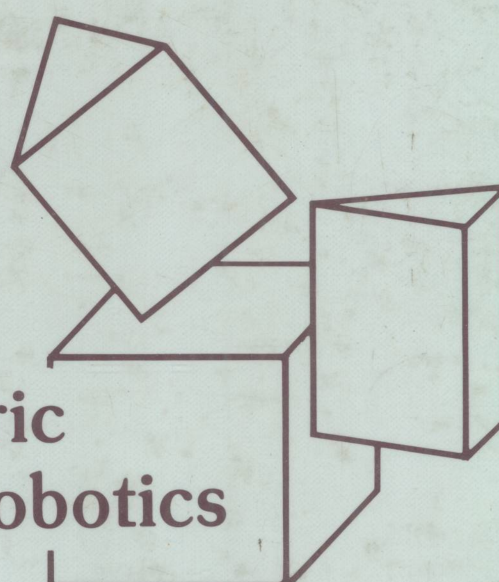
ADVANCES IN ROBOTICS

Volume I



Algorithmic and Geometric Aspects of Robotics

Edited by
Jacob T. Schwartz • Chee-Keng Yap

LEA

Algorithmic and Geometric Aspects of Robotics
Volume 1

Edited by

JACOB T. SCHWARTZ
CHEE-KENG YAP
Courant Institute of Mathematical Science
New York University

LEA LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
1987 Hillsdale, New Jersey London

Lawrence Erlbaum Associates, Inc., Publishers
365 Broadway
Hillsdale, New Jersey 07642

Library of Congress Cataloging in Publication Data

Algorithmic and geometric aspects of robotics.

(Advances in robotics ; vol. 1)
Includes bibliographies and indexes.
1. Robotics. 2. Robot vision. I. Schwartz, Jacob T.
II. Yap, Chee-Keng. III. Series: Advances in robotics
(Hillsdale, N.J.) ; vol. 1.
TJ211.A44 1987 629.8′92 86-23963
ISBN 0-89859-554-1

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Algorithmic and Geometric Aspects of Robotics

ADVANCES IN ROBOTICS

A series of books edited by
Chee-Keng Yap and **Jacob T. Schwartz**

SCHWARTZ and YAP
Algorithmic and Geometric Aspects of Robotics, Vol. 1

Preface

Although robotics has by now been a recognized area of computer science for several decades, it is only during the last few years that the level of theoretical work in it has begun to expand rapidly. Nevertheless, it seems certain that research in this area is destined to affect computer science profoundly. Till now, computer science has been largely combinatorial and symbolic, having the manipulation of patterns and tables of data as its principal content. In robotics, however, computer science makes contact with real-world geometric and physical phenomena such as the compliance of elastic bodies, the frictional phenomena which occur when bodies come in contact, errors in modeling which are inevitable in the real world, the sudden changes of state which occur when bodies collide unexpectedly, and so forth. It is to be expected that much interesting new computer science will emerge from contact with these rich conceptual domains and, in particular, that computer science will become more traditionally mathematical and "continuous."

Fields of science have their own internal rhythms, in which periods of slow progress conditioned by a lack of ideas or by the exhaustion of old ideas alternate with the excitement of rapid advance triggered by conceptual breakthroughs or by maturation of supporting technology. Sustaining technological development and systematic conceptual advance often go together and reinforce one another. After its slow start during the past several decades, robotics stands at the start of a period of rapid advance, current theoretical and pragmatic developments foreshadowing major progress in many of its subfields. The massive computational power created by VLSI technology is a significant driving force: By making computing cycles available in whatever quantity required, the work of VLSI designers is rapidly creating most of the purely "electronic" side of the technological base which robotics will require. Armed with this technology, robotics researchers have begun to perceive ways of radically strengthening the basic capabilities of robots, e.g., their ability to see, to manipulate, and to plan. As these capabilities are improved, additional work integrating them into composite software environments facilitating robot use must also follow.

