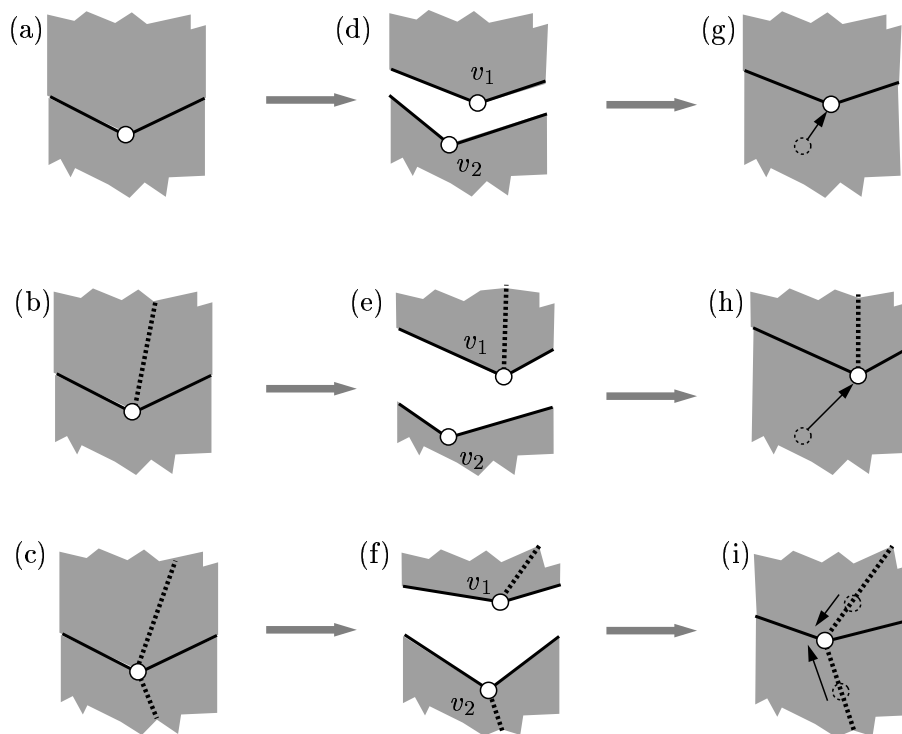


Figure 7.7: Independent correction of neighboring polydisks (left) will separate their boundaries (center) but these can be made coincident by properly moving the vertices again (right) while preserving the overall correctness. Interior and boundary edges are indicated in dashed and solid lines, respectively.

the only difference being that we choose among all the planes meeting a fixed point in the first case, and among all possible planes of 3-space in the second.

Of course, following this strategy, the resulting correct drawing may deviate substantially from \mathcal{D}^{inc} at some vertices, specially if the drawing is large enough. However, from the good convergence behavior seen in the experiments above, we judge this approximation as good enough to avoid local minima.

7.7 Correction of Other Topologies

Real scenes of polyhedra differ substantially from the spherical model assumed in Section 7.2. Hidden edges are not visible when the objects are opaque, and if several objects are present, they may occlude one another (left). As we saw in Section 2.2, when this happens, the incidence structure of the line drawing is not that of a spherical polyhedron anymore. Thus, in practice, the required tool is an algorithm able to correct drawings whose incidence structure is that of a polydisk, maybe with polygonal holes in it. The following observation will give a way to parameterize all possible projections of a polydisk and, hence, a means to apply our correction algorithm to this case.

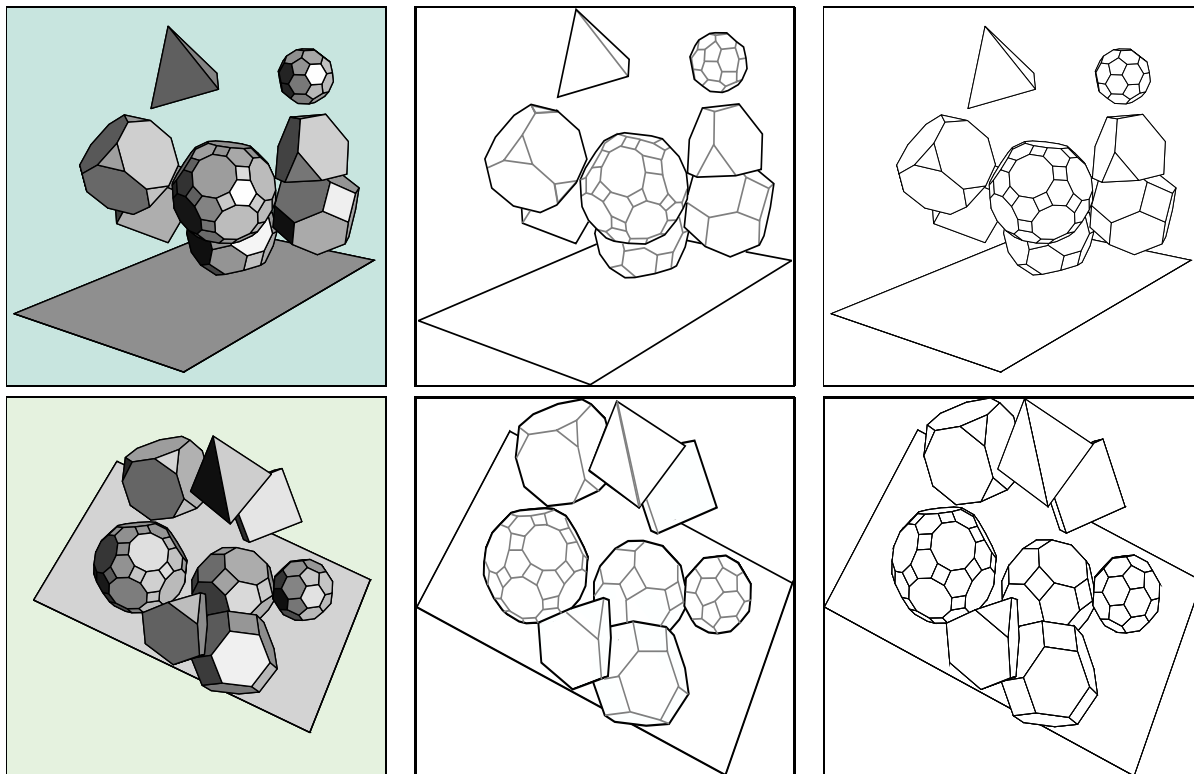


Figure 7.8: Two views of a same polyhedral scene (left) together with two incorrect drawings of them, with boundary edges labelled in black (center), and the final corrections (right).

Given a polydisk \mathcal{P} with an incidence structure $I_{\mathcal{P}} = (V, F, R)$ we can take each of its holes and triangulate it. That is, we can cover each hole with triangles, connecting its boundary vertices¹, to obtain a new plane-faced object \mathcal{P}_{Δ} so that its incidence structure $I_{\mathcal{P}_{\Delta}} = (V, F_{\Delta}, R_{\Delta})$ is that of a spherical polyhedron. Now, we easily see that there is a one-to-one correspondence between the realizations of the incidence structure $I_{\mathcal{P}}$ and those of $I_{\mathcal{P}_{\Delta}}$. Certainly, when we have a realization of $I_{\mathcal{P}}$ we get one of $I_{\mathcal{P}_{\Delta}}$ by adding to F the extra triangles in F_{Δ} and, vice versa, a realization of $I_{\mathcal{P}_{\Delta}}$ is converted to one of $I_{\mathcal{P}}$ by removing the extra triangles in F_{Δ} . Actually, this addition of triangles is innocuous because it does not add any coplanarity constraint to the vertices of \mathcal{P} . But, since $I_{\mathcal{P}_{\Delta}}$ has a spherical topology, we can compute a resolvable sequence for it, which will yield the desired parameterization for the vertices and faces of the original polydisk and its correct projections.

Moreover, note that when several polydisks are present in the drawing, we can treat each one separately in the same way. For this, we only need to have each one of them identified, which can be done by collecting all regions delimited by boundary edges after an edge-labeling algorithm has been applied. However, an issue may arise here. If we

¹Recall that the vertices and edges of these objects are referred to as *boundary* or *interior* according to whether they correspond or not to the boundaries of the spatial objects, as seen from the center of projection.

correct each polydisk separately, the final boundaries of neighboring polydisks may not coincide as they originally did. Depending on the application this disparity might be irrelevant. For example, if all we want is an approximate reconstruction of the objects in the scene, these small errors may be acceptable. On the contrary, if they are not, we propose the following strategy to make the boundaries coincident again. It will only be valid for trihedral scenes, but we note that this is the case that arises when all faces lie on planes in general position.

First, observe that, if the objects are trihedral, only one of the following three situations occurs (figs. 7.7a, b and c, respectively): a boundary vertex either has 1) no incident interior edge, or 2) only one interior edge on one side, or 3) one interior edge on both sides. The correction algorithm will separate the boundaries as depicted in Figure 7.7 d, e and f, yielding two copies of the original vertex, say v_1 and v_2 . In the first two cases, the boundaries can be made coincident again by moving v_2 , the vertex with no incident interior edge, to the position of v_1 . Note that this will not alter the correctness of the polydisk of v_2 (figs. 7.7g and h). In the third case, we can move v_1 and v_2 to the point of intersection of their interior edges without altering the correctness of their respective polydisks (Figure 7.7i).

This strategy has been implemented and Figure 7.8 shows the results on a synthetic polyhedral scene. In this example, the scenes to the left have been projected to yield the drawings in the center, whose vertices have been randomly perturbed to yield the shown incorrect configurations. Boundary edges delimiting each polydisk are marked in thick black lines while interior edges are marked in grey. The resulting correct polydisks are shown in Figure 7.8, right.

7.8 Future Extensions

This Chapter has presented a new approach to correct incorrect projections of polyhedra and has discussed its contributions with respect to the previous method by Sugihara. Surprisingly, our improvements have only been possible thanks to Sugihara's latest finding of the resolvable sequence, offering an unexpected new application of this result. To conclude, it is worth to mention the following points deserving further attention.

Unfortunately, there are topological structures that cannot be corrected. Namely, polyhedra with a genus equal or greater than one do not have a resolvable sequence in general. A simple counterexample is given by the torus in Figure 7.9. Since all vertices are incident to exactly four faces, condition **(C1)** in Section 7.3 will be necessarily violated at some vertex in any sequence. Thus, an interesting open problem would be to characterize the resolvable polyhedra that are not homeomorphic to a sphere or a disk with holes and to construct an algorithm for finding resolvable sequences in such cases.

Although the initial drawing we propose to start the gradient search seems a fairly good approximation of the result, the minimization is still not guaranteed to converge to the global minimum. To mend this up, one can always start the search at several different initial estimations, each derived from a different resolvable sequence of the same incidence

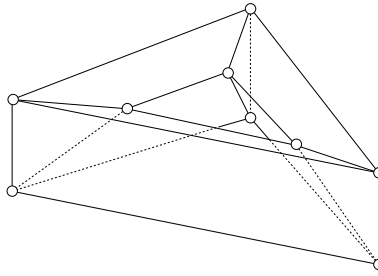


Figure 7.9: An unresolvable torus.

structure, and select the best corrected drawing.

Another possibility could be to derive a polynomial (rather than rational) parameterization, by working in projective instead of affine coordinates, and attempt to find the global minimum through interval arithmetic [39] or Bézier-clipping techniques [10].

Chapter 8

Modifying the Incidence Structure

This chapter presents a second approach for the correction of line drawings. This time, instead of altering the vertex coordinates, we will conveniently modify the incidence structure so that the drawing becomes spatially realizable. To provide enough motivation, the strategy is presented in the context of an application to Solid Modelling, namely to find a compact, minimal and robust representation of solid objects via *trihedral polygonal meshes*, a new type of meshes whose combinatorial structure is dual to that of the usual triangulations. Section 8.1 offers an introduction to this topic, which is joint work with Kokichi Sugihara.

8.1 Introduction

In Computer Graphics, 3D free-form surfaces are commonly approximated by triangular meshes for simplicity reasons. To display such meshes the three Cartesian coordinates of all vertices must be sent to a rendering device. This chapter considers an alternative to this scheme. We propose to represent a surface by a *trihedral polygonal mesh* (one where all vertices have exactly three incident faces) plus some triangles at strategic places. The suggested approximation has several attractive properties. It turns out that the Z-coordinates of its vertices are completely governed by the Z-coordinates assigned to four selected ones. This allows describing the spatial polygonal mesh with just its 2D projection plus the heights of four vertices. Then, since heights of other vertices can be computed from them, rather than be a required input, this ensures a consistent spatial display with cofacial vertices being coplanar, even when perturbations in the input data are introduced. As a consequence, the 2D projections of such meshes essentially capture the “spatial meaning” of the given surface, in the sense that, whatever spatial interpretations are drawn when looking at such projections, all of them are showing essentially the same shape, up to some trivial ambiguities. The resulting 2D projections, though, are not generically realizable, which means that in practice they will be rarely correct. This problem is overcome here by editing their incidence structure, with the addition of new triangular faces at convenient places, until a generically reconstructible line drawing is obtained.

8.1.1 Surface Polygonization

A *polygonal mesh* is a piecewise linear surface made up with planar polygonal patches, glued along the edges, and possibly containing holes. A polygonization method is an algorithm able to construct an approximation—expressed as a polygonal mesh—of a given surface. The literature on polygonization methods, mainly on triangulations, is vast (see [40] for a recent survey on triangulations and algorithms to simplify them). In general the main goal is to obtain approximations which actually converge to the model; that is, methods whose associated errors are zero at the limit. Other goals have been the speed of polygonization and the ability of the polygonizer to satisfy some constraints in the solution (one might request the most accurate approximation possible using a given number of line segments or triangles). In our case, we propose a polygonization able to approximate a wide class of smooth surfaces, with the property that all its vertex heights can be governed by the heights of only four of them.

The polygonization of surfaces has been used in different forms in a large number of applications. Sometimes the original surface exists in the real world and the data has been captured by direct measurement. This is the case of CT scans where the discrete measured data points already define an approximation. In other cases, we require the polygonization of implicit or parametrically defined surfaces for the simple practical reason that we want to display them on a graphics workstation using a Z-buffer hardware. But, in any case, by constructing a mesh that is close to a surface within a known and guaranteed approximation, we begin to understand its shape [86].

8.1.2 Unequivocal Pictures

Consider the picture we get after orthogonally projecting a spatial polygonal mesh onto a plane. Can we totally recover the 3-dimensional mesh this picture represents from just the projection itself? Clearly not, for infinitely many other meshes will generate exactly *the same* projection. For example, for the fully triangulated picture shown in Figure 8.1,

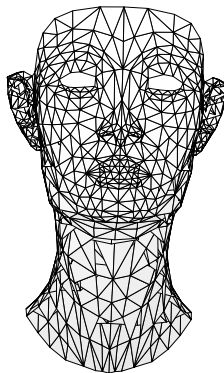


Figure 8.1: A projection of a fully-triangulated mesh.

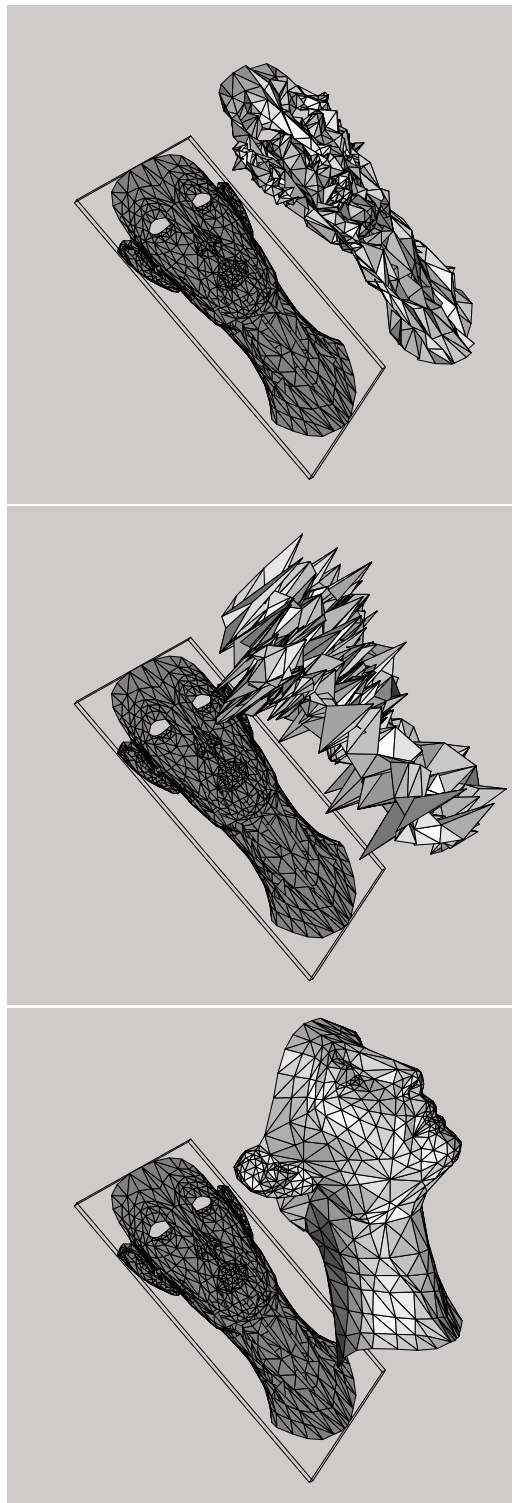


Figure 8.2: Arbitrary liftings of this triangulated drawing have no spatial meaning. But actually, a very specific one of them really does: it shows Nefertiti's face. Can we constrain the plane drawing in such a way that this is the only recovered meaning?

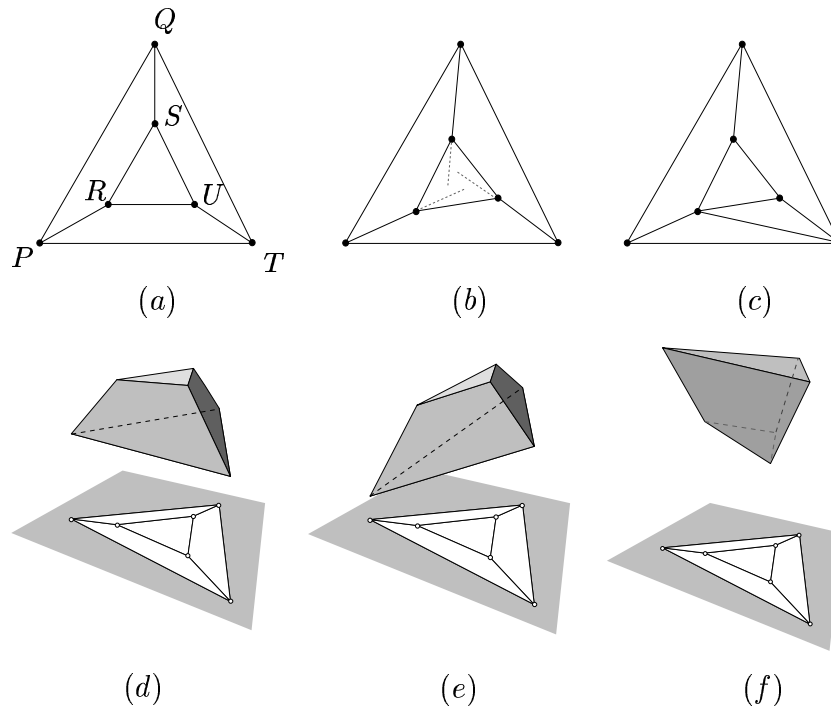


Figure 8.3: A truncated tetrahedron (a) and three possible liftings (d, e, f). The slightest perturbation destroys the correctness of the picture (b), but this can be avoided adding new triangular faces (d).

there are many different *liftings* as illustrated in Figure 8.2.

Here, since any three points in space will be coplanar, to get a lifting we can just give arbitrary heights over the picture plane to all vertices, put a triangular face for every triangle in the drawing, and obtain the spatial mesh. In this case there is a large variety of spatial interpretations for the given picture, not only differing in changes of vertical scale, but also in the many different concavities and convexities that the resulting mesh can have. The first two arbitrary liftings in Figure 8.2 seem to have no meaning at all and just resemble terrain data. But actually, there is a rather “hidden” meaningful interpretation: Nefertiti’s face! Can we constrain this drawing in such a way that this is the only recovered meaning?

This large freedom, or ambiguity, in the spatial interpretation of triangulated pictures is not always permitted. Certainly there is a class of *unequivocal pictures* where just the heights of 4 vertices suffice to determine and “control” the resulting spatial shape. These pictures are unequivocal in the sense that their liftings represent essentially the same object, up to some trivial ambiguities. As an example, Figure 8.3a is an unequivocal representation of a truncated tetrahedron, as seen in Figures 8.3d, e, and f. Observe that indeed it suffices to set the heights of P , Q , T and R to determine those of S and U , using the coplanarity constraints on faces $RPQS$ and $SQTU$.

One of our goals is then to approximate any given surface with a polygonal mesh yielding unequivocal projections that uniquely identify the spatial shape up to the trivial ambiguities produced by changing the heights of only four vertices.

8.1.3 Non-Redundancy, Minimality and Robustness

Actually, unequivocal pictures offer a *non-redundant* description of the mesh they represent, since unnecessary heights of vertices do not need to be transmitted to a rendering device: we just provide a projection plus the Z-coordinates of four selected vertices. This even ensures a consistent visualization where cofacial vertices are actually coplanar, since correct vertex heights are calculated from the four basic ones, rather than be a required input, and are not sensitive to external noise errors. Secondly, such a description is also *minimal* in the sense that, as opposed to fully triangular meshes, the Z-coordinates of only four vertices are required to get the actual shape.

Nevertheless, we need to go beyond these two properties if this representation is to be useful in Computer Graphics applications. Consider what happens if the (x, y) vertex positions in Figure 8.3a are slightly altered, (Figure 8.3b). The new picture no longer represents a *correct projection* of a truncated tetrahedron for, to be so, the edge-lines PR , QS and UT should be concurrent to a common point. Equivalently, note that once P, Q, R and T are given, the height of U is overconstrained, for it can be calculated from *both* the coplanarity of $SQTU$ or that of $RPTU$. For a general projection¹ the two values do not necessarily coincide, and the only spatial lifting that keeps cofacial vertices coplanar is a *trivial* one, with *all* vertices lying on a single plane [80, 79]. This makes the four provided heights inconsistent between each other. In sum, the non-redundancy of the four heights for the picture only holds at very specific positions of the vertices and inevitable discretization errors will make this representation useless. This problem is common in Computer Vision [102] and Computer Graphics [125, 105], and mathematical characterizations of non-redundant configurations are given in [123, 104]. So, our next goal is to make the representation *robust* against these errors.

One way to achieve this follows from a key observation: if the height of a vertex in a picture is overconstrained because the vertex lies on several planes that fix it, we just introduce new triangular faces around it for preventing this to occur (Figure 8.3c). Note that this process actually corrects the incidence structure so that the resulting drawing is generically realizable. As a result, we finally get a robust, non-redundant representation with minimal 3D information on vertex heights.

The rest of the chapter is structured as follows. Section 8.2 presents the trihedral polygonal mesh, the model we use to approximate smooth surfaces. How these meshes are drawn from an input surface is explained in Section 8.3, together with instructions on how to deal with several curvature patterns, and a challenging unsolved configuration left as a point for future research. Following, Section 8.4 introduces T/TT-transformations and gives a fast algorithm that uses them to correct the incidence structure and achieve robust 2D projections. Section 8.5 describes a complementary optimization step that properly puts these transformations to reconstruct the projections with the least possible error. To this end, the problem is reduced to searching the least cost solution of a cyclic AND/OR graph. Points that deserve further attention are discussed in Section 8.6.

¹One where all vertices are in generic position.