In recognition of the increasing importance and pace of work in this area, the present volume initiates an annual series of publications which will track and reflect the progress of robotics over the coming years. We have chosen to begin with a relatively theoretical subarea, namely automatic planning of robot motions, which has attracted considerable attention of late and which serves well to illustrate the depth and a variety of the mathematical and geometric issues arising out of robotics. The problem here is to develop algorithms that will allow a robot which knows the geometry of the environment in which it must move to plan the details of its motion automatically. Moreover, if the robot is grasping a body (of known geometry) it must allow for this in the motion it plans. As the chapters in the present volume show, this motion planning problem has begun to yield to the efforts of theorists and algorithm designers who have found it possible to apply methods developed by topologists and algebraic geometers to this practical area.

However, this is only one of many theoretical and pragmatic problems in a very broad field. The general aim of robotics can be defined as the mechanization of that elementary *operative* intelligence which people use unthinkingly in locating and handling ordinary objects. Such research has two principal aspects: sensory and manipulative. Sensory studies aim to develop techniques which make it possible to organize the raw data gathered by sensors such as video cameras and ultrasonic rangefinders into perceptually meaningful gestalts. Studies of manipulation deal with all the tactics and strategy needed to control bodies moving slowly or rapidly through three dimensional space, both when the controlled (robot) bodies must move avoiding contact with other bodies or obstacles in their environment, and also when the controlled robot bodies need to make contact with portions of their environment or with other robots, e.g., to grip an object which is to be moved, to insert a peg into a hole, etc. Control of highly dynamic motions is another intriguing area of the *manipulative* or *dynamic* side of robotics; control of delicate and dextrous motions is yet a third. Work in this latter area aims at robots that can adjust smoothly and simply to the shapes and physical behavior of delicate 3-D bodies. For example, one wishes to be able to grasp an egg, either to draw some figure on it with a stiff pen, or to carry it to the edge of a cup, crack it, and then (more dynamically) pour its contents into the cup. Here a variety of problems arise. Sophisticated multidimensional feedback control methods are needed and are rapidly being defined. Work in computational geometry is elucidating the interesting geometric issues involved in management of mechanisms with many degrees of freedom. Theoretical attention is being directed to one of the most neglected areas of classical physics, the analysis of the frictional motions of rigid and flexible bodies. Interesting robot hands, which will provide appropriate levels of experimental challenge to control theorists, computational geometers, and robotic software designers, are being developed by several engineering groups.

The enormous field of robot vision illustrates the sensory side of robotics. The problem here is to find techniques which make it possible to organize the raw data gathered by a video camera into perceptually meaningful gestalts. A wide variety of approaches to this deep problem are available. For example, in *model-based* vision studies, attention is confined to scenes containing only known objects or objects belonging to known parametric classes (e.g., cylinders with spherical caps and cylindrical holes bored in them, but of heights and radii not known a priori). This contrasts with *general* vision studies, which aim to impose helpful perceptual groupings on entirely general scenes, e.g., landscapes containing shrubbery. The great advantage of the first problem is that it is entirely objective: Its aim is simply to reduce a scene known to contain objects drawn from a fixed finite set to a table giving the identities and orientations of all objects actually present. In contrast, the deeper *general* vision problem has inherently psychological aspects: Here one aims to devise (the image-analysis portions of) a robot "eye" whose perceptual groupings are close enough to those formed by the human eye for easy communication and mutual understanding to be possible. Even the narrower *model-based* vision problem can be put at various levels of difficulty, to which theoretical and experimental work can be expected to advance progressively over the next few years. Specifically, one can consider:

(i) images of either 2D or 3D bodies, which can either been seen in isolation or as parts of compound scenes.

(ii) the bodies with which one deals can either be wholly visible or partially obscured, and can be present either in constrained or in perfectly general orientations.

(iii) the bodies seen can either be stationary or can be allowed to move.