8.2 Trihedral Polygonal Meshes

A *trihedral polygonal mesh* is a mesh where all vertices are incident to exactly three faces and, for each face, all its vertices are coplanar. As examples of such meshes consider simple polyhedra as the tetrahedron, the cube, or the dodecahedron, or more complicated ones as the truncated icosahedron, the rhombitruncated cubeoctahedron, or the rhombitruncated icosidodecahedron in Figure 8.4, top. The mesh in Figure 8.4 (center and bottom) was obtained by joining the face barycenters of a triangulation of a teapot. Clearly, its combinatorial structure is that of a trihedral mesh, but it violates the additional constraint on the coplanarity of cofacial vertices and it is not useful for our purposes.

These meshes are specially attractive for our purposes because they yield unequivocal pictures when projected. Indeed, Figure 8.5-top shows that in them, after fixing the planes of two adjacent faces, we have enough data to derive the heights of the remaining vertices. Clearly, the heights of the bold vertices fix the shadowed face planes and the heights of other vertices on them. At this point, any other surrounding face has three vertices whose height is known and, thus, its plane can be fixed too. The same argument can be iteratively applied and the result is a *height propagation* reaching all vertices in the picture.

In the schematic representation of this propagation in Figure 8.5, every face f receives three incoming arrows from the three vertices that fix it. The derivation of heights for the rest of vertices on f is indicated with outgoing arrows from f . The result is a tree-shaped structure spanning all vertices and faces, called hereafter a *height propagation*. In this tree, a path from any of the initial four vertices to any other vertex is an alternating sequence of vertices and faces. Such paths will be referred to as *propagation waves* and will play an important role in the algorithms below. Moreover, the following additional issues need to be considered:

- Trihedral polygonal meshes can be extracted from an arrangement of planes in general position, all tangent to the given surface, for every vertex induced by this arrangement will be incident to exactly three planes. Next section gives the details of such a process.
- If the picture is correct, then varying the heights of the four initial vertices we generate all possible spatial liftings. If it is not, as happened with Figure 8.3d, then the four initial heights will be inconsistent between them, since those vertices receiving more than one propagation wave will be assigned multiple different heights. This problem is overcome in Section 8.4, where we present a simple algorithm to correct the incidence structure, so that no vertex receives more than one propagation wave.
- Any height propagation will succeed to fix all heights provided that no three vertices of any face are collinear, for, if this happens, these three vertices will not uniquely determine a spatial plane. We will get around this problem in Section 8.5, giving an algorithm to find height propagations that fix the faces using sets of three vertices in a less degenerate position as possible.

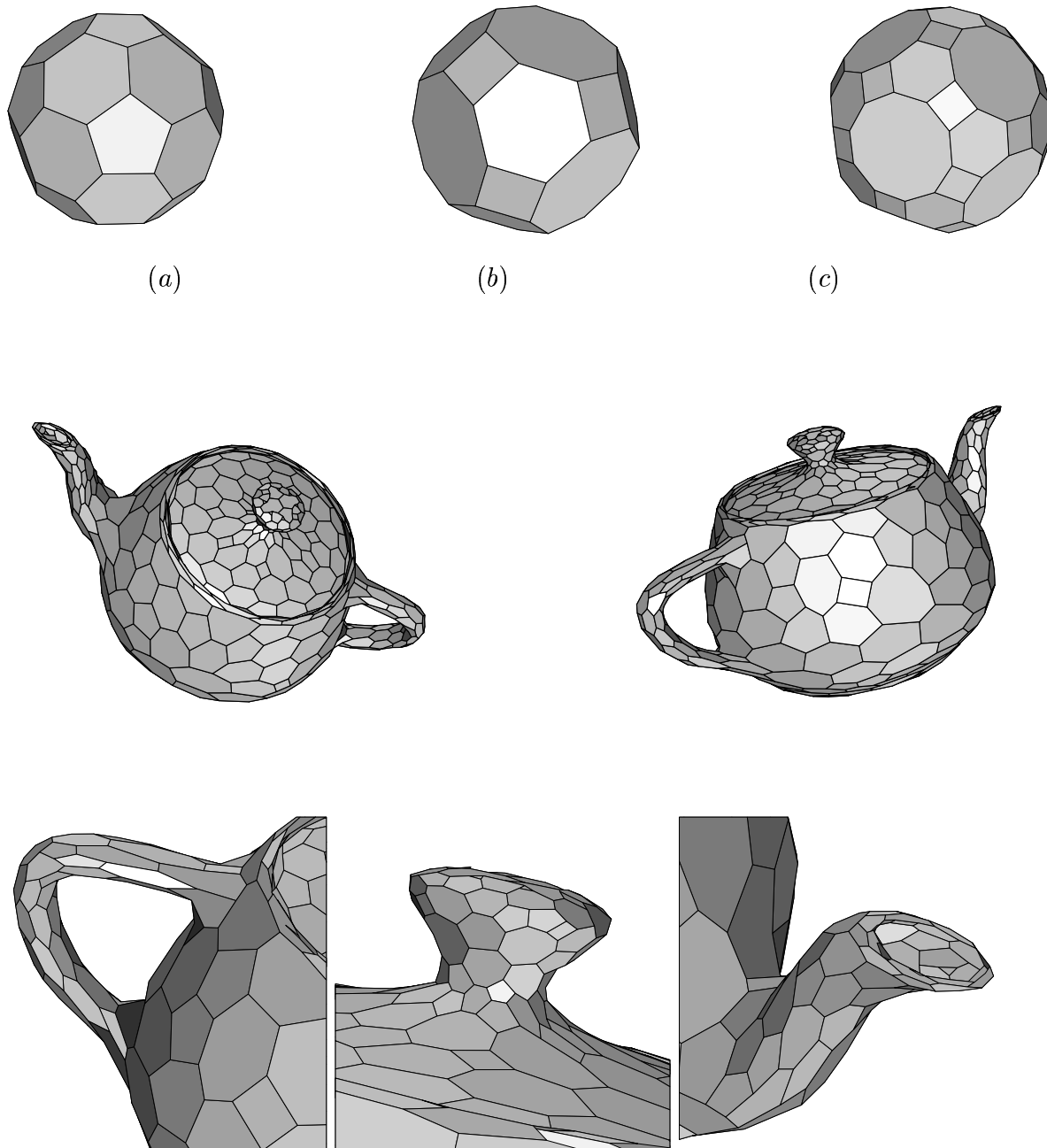


Figure 8.4: Top: trihedral polygonal meshes representing a truncated icosahedron (a), a rhombitruncated cubeoctahedron (b) and a rhombitruncated icosidodecahedron (c). Center and bottom: a mesh with trihedral combinatorial structure.

8.3 Initial Polygonization

Let us assume that we want to obtain a trihedral mesh approximating a convex (or concave) surface. To this end, we can distribute a set of random points all over the surface and compute the tangent planes at these points. This leads to a plane arrangement, clipped by an enclosing box if needed, whose upper envelope —if the surface is convex—

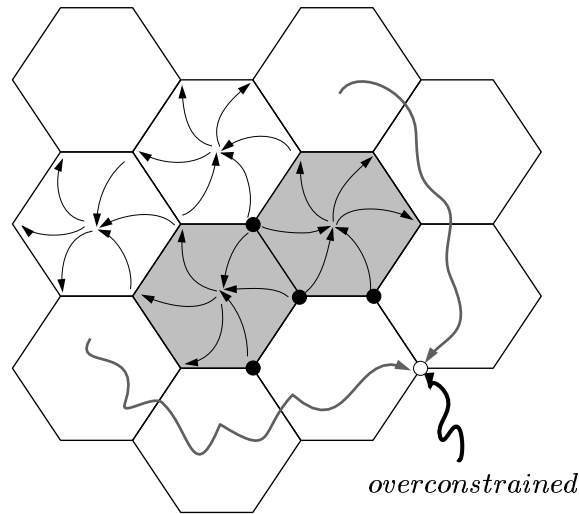


Figure 8.5: A height propagation starting at four pre-specified (bold) vertices. Note that several vertices can have an overconstrained height.

or lower envelope —if it is concave— provides a good mesh approximation of the surface. Since the tangent plane orientations are random, any three of such planes meet in a single point, and hence the mesh is trihedral.

Alternatively, a trihedral mesh approximation of a piece of concave or convex surface can be obtained by starting with a rough polyhedral approximation of it (using for example the above procedure) and iteratively applying a bevel-cutting [30] (and/or a corner-cutting [11]) operation to attain the desired approximation.

Obviously, the situation becomes much more complex when concavities and convexities are simultaneously present. The first step in these cases would be to decompose the surface into patches having congruent signs for the two eigenvalues of the Hessian matrix computed at all their points. If this is done for a general C^∞ surface, we would get patches labelled $(+, +)$, $(+, -)$, $(-, +)$ or $(-, -)$ separated by curves which could be labelled with $(-, 0)$, $(+, 0)$, $(0, -)$, or $(0, +)$, and isolated points (actually, maxima or minima) which would be labelled with $(0, 0)$. Saddle points would be also labelled with $(0, 0)$ but they would appear as intersections of separating curves. If we extend this treatment to C^2 surfaces, we could get entire patches with one of the above nine possible labels. For example, all plane patches would have the label $(0, 0)$.

Patches labelled with $(+, +)$ or $(-, -)$ represent fully convex or concave patches and thus they can be polygonized as described above. Patches labelled as $(*, 0)$ or $(0, *)$ can be polygonized by locating random points along the direction of maximum curvature (i.e., the direction orthogonal to the eigenvector associated with the 0 eigenvalue). Patches $(0, 0)$ would only require a single point on it. Unfortunately, the treatment of $(+, -)$ or $(-, +)$ patches remains as an open problem for us.

The connection between polygonized patches can be obtained by computing tangent planes on points along their common boundaries. In sum, the polygonization we propose

can be done, first for each patch by generating the tangent planes in a sufficiently high density, and next by connecting them using the tangent planes generated along their common boundaries.

8.4 T and TT-Transformations

In a trihedral mesh a projection is overconstrained because any of its vertices lies on three faces and, potentially, up to three propagation waves can determine a height at the same time. However, as done in Figure 8.3c, this can be avoided by adding triangular faces.

To this end, we first compute an arbitrary height propagation spanning all vertices, and check which of them receive more than one wave. We then take one overconstrained vertex v at a time and prevent that all *but one* of the waves reaches v as follows.

To stop the wave getting v from face f , we apply either of these two transformations (Figure 8.6a and b):

- A *T-transformation* places a new edge joining the two neighbouring vertices of v in f , say v_l and v_r .
- A *TT-transformation* places a new vertex v' on f near v and the three new edges (v', v) , (v', v_l) and (v', v_r) .

After either transformation, the constraint “ v must lie on f ” has been dropped from the picture, and f cannot constrain the height of v anymore. Also, the added triangles are innocuous: they do not constrain the space of liftings in any way since every assignment of heights to v , v' , v_l and v_r will generate a plane face for them. Further, note that, after this, all heights can still be determined from the four initial ones and, thus, these transformations keep the picture unequivocal.

Which transformation is preferred depends on the geometry of face f around vertex v . If all points inside the triangle $v_l v_r v$ belong to f , we say that f is *locally convex* at v , otherwise f is said to be *locally non-convex* at v . So, for situations where f is locally convex at v , simplicity prevails and T-transformations are enough. When local non-convexities are present (Figure 8.6c), T-transformations yield occluded or partially occluded crossing edges whose spatial interpretations have overhanged parts, or self-intersecting faces. Here, TT-transformations are preferred for they can avoid this.

An important observation complements the strategy. In an overconstrained vertex v , either two or three incoming propagation waves arrive. If *no more* than one of them comes through a locally non-convex face, then we can always drop the incidence constraint in this vertex just with T-transformations: we just leave the eventual “bad” wave to determine the height of v and stop the others with T-transformations.

This completes the description of a one-sweep simple algorithm removing overdetermination. As an application, Figure 8.7 shows a projected dodecahedron after applying T-transformations.

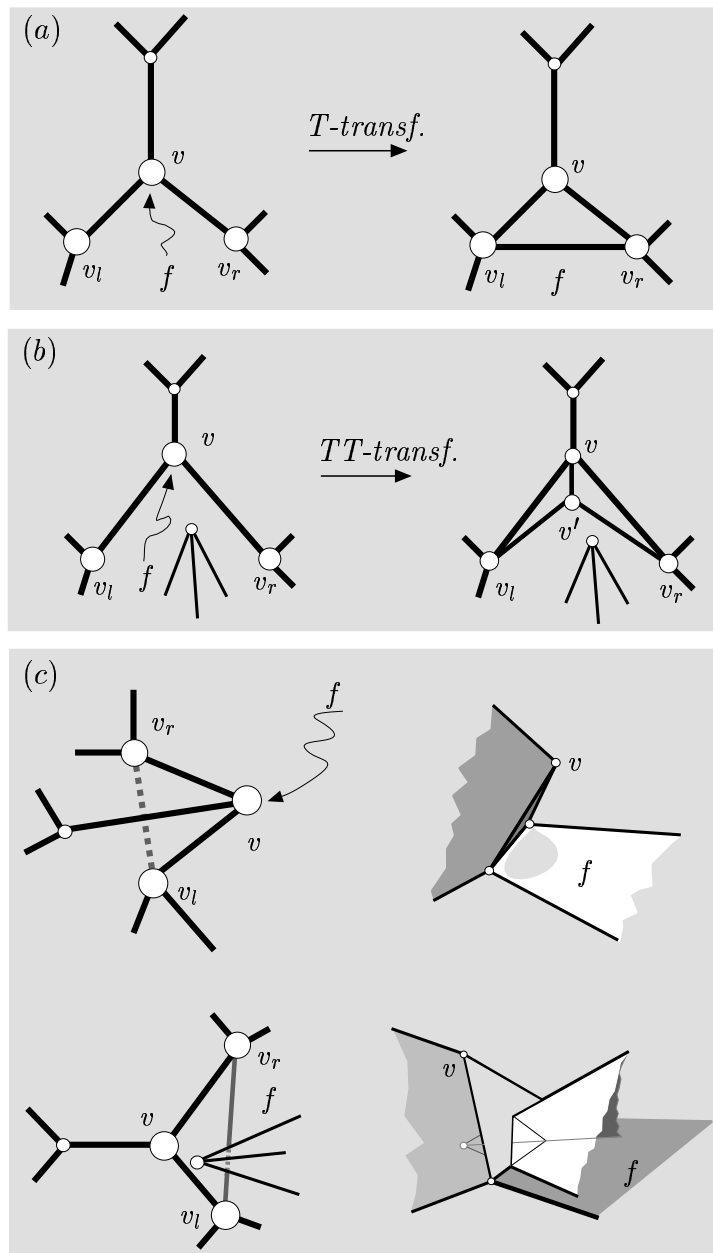


Figure 8.6: (a and b) T and TT-transformations. (c) Overhanged and self-intersecting liftings induced by T-transformations at locally non-convex faces.

In general, when the approximated surface is uniformly convex, or uniformly concave, all faces of the resulting trihedral polygonal mesh will be locally convex, and hence T-transformations will suffice. However, even when local non-convexities exist at the faces, there still might be some height propagations where only T-transformations suffice. In Figure 8.7e, for example, an algorithm computing an arbitrary propagation can be forced to use TT-transformations, whereas with a proper search, a robust projection is obtained only with T-transformations (Figure 8.7f). But one certainly finds correct projections where no propagation strictly using T-transformations can be found (Figure 8.7g). This poses an interesting open problem which we leave for further consideration: to devise an

efficient algorithm to search for the height propagation inducing the least possible number of TT-transformations.

8.5 Optimal Propagations

The algorithm in the preceding section corrects the incidence structure by finding an arbitrary height propagation and inserting a T or a TT-transformation whenever a vertex height is determined by two or more faces. However, arbitrary propagations might travel along “degenerate paths” where the planes for some of the faces are determined by three aligned (or almost aligned) vertices. Clearly, these *degenerate propagations* must be avoided if we want to minimize the errors during the reconstruction of the spatial shape from the initial set of four heights. This section provides an algorithm to find height propagations that avoid travelling along these degenerate configurations as much as possible. The issue is formulated as an optimization problem, namely that of finding the least cost solution of a cyclic AND/OR graph.

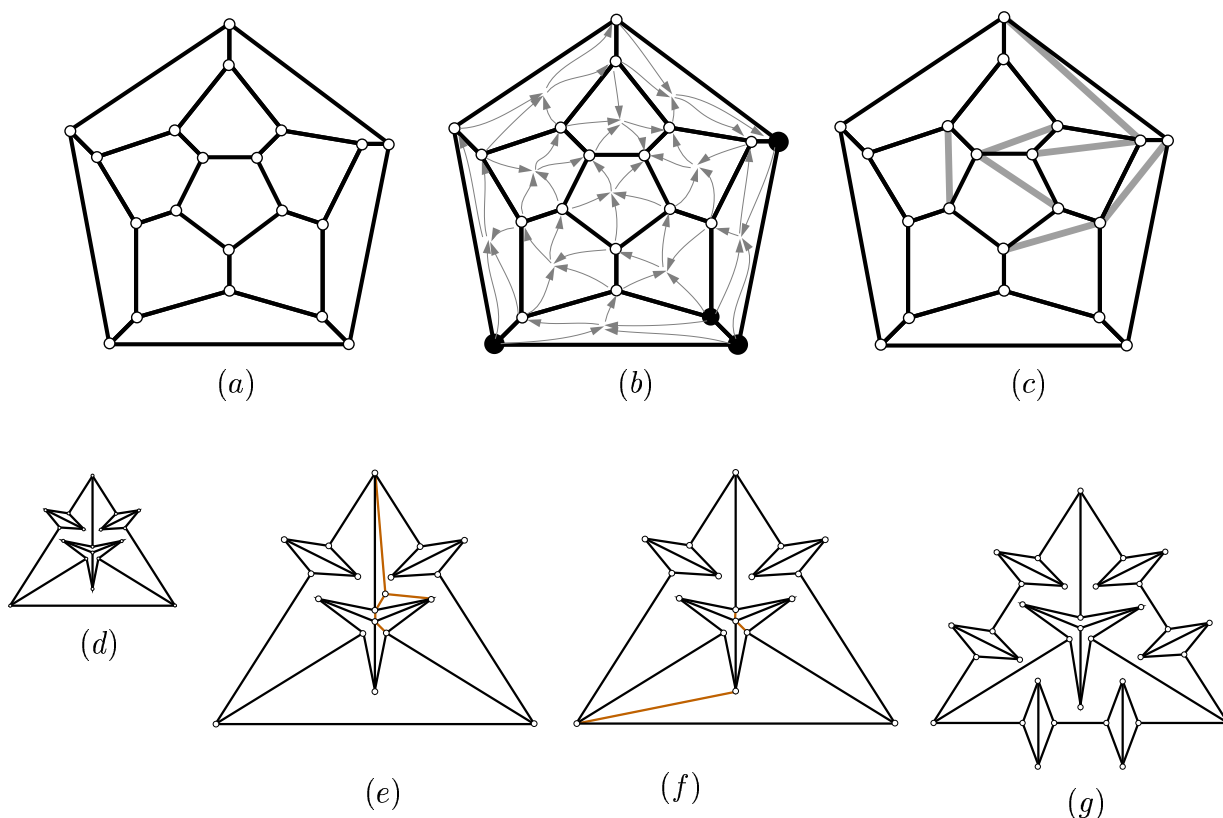


Figure 8.7: A projected dodecahedron (a) together with a height propagation (b) and the result of applying T-transformations (c). A protruded tetrahedron (d) and two possible corrections: (e), involving TT-transformations, and (f), an optimal correction only with T-transformations. There are meshes where propagations involving only T-transformations do not exist (g).

8.5.1 Cyclic AND/OR Graphs

The use of AND/OR graphs for representing problems originated during the sixties within the domain of Artificial Intelligence. Since then, it has spread to other fields such as Operations Research and Robotics, where they are used to represent cutting problems, polyhedral interference tests and assembly disassembly plans. [46, 47] provides a recent survey of available algorithms. We now recall some preliminary concepts about this kind of graphs.

An AND/OR *directed* graph G , can be regarded as a hierarchic representation of possible solution strategies for a major problem, represented as a *root node* r in G . Any other node v represents a subproblem of lower complexity whose solution contributes to solve the major problem at hand.

There are three types of nodes: AND nodes, OR nodes and TERMINAL nodes. Every node v has a set $S(v)$ of *successor nodes*, possibly empty, to which it is connected in either of two ways:

- An AND node v is linked to all nodes $s_i \in S(v)$ through directed AND arcs (v, s_i) , meaning that the subproblem for v can be trivially solved once *all* subproblems for the nodes in $S(v)$ have been solved.
- An OR node v is linked to all nodes $s_i \in S(v)$ through directed OR arcs (v, s_i) , meaning that the subproblem for v can be trivially solved once *any one* of the subproblems for the nodes in $S(v)$ has been solved”.
- A TERMINAL node represents a yet-solved or trivial subproblem and has no successors.

With this setting, a *feasible solution* to the problem becomes represented as a directed subgraph T of G verifying:

- r belongs to T .
- If v is an OR node and belongs to T , then exactly one of its successors in $S(v)$ belongs to T .
- If v is an AND node and belongs to T , then every successor in $S(v)$ belongs to T .
- Every leaf node in T is a TERMINAL node.
- T contains no cycle, it is a tree.

One can also assign a cost $c(u, v) > 0$ to every arc (u, v) in G and ask for the solution T with minimum overall cost

$$C(T) = \sum_{(u,v) \in E(T)} c(u, v),$$

where $E(T)$ is the set of arcs of T . Note that, as defined, G can contain cycles. This turns out to be the main difficulty for this optimization problem, which in the past was usually tackled by a rather inefficient trick: “unfolding” the cycles and applying standard AND/OR search methods for acyclic graphs. However, explicit treatment of cycles has recently been considered, and an algorithm is achieved in [47], whose complexity is $O(n^3)$ in the number of cost computations at the nodes (see also [46, chapter 5]).

The search for an optimal height propagation is next converted to this model. This amounts to (1) constructing an AND/OR graph G_{hp} whose feasible solutions define a height propagation, and (2) define a cost function that promotes non-degenerate propagations over degenerate ones.

8.5.2 Feasible Height Propagations

A height propagation can be defined by the following rules, with the given straightforward translation into AND/OR subgraphs.

R1: *Four selected vertices of the projection trigger the propagation.*

For this, we put a TERMINAL node for each of the triggering vertices.

R2: *Every face in the polygonization can be determined once the heights of any three of its vertices are determined.*

If $\text{deg}(f)$ denotes the number of vertices of face f , then there are $c_f = \binom{\text{deg}(f)}{3}$ possible combinations of three vertices determining f . If we put a node in G_{hp} for every vertex, except for the four triggering ones, then this rule is translated by adding an OR node for every face, linked to c_f new “dummy-face” AND nodes, each representing one of the above combinations. Each dummy-face node is in turn linked with arcs to the three involved vertices in the combination. Figure 8.8 gives a schematic representation. The newly introduced vertex nodes have not been assigned a type yet. This type is induced by the following rule.

R3: *Except for the initial four vertices, the height of every other vertex is determined once one of its incident faces has a determined plane.*

This implements the fact that the propagation wave fixing the height of a vertex can come from any of its three incident faces (Figure 8.8b). The rule can be represented by setting each vertex node as OR type, and linking it to the face nodes of its incident faces (Figure 8.8c). This creates three cycles for each vertex of the polygonal mesh.

R4: *The height propagation must reach all vertices.*

The translation is here very simple. We add a root AND node r to G_{hp} and link it to all vertex nodes.

Note that a feasible solution of G_{hp} provides the actual instructions to draw a height propagation that reaches all vertices, starting at the four pre-specified heights. The solution tree will actually have the same structure as the propagation scheme depicted in

Figure 8.5, the only difference being in the orientation of the arcs which is opposite to the one given there, in accordance with the common criterion for AND/OR graphs.

Note that a feasible solution tree T of G_{hp} provides instructions to derive a height propagation that reaches all vertices, starting at the four pre-specified heights. Actually, T will have nearly the same structure as the propagation scheme depicted in Figure 8.5, with only the following two differences:

- The orientation of its arcs is opposite to the one given there (in accordance with the common criterion for AND/OR graphs).
- Every vertex node in this tree will only receive a single propagation wave, whereas, as depicted in Figure 8.5, the height propagation may overconstrain a vertex through more than one wave. However, the waves not represented in T can be trivially inferred to complete the whole propagation pattern.