(iv) the bodies seen can either conform exactly to their models, or can be affected by extra error features such as "burrs," "dents," "flash," etc.

Recent work makes successful treatment of all these problems appear feasible.

Vision studies are also conditioned by the form of input they assume, and a variety of schemes have been developed for acquiring information-rich images when use of simpler images complicates the object identification problem. The images from which a model-based approach works can either be:

(a) ordinary intensity images
(b) high quality silhouettes, obtained, e.g., by "backlighting" a scene
(c) "depth images," in which each pixel records the true geometric distance

of an observed point P on a body surface from the camera, or, equivalently, P's true geometric position in 3-dimensional space.

It appears likely that effective analysis of all of these kinds of images will be possible provided that only finitely many objects of shapes known a priori can be present in the scene being viewed.

The topics briefly noted in the preceding paragraphs, plus many others that can be expected to arise as new aspects of the rapidly expanding field of robotics research, will be subjected to detailed scrutiny in the series of volumes hereby initiated. Early volumes dedicated to such subareas as robot vision, control-theoretic aspects of robotics, robot kinematics, robot software systems and programming languages, as well as the theoretical aspects of such apparently pragmatic subjects as parts mating and other manufacturing-science related issues will be discussed. It is hoped that these volumes succeed in conveying the interest and excitement of this relatively new field to the widest possible audience.

J. T. Schwartz

Contents

Preface ix

Introduction 1

Chee-Keng Yap

1. Why Robotics? *1*
2. An Overview of the Contents of the Volume *3*
3. Brief Review of Standard Terminology *4*

1. The Challenge of Robotics for Computer Science 7

John E. Hopcroft and Dean B. Krafft

1. What is Robotics? *7*
2. A New Science *9*
3. Representations for Physical Objects and Processes *10*
4. Manipulating Object Representations *22*
5. Reasoning *28*
6. Conclusions *40*

References *40*

2. Computational Geometry—A User's Guide 43

David P. Dobkin and Diane L. Souvaine

1. Introduction *43*
2. Hierarchical Search *44*
3. Hierarchical Computation *66*
4. Geometric Transformations *77*
5. Conclusion *90*

References *91*

3. Algorithmic Motion Planning **95**
Chee-Keng Yap

1. History *97*
2. Basic Concepts *99*
3. A Taxonomy of Motion Planning Problems *103*
4. Issues Related to Algorithmic Motion Planning *106*
5. Moving a Disc *108*
6. Moving a Ladder *112*
7. Two Approaches to Motion Planning *119*
8. Solution to the General Motion Planning Problem *123*
9. Retraction Via Cell Complexes *129*
10. Lower Bounds *132*
11. Summary: New Directions and Open Problems *137*
12. Conclusion *140*

References *141*

4. Approximation and Decomposition of Shapes **145**
Bernard Chazelle

1. Introduction *145*
2. Approximation of Shapes *146*
3. Decomposition of Shapes *165*
4. Epilogue *180*

References *181*

5. Intersection and Proximity Problems and Voronoi Diagrams **187**
Daniel Leven and Micha Sharir

1. Introduction *187*
2. A Simple Intersection Detection Algorithm in Two-Dimensional Space *190*
3. Generalized Planar Voronoi Diagrams *192*
4. Applications of Voronoi Diagrams *197*
5. Efficient Construction of Voronoi Diagrams *202*
6. Dynamic Intersection and Proximity Problems *216*
7. The Three-Dimensional Case *216*

References *226*

6. Fleshing Out Wire Frames: Reconstruction of Objects, Part I **229**
George Markowsky and Michael A. Wesley

1. Introduction *229*
2. Basic Concepts *231*
3. The Wire Frame Algorithm *237*
4. Examples *252*

Appendix A: Topological Concepts *256*
References *258*

7. Fleshing Out Projections: Reconstruction of Objects, Part II **259**
Michael A. Wesley and George Markowsky

1. Introduction *259*
2. Basic Concepts and Results *261*
3. Fleshing Out Unlabeled Projections *269*
4. Additional Information from Drawing Conventions *280*
5. Examples *282*
6. Summary *292*