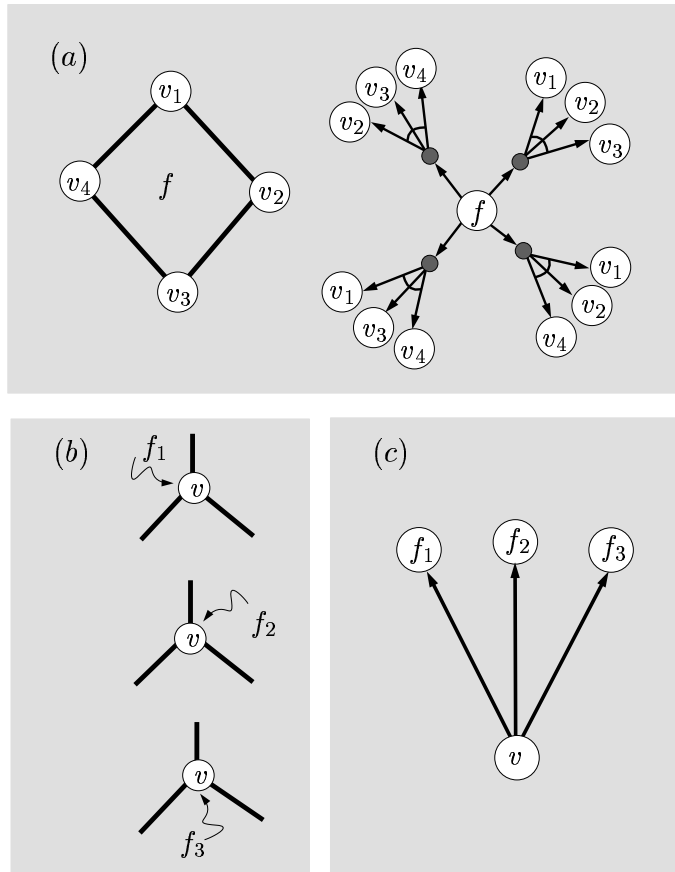


Figure 8.8: AND/OR subgraphs for the propagation rules. AND nodes are indicated by joining all their emanating arcs. (a) Constructed subgraph translating rule **R2** for a quadrilateral face. Dummy-face nodes are shadowed in grey. Note that there is only one vertex node for each vertex in the trihedral mesh, but for clarity they are here duplicated. (b) Propagation waves reaching a vertex. (c) Subgraph for rule **R3**, with an arc for each of the possibilities in (b).

8.5.3 Cost Function

In order to penalize propagations using sets of almost-aligned vertices, we proceed as follows. Consider a height propagation that fixes a face plane f from the point coordinates of three previously fixed vertices v_i , v_j and v_k . We can simply penalize the corresponding arcs in G_{hp} emanating from f by giving them a cost that is inversely proportional to the area of the triangle defined by v_i , v_j and v_k in the projection. In doing this, the least-cost solution of the AND/OR graph will be the least degenerate height propagation. The rest of arc costs are actually irrelevant, but need to be positively defined. In sum, for every directed arc (u, v) we define its cost as follows:

1. $c(u, v) = 1/\det(v_1, v_2, v_3)$, if u is a dummy-face AND node and v is any one of its descendants. Here, v_i , v_j and v_k are the homogeneous coordinates of the vertices associated with the three descendants of u .
2. $c(u, v) = 1$, if u is an OR node.
3. $c(u, v) = 1$, if u is the root AND node.

Once the least cost solution T is found, and its associated height propagation derived, the projection can already be corrected by putting T or TT-transformations as follows: at a vertex v receiving more than one propagation wave, we put one such transformation for all faces fixing v except for the one on the propagation wave represented in T .

8.5.4 Complexity Analysis

As said before the worst-case complexity of computing the optimal solution of a cyclic AND/OR graph with n nodes is $O(n^3)$. We now prove that, in an average case, this bound still holds by seeing that the number of nodes in G_{hp} grows linearly with the number of vertices of the trihedral polygonal mesh.

Let e , v and f be the number of edges, vertices and faces of the given mesh. Then, $2e = 3v$ because the mesh is trihedral. Moreover, if the mesh has h holes, with “the outside” of the mesh counting as a hole too, then Euler’s relation says that $v - e + f = 2 - h$. From these two equalities the number of faces of the mesh can be written in terms of the number of vertices and holes, $f = \frac{v+4}{2} - h$. Let us now count the number of nodes added by each of the rules **R1**, ..., **R4**:

- Rule **R1** adds four vertex nodes.
- Rule **R2** adds one OR node for each face, amounting to $f = \frac{v+4}{2} - h = O(v)$ total nodes, assuming a constant number of holes. Also, for every face f this rule adds $c_f = \binom{\deg(f)}{3}$ dummy-face AND nodes. Although this number is clearly in the worst case $O(\deg(f)^3)$, if we divide the sum of face degrees by the number of faces, the

average face degree is six, at an increasing number of randomly placed vertices in the mesh:

$$\frac{\sum_{\text{all faces}} \text{deg}(f_i)}{f} = \frac{3v}{\frac{v+4}{2} - h} = \frac{6v}{v+4-2h}$$

which will keep the number of dummy-face AND nodes linearly growing:

$$\binom{6}{3} f = 20 \left(\frac{v+4}{2} - h \right) = O(v)$$

- Rule **R3** adds a linear number of OR vertex nodes.
- Rule **R4** only adds one AND node, the root.

Up to now we have assumed that the four vertices triggering the propagation are a priori selected. But other height propagations starting at other four vertices could yield better height propagations. To test all possibilities, we do not need to repeat the AND/OR search for every different combination of four vertices. Indeed, note that these vertices just fix the planes of the faces they belong to. So, any other set of four vertices on these faces will yield the same optimal propagations, provided that two of them lie on the common edge. We can equivalently think of pairs of faces triggering the propagation and use their face nodes as TERMINAL in G_{hp} . The choice of TERMINAL vertices (instead of TERMINAL faces) was done to be coherent with previous explanations. In sum, if one wants to search over all possible starting places of propagation, then for each pair of adjacent faces the AND/OR search needs to be repeated. This amounts to solve $e = \frac{3}{2}v$ optimization problems in the worst case, meaning that the overall complexity will be $O(v^4)$, under the assumption that the face degree is six in average.

8.6 Further Extensions

We have presented a correction strategy that takes a trihedral line drawing as input and uses T and TT-transformations to place triangular faces at strategic places until a generically correct line drawing is obtained. We have also shown where to put these triangles so that the spatial reconstruction of the drawing is performed in the most accurate way as possible, avoiding height propagations along degenerate paths. Moreover, we have shown how to obtain trihedral meshes that approximate a given object's surface. However, although we can deal with an important range of possible surfaces, no algorithm has been devised to obtain trihedral meshes approximating surfaces containing saddle-crests or saddle-valleys —surfaces concave and convex at the same time. This point, as well as the implementation of the whole procedure, will concentrate our future efforts.

To conclude, it is worth mentioning two issues related to the proposed AND/OR graph search that deserve further attention. First, since long propagation paths make the heights more sensitive to an error in the four triggering vertices, under some circumstances it might be better to also minimize the length of the longest propagation wave, while still

avoiding degenerate height propagations. It would be interesting to define a cost function for this combined optimization problem. Note, however, that even when long propagation waves produce large errors in the recovered heights, the unequivocal character of the 2D picture will still permit a reasonable interpretation of the 3D meaning that it is encoding. Finally, it would be interesting to reduce the complexity of the search. The AND/OR graphs we use are almost the same in their structure; the only difference being the nodes set as TERMINAL. Therefore, once we get a solution tree T with minimum cost for some graph, T may help to find the minimum cost tree for another AND/OR graph. There is the possibility that the time complexity can be reduced in this way.

Chapter 9

Conclusions

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

Alan Turing —From his paper on the Turing test “Computing Machinery and Intelligence”, 1950 [107].

During the seventies and early eighties, the spatial interpretation of line drawings was one of the hot topics in Computer Vision and Artificial Intelligence. This is not too surprising, as one of the aims of these disciplines is to furnish a robot with visual capabilities to interact with the environment, which means that some means of automatic scene reconstruction needs to be developed. Line drawings offered a simple idealized picture of what a robot could observe, but precisely this simplicity, together with the fact that humans are also able to infer 3D shape just from boundary information, made the problem interesting and one of coveted solution.

The main issues to attack were not clear initially, but after the works by Guzmann, Huffmann, Clowes and Waltz, the *realizability problem* emerged as a reasonable one to start with: what conditions must be verified on a line drawing so that it represents a true projection of a polyhedral scene? By the end of the seventies, many computer vision scientists who were later to gain international prominence, had already engaged in a race to solve this question. Gradually, partial solutions were found and published, from the labelling techniques to the gradient space approach, authored by Waltz, Mackworth, Kanade, Sugihara and Shapira, to name a few renowned ones. It was finally Sugihara who gave a complete set of conditions, in a series of papers [100, 103, 102, 101] that culminated with the publication of his celebrated book in 1986 [104].

Despite having resolved the main questions, in the course of his investigation Sugihara noticed the connecting threads between line drawing interpretation and the rigidity of plane skeletal structures and began establishing contacts with other geometers working in Rigidity Theory, specially with Walter Whiteley and Henry Crapo of the Canadian Structural Topology Group, who were working on complete proofs and extensions of Maxwell’s theorem by that time. This contact led to a remarkable theorem by Sugihara on the combinatorial characterization of generic realizability (see Theorem 3.1 in Chapter 3). The

proof was partially achieved by him in [99] and finally completed by Whiteley in [123]. Sugihara reports on this proof in his book, but also devotes a full chapter to collect the main connections of Rigidity Theory with line drawing interpretation. In the introduction he writes: “this chapter is rather a digression from the main story of the book, but seems important for future researches on line drawings” [104, page 12]. It was probably this sentence one that mostly influenced us to pursue new enhancements, to clarify the connections with the mechanics of articulated frameworks, and to try to approach the problem in a less algebraic fashion as possible. We then discovered the concurrence conditions for a correct truncated tetrahedron, a 4-calotte and a 5-calotte (Figure 4.1), which motivated our first question in 1997: could it be possible that a drawing’s correctness be checkable solely verifying concurrence conditions on groups of three lines?

We realized that the answer is “yes” with a rather sophisticated proof that used delta-wye transformations to reduce the graph of the depicted polyhedron to one of a tetrahedron [80, 79]. However, in the course of a stay with Whiteley’s group, we realized that he had already discovered a simpler version of this fact several years before, involving a smaller number of concurrence tests [124]. His cross-section test though, was only proven for spherical polyhedra, and we were seeking a more general method, suitable to test drawings of several opaque polyhedra with occlusions among them. This motivated our research on synthetic geometric tools presented in Chapter 4 and in [81]. Our contributions on this regard have been mainly two:

1. An extension of the range of applicability of the cross-section test, which now includes the case of line drawings of polyhedral disks with holes.
2. A simpler proof to the fact that cross-sections are necessary and sufficient to decide the correctness of these types of drawings.

These results can be extended in a number of ways. On the one hand, we note that the cross-section is just one among several known *reciprocal diagrams*. As seen in Chapter 3, other similar diagrams have been found and used for the same purposes, like Maxwell’s reciprocal or the gradient-space approach, for example. The known fact that Maxwell’s reciprocal can be transformed to the cross-section through a plane polarity [124] indicates that all these diagrams are essentially the same, up to projective transformations. It would be helpful to clarify which are these transformations with a unifying aim in mind, since one usually has the impression that many authors use the same concepts under a different language, when browsing the literature [54, 44, 25, 48, 49]. On the other hand, the main drawback of cross-sections is that no general method has been given yet to construct a cross-section or otherwise show that none exists, using a pencil and a straightedge alone. Although we have shown an incremental construction that works for trihedral drawings, a general tool still remains unknown. Its development, or a proof of its non-existence, are challenging open problems for further consideration.

On the contrary, if algebraic methods are used, we have seen that an algorithm to construct cross-sections does exist, which also constitutes a powerful method to decide the correctness of drawings of objects with complicated topologies, with any genus and an arbitrary number of boundaries (see Chapter 6). This algebraic algorithm, though,

relies on a highly kinematic-geometric observation, realized by Whiteley in [121]. Namely, that motions of an associated panel framework are in one-to-one correspondence with the spatial reconstructions of the drawing. Then, telling whether a drawing is correct reduces to the study of the instantaneous motion space of this framework, and it is precisely here where the tools of the Grassmann-Cayley algebra have become useful (Chapter 5). We have further exploited this connection in a number of ways, leading to an approach whose main advantages over Sugihara's are the following:

1. One can decide the correctness of a drawing by simply computing a vector basis of the kernel of a matrix, whose entries are just Plücker coordinates of the edges. This offers a simpler alternative to Sugihara's approach based on Linear Programming.
2. As opposed to Sugihara's, this method is straightforwardly implementable in floating point arithmetic, as it can be made robust to the super-strictness problems induced by numerical round-off errors by simply using the singular value decomposition to find a basis of the aforementioned kernel.
3. A single test is enough to check many differently labelled versions of the same drawing \mathcal{D} . If two labellings assign occlusive labels to the same edges, they yield the same incidence structure for \mathcal{D} , regardless of whether the orientation of the labels (\rightarrow or \leftarrow) coincides on each edge or not. In consequence, a single test is needed to check both labellings simultaneously.
4. $\{+, -\}$ -labellings can be generated in a constructive fashion, rather than by solving a constraint satisfaction problem. The only pre-requisite to this end is that the incidence structure be known or, equivalently, that the input line drawing has its boundary edges identified.

Once the correctness is ensured, the best we can do is to recover all liftings in a parameterized fashion. However, humans do not seem to get such an infinite family of reconstructions, but a single one among them, and a possible extension that we envisage here is to achieve a similar result. This can be done in a number of ways. For example, we can use shading information from the image to select a lifting that best approximates the given brightness pattern (under some model of illumination). This is currently a joint work with Michel Devy from the LAAS lab at Toulouse. The same effect could be achieved with an assumption of which liftings a human seems to prefer. Recent studies by Marill [56, 57, 58] indicate that a human might tend to select those interpretations that minimize the standard deviation of the dihedral angles between pairs of faces. The weaknesses of this study are criticized in [52] by Leclerc and Fischler, arguing that Marill's simple technique may produce liftings with non-coplanar faces even on simple line drawings. Nevertheless, we anticipate that these non-coplanarities can be avoided if the linear parameterization of the interpretation space is used in this minimization.

The last part of the Thesis is devoted to the correction of line drawings and it is mainly the outcome of a research stay with Sugihara. We propose a first strategy that minimally modifies the positions of all vertices until the closest correct drawing is found. This approach presents several advantages over previously existing ones:

1. Since we permit the movement of all vertices and we minimize a sum of Euclidean distances between vertex pairs, we get closer corrected versions of the input drawing, as compared to those one can achieve with the techniques in [104].
2. The correction strategy is itself a realizability test. If the input drawing is correct, this is quickly detected by the method that finds an initial approximation to the minimization problem (Section 7.6).
3. The method of Ponce and Shimshoni [77, 92] is not complete, since it can classify an incorrect drawing as correct. Our approach does not suffer from this drawback.

However, we still foresee some enhancements to our technique. The most relevant of them is to turn the rational parameterization in Section 7.4 into a polynomial one, by using projective rather than Euclidean coordinates for the vertices. We have preliminary studies showing that this avoids the presence of divisions by zero in the objective function, and could even prevent from falling on bad solutions whose liftings have coplanar faces. This polynomial parameterization would open the door to the use of more powerful optimization techniques, in an attempt to attain the global minimum. Interval arithmetic [39] or Bézier-clipping methods could be very suitable to this end [10].

Finally, in Chapter 8, we give a second alternative to correct line drawings, this time by keeping the vertices fixed but modifying the incidence structure. This has been presented within an application to robust solid modelling, for which several enhancements are under consideration. Among these, we are pursuing an algorithm to obtain trihedral meshes approximating surfaces containing saddle-crests or saddle-valleys and a full implementation of the whole procedure.

Despite the importance of line drawing interpretation in the past decades, nowadays we observe a gradual decrease of researchers' interest on these topics. This decay is probably motivated by two reasons: on the one hand, the main theoretical questions were already solved by the middle eighties, with the appearance of Sugihara's papers and his book in 1986 [104]; on the other hand, the problem itself was too idealistic to be applied to practical situations, since line data extracted from real images are far from perfect line drawings, and it seems difficult to use the line drawing interpretation theory directly for real image processing. Moreover, Robotics itself is experiencing substantial changes in the way of approaching its fundamental questions. Most significantly, classical solutions to planning a robot's actions, which used to rely on highly symbolic reasoning over simplified models of reality, turn out to be infeasible on complex unstructured environments and, in consequence, they are being gradually substituted by behaviour-based approaches that avoid internal representations of the world as much as possible. As a result, reconstruction methods like the ones we present here are currently disregarded, due to their highly symbolic nature, and the strong assumptions they introduce about the external world—like polyhedrality in our case.

Nevertheless, driven by the pervasiveness of diagrams in human communication and by the increasing availability of graphical environments in computerized work, the study of diagrammatic notations is emerging as a research field in its own right. This development has simultaneously taken place in several scientific disciplines, including Cognitive Science,

Artificial Intelligence and Computer Science among others, which indicates a return to the main questions of drawing interpretation inside a broader context: drawings are not only regarded as a means of communication with computers but they are also recognized as a key ingredient of visual intelligence.

Bibliography

- [1] AGULLÓ, J. *Mecànica del Sòlid Rígid*. Publicacions OK Punt, 1995.
- [2] ALEVIZOS, P. A linear time algorithm for labelling planar projections of polyhedra. In *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems* (Osaka, Japan, 1991), pp. 595–601.
- [3] BALL, R. S. *A Treatise on the Theory of Screws*. Cambridge University Press, 1900. Reprinted by Cambridge University Press in 1998.
- [4] BALLARD, B. H., AND BROWN, C. M. *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [5] BARROW, H. G., AND TENENBAUM, J. M. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, A. Hanson and E. M. Riseman, Eds. Academic Press, New York, 1978.
- [6] BARROW, H. G., AND TENENBAUM, J. M. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence* 17 (1981), 75–116.
- [7] BARROW, H. G., AND TENENBAUM, J. M. Retrospective on “interpreting line drawings as three-dimensional surfaces”. *Artificial Intelligence* 59 (1993), 71–80.
- [8] BIGGS, N. *Algebraic Graph Theory*. Cambridge Tracts in Mathematics. Cambridge University Press, 1974.
- [9] BOLLOBÁS, B. *Graph Theory*. Springer Verlag, 1979.
- [10] BOMBÍN, C., ROS, L., AND THOMAS, F. A concise Bézier clipping technique for solving inverse kinematics problems. In *Advances in Robot Kinematics*, J. Lenarcic and M. M. Stanisic, Eds. Kluwer Academic Publishers, 2000, pp. 53–60.
- [11] BOOR, C. Cutting corners always works. *Computer Aided Geometric Design* 4 (1987), 125–131.
- [12] BOTTEMA, O., AND ROTH, B. *Theoretical Kinematics*. Dover Publications, Inc., 1990. This Dover edition is an unabridged and corrected republication of the work originally published in 1979 by North-Holland Publishing Company, Amsterdam, as Volume 24 of the North-Holland Series in Applied Mathematics and Mechanics.

- [13] BRISSON, E. Representing geometric structures in d -dimensions: topology and order. *Discrete Computational Geometry 9* (1993), 387–426.
- [14] BURKE, J. *The Pinball Effect. How Renaissance Water Gardens Made the Carburetor Possible*. Little, Brown and Company, 1996.
- [15] CASTELLET, M., AND LLERENA, I. *Àlgebra Lineal i Geometria*. Publicacions de la Universitat Autònoma de Barcelona, 1990.
- [16] CLOWES, M. B. On seeing things. *Artificial Intelligence 2*, 1 (1971), 79–116.
- [17] COOPER, M. C. Linear-time algorithms for testing the realisability of line drawings of curved objects. *Artificial Intelligence 108* (1999), 31–67.
- [18] CRAPO, H. Invariant-theoretic methods in scene analysis and structural mechanics. *Journal of Symbolic Computation 11* (1991), 523–548.
- [19] CRAPO, H. Structural topology, or the fine art of rediscovery. *The Mathematical Intelligencer 19*, 4 (1997), 27–35.
- [20] CRAPO, H., AND WHITELEY, W. Statics of frameworks and motions of panel structures, a projective geometric introduction. *Structural Topology 6* (1982), 42–82.
- [21] CRAPO, H., AND WHITELEY, W. Plane self stresses and projected polyhedra I: The basic pattern. *Structural Topology 20* (1993), 55–78.
- [22] CREMONA, L. *Graphical Statics: Two Treatises on the Graphical Calculus, and Reciprocal Figures in Graphical Statics*. Oxford University Press, 1890. English Translation by T. H. Beare of “Le figure reciproche nelle statica grafica”, 1872.
- [23] DOUBILET, P., ROTA, G.-C., AND STEIN, J. On the foundations of combinatorial theory: IX. Combinatorial methods in invariant theory. *Studies in Mathematics 53*, 3 (Sept. 1974).
- [24] DRAPER, S. W. The Penrose triangle and a family of related figures. *Perception 7* (1978), 283–296.
- [25] DRAPER, S. W. The use of gradient and dual space in line-drawing interpretation. *Artificial Intelligence 17* (1981), 461–508.
- [26] EDELSBRUNNER, H. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Heidelberg, 1987.
- [27] EINSTEIN, A., AND INFELD, L. *The Evolution of Physics. From Early Concepts to Relativity and Quanta*. Simon and Schuster, 1967.
- [28] FORDER, H. *The Calculus of Extension*. Chelsea, New York, 1960.
- [29] FORSYTHE, G. E., MALCOLM, M. A., AND MOLER, C. B. *Computer Methods for Mathematical Computations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1977.

- [30] FOX, D., AND JOY, K. I. On polyhedral approximations to a sphere. In *IEEE Int. Conf. on Visualization* (1998), pp. 426–432.
- [31] GRAVER, J., SERVATIUS, B., AND SERVATIUS, H. *Combinatorial Rigidity*, vol. 2 of *Graduate Studies in Mathematics*. American Mathematical Society, 1993.
- [32] GREENBERG, M. J., AND HARPER, J. R. *Algebraic Topology, A First Course*. Addison-Wesley, 1981.
- [33] GRIMSTEAD, I. J., AND MARTIN, R. R. Creating solid models from single 2D sketches. In *Proceedings of the Third Symposium on Solid Modelling and Applications* (1995), C. Hoffman and J. Rossignac, Eds., ACM Press, pp. 323–337.
- [34] GRIMSTEAD, I. J., AND MARTIN, R. R. Incremental line labelling for sketch input of solid models. *Computer Graphics Forum* 15, 2 (1996), 155–166.
- [35] GUIBAS, L. J., AND STOLFI, J. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics* 4 (1985), 74–123.
- [36] GUZMAN, A. Computer recognition of three-dimensional objects in a visual scene. Tech. Rep. MAC-TR-59, Massachusetts Institute of Technology, 1968.
- [37] GUZMAN, A. *Computer Recognition of Three-Dimensional Objects in a Visual Scene*. PhD thesis, Massachusetts Institute of Technology, 1968.
- [38] HALPERIN, D. Arrangements. In *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke, Eds. CRC Press, 1997, pp. 389–412.
- [39] HANSEN, E. *Global Optimization Using Interval Analysis*. Pure and Applied Mathematics. Marcel Decker, Inc., New York, 1992.
- [40] HECKBERT, P., AND GARLAND, M. Survey of polygonal surface simplification algorithms. In *Multiresolution Surface Modeling Course SIGGRAPH'97* (May 1997). Available at <http://www.cs.cmu.edu/~ph>.
- [41] HODGE, W. V. D., AND PEDOE, D. *Methods of Algebraic Geometry*, vol. I. Cambridge University Press, 1968. Reedited by Cambridge Medical Library in 1994.
- [42] HORN, B. K. P. Understanding image intensities. *Artificial Intelligence* 8, 2 (1977), 201–231.
- [43] HUFFMAN, D. A. Impossible objects as nonsense sentences. *Machine Intelligence* 6 (1971), 295–323.
- [44] HUFFMAN, D. A. A duality concept for the analysis of polyhedral scenes. *Machine Intelligence* 8 (1977).
- [45] HUNT, K. H. *The Kinematic Geometry of Mechanisms*. Oxford University Press, 1978. Reprinted with corrections in 1990.