References *295*

AUTHOR INDEX **297**
SUBJECT INDEX **303**

Introduction

Chee-Keng Yap

1. WHY ROBOTICS?

Robotics is the discipline concerned with the science and technology of computer controlled autonomous mechanisms. It includes both an *effectuator* side (which focuses on such matters as the geometry and control of manipulator arms, grippers, robot locomotion and balance, etc.) and a *sensory* side (concerned with robot vision, tactile sensation, object identification using sensory inputs in visual, tactile, or other modalities such as ultra sound). As this remark indicates, robotics is highly interdisciplinary in character, combining computer science (e.g., real-time software design, computer graphics, computational geometry, design of special-purpose computational devices), engineering (design and control of mechanisms and of sensors), and applied physics (materials science, theory of elasticity and of friction.) The *Advances* series which this volume initiates will aim to serve the many academic and industrial communities whose skills must play a role in its healthy development, and will cover all parts of this extensive field. We plan to address issues in computer science, theoretical mechanics and mechanism design, control theory, sensor technology and analysis of sensory data, and automated assembly technology, insofar as these contribute to robotics proper.

This first volume concerns itself with the algorithmic and geometric side of robotics, an area of the subject whose technical sophistication has increased very greatly in the last few years. The techniques reviewed reflect three fundamental appreciations that have grown out of 2 decades of work on *algorithm design* and *complexity theory*:

(a) Suitably ingenious design of key computational procedures can increase their efficiency far beyond the levels which naive "first cuts" at these same procedures might lead one to think possible;

(b) In the design of high-quality algorithms, formal analysis of procedure efficiency is required to quantify success (or to pinpoint remaining opportunities for efficiency improvement):

(c) Certain computational tasks are inherently intractable (and therefore should never by attempted in full generality if they are to be used in larger computations that aim to be practical).

During the past decade these fundamental insights have inspired rapid growth of many fields of algorithm design, including the very new field of *computa-*

tional geometry. Most recently the methods developed by computational geometers have been applied to robotics (whose extensive geometric content is obvious), and conversely robotics has begun to enrich the range of problems that challenge the geometer.

Each of the review articles collected in this initial volume focuses on some aspect of the fruitful interaction between robotics and computational geometry. Many of the methods presented are applicable not only to robotics, but also to such other related areas as geometric modeling and simulation to computer aided design and manufacturing, and (to a more limited degree) to computer vision and pattern recognition. Moreover, though necessarily specialized, the techniques described in these articles exemplify the beautiful interaction between data structures and geometric properties that are often key to attaining efficiency in computational geometry, and will reward their careful readers by suggesting general efficiency paradigms applicable to many geometric problems other than those specific to the present volume.

From a deeper point-of-view, the algorithms presented by our authors highlight the fundamental difficulties of robotics by pinpointing some of the very challenging problems that arise whenever one attempts to make a computer duplicate even the most elementary human manipulative or sensory activities. Consider, for example, the problem of using a robot hand to move an object from a given starting position (in a cluttered environment) to a specified target position. Though trivial for humans, even this elementary task raises many challenging algorithmic subproblems: If the position of the object or the geometry of the environment is not fully known, we may have to utilize some sort of sensory device and find some appropriate way of analyzing its output to supply the missing information. Once this information is available, we need to generate a plan for grasping the object, a task which involves analysis of the geometry of the object in relationship to the geometry of the moving manipulator and its grasping hand, and beyond this requires us to analyze the state and frictional reactions of an object to which forces are applied at several points. After deciding on a grasp, we need to plan a motion from start to finish position of the hand-object composite which results. All this must be done with acceptable efficiency, generality, and robustness.