- [46] JIMENEZ, P. *Static and Dynamic Interference Detection Between Non Convex Polyhedra*. PhD thesis, Institut de Robòtica i Informàtica Industrial - Polytechnic University of Catalonia, Gran Capità, 2-4. 08034 Barcelona, Spain, June 1998. Available at <http://www-iri.upc.es/people/jimenez/>.
- [47] JIMENEZ, P., AND TORRAS, C. An efficient algorithm for searching implicit AND/OR graphs with cycles. *Artificial Intelligence* (Accepted for publication).
- [48] KANADE, T. A theory of origami world. *Artificial Intelligence* 13 (1980), 279–311.
- [49] KANADE, T. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence* 17 (1981), 409–460.
- [50] KIROUSIS, L. M., AND PAPADIMITRIOU, C. H. The complexity of recognizing polyhedral scenes. *Journal of Computer and System Sciences* 37 (1988), 14–38.
- [51] KLEIN, F. *Elementary Mathematics from An Advanced Standpoint*. Dover Publications, 1939.
- [52] LECLERC, I. G., AND FISCHLER, M. A. An optimization-based approach to the interpretation of single line-drawings as 3D wire-frames. *International Journal of Computer Vision* 9 (1992), 113–136.
- [53] LIPSON, H., AND SHPITALNI, M. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design* 28, 8 (August 1996), 651–663.
- [54] MACKWORTH, A. K. Interpreting pictures of polyhedral scenes. *Artificial Intelligence* 4 (1973), 121–137.
- [55] MACKWORTH, A. K. Consistency in networks of relations. *Artificial Intelligence* 8 (1977), 99–118.
- [56] MARILL, T. Emulating the human interpretation of line-drawings as three-dimensional objects. *International Journal of Computer Vision* 6, 2 (June 1991), 147–161.
- [57] MARILL, T. Why do we see three-dimensional objects? A.I. Memo 1366, Massachusetts Institute of Technology, June 1992.
- [58] MARILL, T. Recovery of three-dimensional objects from single perspective images. A.I. Memo 1630, Massachusetts Institute of Technology, March 1998.
- [59] MARKOWSKY, G., AND WESLEY, M. A. Fleshing out wire frames: reconstruction of objects, part I. *IBM Journal of Research and Development* 24, 5 (September 1980), 229–258.
- [60] MARR, D. Representing visual information. In *Computer Vision Systems*, A. Hanson and E. M. Riseman, Eds. Academic Press, New York, 1978.

- [61] MARTÍ, E., REGINCÓS, J., LÓPEZ-KRAHE, J., AND VILLANUEVA, J. J. Hand line drawing interpretation as three-dimensional objects. *Signal Processing* 32, 1-2 (May 1993), 91–110.
- [62] MARTÍ, E., REGINCÓS, J., LÓPEZ-KRAHE, J., AND VILLANUEVA, J. J. Line drawing interpretation as polyedral objects to man-machine interaction in CAD systems. In *Pattern Recognition and Image Analysis: V Spanish Symposium of Pattern Recognition and Image Analysis* (1994), World Scientific Publishers, pp. 158–169.
- [63] MAXWELL, J. C. On reciprocal figures and diagrams of forces. *Philosophical Magazine*, Ser. 4, Vol. 27, pp. 250-261, 1864.
- [64] MAXWELL, J. C. On reciprocal figures, frames, and diagrams of forces. *Transactions of the Royal Society. Edinburgh* 26 (1869-72), 1–40.
- [65] NALWA, V. S. Line-drawing interpretation: a mathematical framework. *International Journal of Computer Vision*, 2 (September 1988), 103–124.
- [66] NALWA, V. S. Line-drawing interpretation: Straight lines and conic sections. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 10, 4 (July 1988), 514–529.
- [67] O’CONNOR, J. J., AND ROBERTSON, E. F. The MacTutor History of Mathematics Archive. Mathematical Institute. St Andrews Fife. KY16 9SS, Scotland. <http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Weyl.html>.
- [68] PARODI, P. The complexity of understanding line-drawings of origami scenes. *International Journal of Computer Vision* 18, 2 (May 1996), 139–170.
- [69] PARODI, P., LANCEWICKI, R., VIJH, A., AND TSOTSOS, J. K. Empirically-derived estimates of the complexity of labelling line drawings of polyhedral scenes. *Artificial Intelligence* 105, 1-2 (1998), 47–75.
- [70] PARODI, P., AND PICCIOLI, G. 3D shape reconstruction by using vanishing points. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 18, 2 (February 1996), 211–217.
- [71] PARODI, P., AND TORRE, V. A linear complexity procedure for labelling line drawings of polyhedral scenes using vanishing points. In *ICCV93* (1993), pp. 291–295.
- [72] PARODI, P., AND TORRE, V. On the complexity of labelling perspective projections of polyhedral scenes. *Artificial Intelligence* 70, 1-2 (October 1994), 239–276.
- [73] PENNE, R. Isostatic bar and joint frameworks in the plane with irreducible pure conditions. *Discrete Applied Mathematics* 55 (1994), 37–57.
- [74] PENROSE, L. S., AND PENROSE, R. Impossible objects: a special type of visual illusion. *British Journal of Psychology* 49 (1958), 31–33.
- [75] PENROSE, R. *The Emperor’s New Mind: Concerning Computers, Minds, and the Laws of Physics*. Oxford University Press, 1990.

- [76] PÓLYA, G. *How to Solve It; A New Aspect of Mathematical Method*. Princeton University Press, 1988. Reissue edition. First edition of 1945.
- [77] PONCE, J., AND SHIMSHONI, I. An algebraic approach to line-drawing analysis in the presence of uncertainty. In *IEEE Int. Conf. on Robotics and Automation* (May 1992), pp. 1786–1791.
- [78] ROS, L., SUGIHARA, K., AND THOMAS, F. A new approach to smooth surface polygonization with applications to 3D robust modelling. To appear. Available as a preprint by request to llros@iri.upc.es.
- [79] ROS, L., AND THOMAS, F. Analysing spatial realizability of line drawings through edge-concurrence tests. In *IEEE Int. Conf. on Robotics and Automation* (Leuven, Belgium, May 1998), vol. IV, IEEE Computer Society Press, pp. 3559–3566.
- [80] ROS, L., AND THOMAS, F. Applying delta/star reductions for checking the spatial realizability of line drawings. In *14th European Workshop on Computational Geometry* (Barcelona, Spain, March 1998), F. Hurtado and J. Trias, Eds., pp. 127–131.
- [81] ROS, L., AND THOMAS, F. Shape-from-image via cross-sections. In *15th International Conference on Pattern Recognition* (Barcelona, 3-8 September 2000).
- [82] ROS, L., AND THOMAS, F. Correcting polyhedral projections for scene reconstruction. In *IEEE Int. Conf. on Robotics and Automation* (Seoul, Korea, May 2001), IEEE Computer Society Press. In press.
- [83] SANKER, P. V. A vertex coding scheme for interpreting ambiguous trihedral solids. *Computer Graphics and Image Processing* 6 (1977), 61–89.
- [84] SCHLIK, T., AND FOGELSON, A. TNPACK-A truncated Newton minimization package for large-scale problems: I. Algorithm and usage. *ACM Transactions on Mathematical Software* 18, 1 (March 1992), 46–70.
- [85] SCHLIK, T., AND FOGELSON, A. TNPACK-A truncated Newton minimization package for large-scale problems: II. Implementation examples. *ACM Transactions on Mathematical Software* 18, 1 (March 1992), 71–111.
- [86] SEIBOLD, W., AND WYVILL, G. Towards an understanding of surfaces through polygonization. In *IEEE Int. Conf. on Visualization* (1998), pp. 416–425.
- [87] SHAPIRA, R. A note on Sugihara’s claim. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 6, 1 (January 1984).
- [88] SHAPIRA, R. The use of objects’ faces in interpreting line drawings. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 6, 6 (November 1984), 789–794.
- [89] SHAPIRA, R. More about polyhedra. Interpretation through constructions in the image plane. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 7, 1 (January 1985), 1–16.

- [90] SHAPIRA, R., AND FREEMAN, H. A cyclic-order property of bodies with three-face vertices. *IEEE Trans. Computers C-26*, 10 (October 1977), 1035–1039.
- [91] SHAPIRA, R., AND FREEMAN, H. The cyclic-order property as an aid in scene analysis. *Commun. ACM* 22, 6 (June 1979), 369–375.
- [92] SHIMSHONI, I. *Interpreting Images of Polyhedral Objects in the Presence of Uncertainty*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.
- [93] SHPITALNI, M., AND LIPSON, H. Identification of faces in a 2D line drawing projection of a wireframe object. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 18, 10 (October 1996), 1000–1012.
- [94] SHUFELT, J. A. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 21, 3 (March 1999), 282–288.
- [95] STAFFETTI, E., ROS, L., AND THOMAS, F. Finding infinitesimal motions of objects in assemblies using Grassmann-Caley algebra. In *Tenth World Congress on the Theory of Machine and Mechanisms* (Oulu, Finland, June 1999), vol. 2, Oulu University Press, pp. 584–592.
- [96] STAFFETTI, E., ROS, L., AND THOMAS, F. A simple characterization of the infinitesimal motions separating general polyhedra in contact. In *IEEE Int. Conf. on Robotics and Automation* (Detroit, USA, May 1999), vol. 2, IEEE Computer Society Press, pp. 571–578.
- [97] STURMFELDS, E., AND WHITELEY, W. On the synthetic factorization of projectively invariant polynomials. *Journal of Symbolic Computation* 11 (1991), 439–453.
- [98] SUGIHARA, K. Picture language for skeletal polyhedra. *Computer Graphics and Image Processing* 8 (1978), 382–405.
- [99] SUGIHARA, K. Mathematical structures of line drawings of polyhedrons. Towards man-machine communication by means of line drawings. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 4 (September 1982), 458–469.
- [100] SUGIHARA, K. A unifying approach to descriptive geometry and mechanisms. *Discrete Applied Mathematics* 5 (1983), 313–328.
- [101] SUGIHARA, K. An algebraic and combinatorial approach to the analysis of line drawings of polyhedra. *Discrete Applied Mathematics* 9 (1984), 77–104.
- [102] SUGIHARA, K. An algebraic approach to shape-from-image problems. *Artificial Intelligence* 23 (1984), 59–95.
- [103] SUGIHARA, K. A necessary and sufficient condition for a picture to represent a polyhedral scene. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 6 (September 1984), 578–586.
- [104] SUGIHARA, K. *Machine Interpretation of Line Drawings*. The MIT Press, 1986.

- [105] SUGIHARA, K. Resolvable representation of polyhedra. *Discrete and Computational Geometry* 21, 2 (1999), 243–255.
- [106] TESSARI, D. *La Teoria delle Ombre e del Chiaro-scuro*. Turin, 1880.
- [107] TURING, A. M. Computing machinery and intelligence. *Mind* 59, 236 (October 1950). Reprinted under the title *Can a Machine Think?*, in *The World of Mathematics*, ed. James R. Newman, pp 2099-2123, Simon and Schuster (1956); and in *The Philosophy of Artificial Intelligence*, ed. Margaret Boden, Oxford University Press (1990). Available at <http://www.abelard.org/turpap/turpap.htm>.
- [108] ULLMAN, S. *High-level Vision. Object Recognition and Visual Cognition*. The MIT Press, 1996.
- [109] WALTZ, D. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, P. H. Winston, Ed. Mc. Graw Hill, 1975, pp. 19–91.
- [110] WESLEY, M. A., AND MARKOWSKY, G. Fleshing out wire frames: reconstruction of objects, part II. *IBM Journal of Research and Development* 25, 6 (November 1981), 259–295.
- [111] WESLEY, M. A., AND MARKOWSKY, G. Generation of solid models from two-dimensional and three-dimensional data. In *Solid Modeling by Computers*, M. S. Pickett and J. W. Boyse, Eds. Plenum Publishing Corporation, 1984, pp. 23–51.
- [112] WHITE, N. L. Multilinear Cayley factorization. In *Symbolic Computations in Geometry*. IMA Preprint Series 389. Institute for Mathematics and Its Applications. University of Minnesota, Jan. 1988.
- [113] WHITE, N. L. Multilinear Cayley factorization. *Journal of Symbolic Computation* 11 (1991), 421–438.
- [114] WHITE, N. L. Grassmann-Cayley algebra and robotics. *Journal of Intelligent and Robotic Systems* 11 (1994), 91–107.
- [115] WHITE, N. L. A tutorial on Grassmann-Cayley algebra. In *Invariant Methods in Discrete and Computational Geometry*, N. White, Ed. Kluwer Academic Publisher, 1995, pp. 93–106.
- [116] WHITE, N. L. Geometric applications of the Grassmann-Cayley algebra. In *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.
- [117] WHITE, N. L., AND WHITELEY, W. The algebraic geometry of stresses in frameworks. *SIAM Journal of Algebraic and Discrete Methods* 4, 4 (December 1983), 481–511.
- [118] WHITE, N. L., AND WHITELEY, W. The algebraic geometry of motions of bar-and-body frameworks. *SIAM Journal of Algebraic and Discrete Methods* 8, 1 (January 1987).

- [119] WHITELEY, W. Introduction to Structural Geometry I: Infinitesimal motions and infinitesimal rigidity. Preprint. Champlain Regional College. 900 Riverside Drive, St. Lambert, Quebec, April 1977.
- [120] WHITELEY, W. Introduction to Structural Geometry II: Statics and stresses. Preprint. Champlain Regional College. 900 Riverside Drive, St. Lambert, Quebec, February 1978.
- [121] WHITELEY, W. Motions and stresses of projected polyhedra. *Structural Topology* 7 (1982), 13–38.
- [122] WHITELEY, W. From a line drawing to a polyhedron. *Journal of Mathematical Psychology* 31 (1987), 441–448.
- [123] WHITELEY, W. Some matroids on hypergraphs with applications to scene analysis and geometry. *Discrete and Computational Geometry* 4 (1988), 75–95. Previously appeared in 1984 as: Preprint, Champlain Regional College (900 Riverside Drive, St. Lambert, Quebec, J4P 3B8, Canada).
- [124] WHITELEY, W. Weavings, sections and projections of spherical polyhedra. *Discrete Applied Mathematics* 32 (1991), 275–294.
- [125] WHITELEY, W. How to design or describe a polyhedron. *Journal of Intelligent and Robotic Systems* 11 (1994), 135–160.
- [126] WHITELEY, W. The decline and rise of geometry in 20th century North America. In *Proceedings of the Canadian Mathematics Education Study Group Conference* (1999). Available at <http://www.math.yorku.ca/Who/Faculty/Whiteley/menu.html>.
- [127] WILKINSON, J. H., AND REINSCH, C. *Handbook for automatic computation*, vol. II, Linear Algebra. Springer Verlag, 1971.

Author Index

- Agulló, J. 61
Alevizos, P. 17
- Ball, R. S. 27
Ballard, B. H. 16
Barrow, H. G. 1, 2, 3
Biggs, N. 77
Bollobás, B. 77
Bombín, C. 104, 126
Boor, C. 112
Bottema, O. 61
Brisson, E. 89
Brown, C. M. 16
Burke, J. 15
- Castellet, M. 59
Clowes, M. B. 14, 16
Cooper, M. C. 16
Crapo, H. 27, 28, 30, 42, 53, 60, 69
Cremona, L. 27
- Doubilet, P. 27, 53, 58
Draper, S. W. 3, 6, 28, 124
- Edelsbrunner, H. 89, 90
Einstein, A. 1
- Fischler, M. A. 18, 125
Fogelson, A. 101
Forder, H. 66
Forsythe, G. E. 81
Fox, D. 112
Freeman, H. 18
- Garland, M. 106
Graver, J. 71
Greenberg, M. J. 55
Grimstead, I. J. 93
Guibas, L. J. 89
Guzman, A. 15
- Halperin, D. 88
Hansen, E. 104, 126
- Harper, J. R. 55
Heckbert, P. 106
Hodge, W. V. D. 58, 59
Horn, B. K. P. 1
Huffman, D. A. 3, 6, 14, 16, 28, 124
Hunt, K. H. 61
- Infeld, L. 1
- Jimenez, P. 116, 117
Joy, K. I. 112
- Kanade, T. 16, 124
Kirosis, L. M. 17
Klein, F. 40, 66
- Lancewicki, R. 18
Leclerc, I. G. 18, 125
Lipson, H. 12, 93
Llerena, I. 59
López-Krahe, J. 12, 93
- Mackworth, A. K. 6, 16, 28, 124
Malcolm, M. A. 81
Marill, T. 18, 125
Markowsky, G. 93
Marr, D. 1
Martí, E. 12, 93
Martin, R. R. 93
Maxwell, J. C. 25, 27
Moler, C. B. 81
- Nalwa, V. S. 21
- O'Connor, J. J. 9
- Papadimitriou, C. H. 17
Parodi, P. 14, 17, 18
Pedoe, D. 58, 59
Penne, R. 30
Penrose, L. S. 3
Penrose, R. 3, 35
Piccioli, G. 17
Pólya, G. 93

- Ponce, J. 25, 126
- Regincós, J. 12, 93
- Reinsch, C. 81
- Robertson, E. F. 9
- Ros, L. 30, 36, 53, 61, 104, 109, 124, 126
- Rota, G.-C. 27, 53, 58
- Roth, B. 61
- Sanker, P. V. 16
- Schlik, T. 101
- Seibold, W. 106
- Servatius, B. 71
- Servatius, H. 71
- Shapira, R. 18, 20
- Shimshoni, I. 25, 101, 126
- Shpitalni, M. 12, 93
- Shufelt, J. A. 14, 17
- Staffetti, E. 53, 61
- Stein, J. 27, 53, 58
- Stolfi, J. 89
- Sturmfelds, E. 31
- Sugihara, K. 12, 14, 16, 17, 19, 20, 22, 30, 36, 71, 93, 98, 101, 109, 123, 124, 126
- Tenenbaum, J. M. 1, 2, 3
- Tessari, D. 27
- Thomas, F. 30, 36, 53, 61, 104, 109, 124, 126
- Torras, C. 116, 117
- Torre, V. 14, 17
- Tsotsos, J. K. 18
- Turing, A. M. 123
- Ullman, S. 2
- Vijh, A. 18
- Villanueva, J. J. 12, 93
- Waltz, D. 14, 16
- Wesley, M. A. 93
- White, N. L. 30, 31, 53, 57
- Whiteley, W. 22, 24, 27, 28, 29, 30, 31, 35, 36, 42, 53, 68, 69, 71, 97, 109, 124, 125
- Wilkinson, J. H. 81
- Wyvill, G. 106

Index

- $\Lambda(\mathbb{V})$, 58, 59
- $\mathbb{V}^{(k)}$, 59
- k -extensor, 58
 - step of \mathbf{a} , 58
 - support of \mathbf{a} , 58
- 2-extensor, 28, 60, 63, 64, 72, 73, 75, 78
- 2-manifold, 11
- 4-calotte, 36

- Algebraic Geometry, 30
- AND/OR graphs, 118–119
- Artificial Intelligence, 1, 118, 125

- Cayley factorization, 31
- composition of motions, 63, 67
- Computational Geometry, 86
- conjugate gradient minimization, 100
- consistent labelling, *see* labelling
- Constraint Programming, 16
- coordinate
 - Plücker, *see* Plücker coordinate
- coordinates
 - cartesian, 54
 - homogeneous, 54
- correction
 - of drawings
 - by modifying all vertex positions, 93–106
 - by modifying the incidence structure, 107–123
 - problem, 4
- Cremona, Luigi, 27
- cross-section, 29
 - algebraic construction of \mathbf{a} , 73–76
 - compatible, 29, 37
 - graphical construction of \mathbf{a} , 43–48
 - test, 29–30, 36–40

- dual diagram, 29

- edge
 - boundary, 11
 - concave, 13
 - convex, 13
 - interior, 11
 - occlusive, 14

- face
 - adjacent, 10, 11
 - of a polyhedral surface, 11
 - of an incidence structure (or line drawing), 9
 - plane, 10
- framework
 - bar-and-joint, 25–31
 - panel-and-hinge, 72–73

- generic position, 21, 22
- genus, 40
- Graphical Statics, 27
- Grassmann-Cayley algebra, 28, 58–61

- height propagation, 112
- hinge, 72
- hyperplane arrangement, 88
 - incremental construction of \mathbf{a} , 88

- incidence structure, 9, 10
 - obtaining the, 12–14
- indecomposable k -tensor, 59
- instantaneous center of rotation, 61
 - of two hinged bodies, 64
- instantaneous motion of a panel-and-hinge framework, 72
- instantaneous screw axis, 61
 - of two hinged bodies, 68
- interpretation, *see* lifting

- join operator, 58, 59
- junction, 9
 - dictionary, 16

- Kennedy-Aronhold theorem of the three centers, 65

- labelling
 - {+, −, →, ←}-, 14
 - {*boundary, interior*}-, 14
 - algorithms, 15–18
 - consistent, 13, 16
 - relationship with a motion assignment, 83
- lifting, 10
 - sharp, 10
 - trivial, 10
- line drawing
 - augmented, 9, 10, 14
 - correct, or realizable, 10
 - incorrect, 10
 - interpretation of *a*, *see* lifting
 - lifting of *a*, *see* lifting
 - raw, 9, 12, 93
 - triheral, 11
- Machine Vision, 1, 15, 21, 29
- Maxwell, James Clerk, 25
- meet operator, 58, 60
- motion assignment, 73
- motion of a point
 - in a pure rotation, 62, 66
 - in a pure translation, 63, 67
 - in planar kinematics, 61
 - in spatial kinematics, 65
- Necker reversal, 18
 - mathematical model, 84
- panel, 72
- parameterization
 - of all liftings of a line drawing, 10
 - of correct projections, 95, 98–99
- path of panels and hinges, 72
- Plücker coordinate, 55
 - vector, 55
 - of a line at infinity, 56
 - of a plane, 57
 - of a proper line, 56
- polydisk, 11
 - with holes, 11
- polygonal mesh, 108
 - triheral, 107, 112
- polyheral disk, *see* polydisk
- polyheral sphere, *see* polysphere
- polyheral surface, *see* polysurface
- polysphere, 11
 - polysurface, 11
 - triheral, 11
 - projective space, 54
 - vector space associated with *a*, 54
 - propagation wave, 112
 - pure condition, 30
- quadratic *p*-relations, 59
- realizability
 - for several topologies, 40–42
 - generic, 21–22
 - in floating-point arithmetic, 81
 - problem, 3, 125
 - superstrict test of, 23, 28
- reciprocal figure, 25
- reconstruction
 - problem, 4
- resolvable sequence, 96
- rigidity matrix, 76
- scene
 - of opaque polyhedra, 10, 35
 - polyheral, 2, 3
- screw axis, *see* instantaneous screw axis
- segment, 9
- self-stress, 25, 26, 28
 - strict, 25, 26, 28, 30
- singular value decomposition, 81
- sketch, 10, 93
- Solid Modelling, 107
- Structural Geometry, 27–31
- superstrictness, 22–25, 81, 95
- T and TT-transformations, 115–117
- TNPACK, 100
- unequivocal picture, 108–110
- vertex
 - of a polyheral surface, 11
 - of an incidence structure (or line drawing), 9