Some of this list of problems belong to the domain of computational geometry to which this volume is devoted. Others raise entirely different issues, relating, e.g., to computer vision, statics, dynamics, theory of frictional movements, etc.; subject areas to be treated in subsequent volumes in this series. Finally, practical application of the algorithms and analyses generated by detailed study of the special problems raised by such practical tasks as object transport by robots creates extensive software engineering problems, an aspect of robotics which also will be reviewed in subsequent volumes in this series.

The foregoing reflections justify the claim that robotics has a truly rich scientific content and is likely to exert a major influence on computer science during coming years.

2. AN OVERVIEW OF THE CONTENTS OF THE VOLUME

The seven chapters comprising this volume review new work on the geometric side of robotics. It is hoped that these articles will serve the needs of the growing community of students and researchers wishing to acquire systematic knowledge of recent developments in this field.

Hopcroft and Krafft (*The Challenge of Robotics for Computer Science*) define the fundamental goal of robotics in very broad terms: to develop techniques for the *representation of, and formal reasoning about, physical objects and physical processes*. In accordance with this view of the field, they propose the name *stereo-phenomenology* as a better name for it than the too application-oriented term *robotics*. They emphasize the view that the concerns of this new field are likely to involve all properties of systems of solid objects amenable to mathematical formalization (and thus ultimately amenable to automated reasoning), including not only geometric properties, but also physical properties such as force, mass, velocity, energy, friction, vibration, etc., as well as more special materials-related properties. They outline a staggering research agenda, each of whose items may well come to occupy a generation of researchers.

Dobkin and Souvaine (*Computational Geometry—A User's Guide*) present some of the recently developed techniques that have begun to address the geometric side of the research agenda propounded by Hopcroft and Krafft. Their article illustrates some of the most basic paradigms of efficient algorithm design in the geometric area: divide-and-conquer, hierarchical search, duality transformation, ordered scanning, etc. Although some of these ideas, and the data structures which carry them, may be familiar from other fields of algorithm design, their applications in computational geometry are often remarkable, unexpected, and of great elegance.

Yap (*Algorithmic Motion Planning*) reviews several applied geometric ideas that have been particularly central to recent work on the important problem of motion planning. He describes the two main approaches, the so-called *retraction* and *decomposition* methods, on which systematic algorithmic treatments of the motion-planning problem have been based, and discusses some of the technical problems involved in developing efficient algorithms within the general frameworks defined by these approaches. The manner in which complexity theory techniques can be used to bound the difficulty of some robotic problems from below is also described and illustrated. The chapter closes with a list of open problems.

Chazelle (*Approximation and Decomposition of Shapes*) develops a theme in computational geometry having obvious significance for the simplification of practical robotics problems, namely approximation or decomposition of complex geometric objects by (or into) simple ones. For example, given a polygon P we can consider the number of sides of P as a measure of P's complexity. Then, for any P, one can ask for that other polygon, of a specified smaller number of sides, which best approximates P. Such a reduction can be very useful in applications, since the simplified objects may be substantially easier to manipulate. Another possibility is to decompose a complex object into several simple objects, e.g., to decompose a nonconvex polygon or polyhedron into a minimal number of convex figures, which may or may not be allowed to overlap. This chapter is one of the first expositions to provide a systematic account of such *decomposition* and *covering* problems, and of the related *enclosure* problem namely that of including a specified figure optimally in some larger figure drawn from a specified class.

Sharir and Leven (*Intersection and Proximity Problems and Voronoi Diagrams*) deepen a theme opened in Chapters 2 and 3, namely the use of Voronoi diagrams (which are defined to be the subspaces consisting of all those points, lying within larger abstract spaces of "positions," which represent "critical positions" in one or another sense, e.g., positions simultaneously nearest to several of a given collection of points or other geometric elements). Such diagrams have turned out to be among the most versatile and effective structures in computational geometry, having application in such diverse areas as motion planning, geometric searching, pattern recognition, computational fluid dynamics, solid state physics, chemistry, and statistics. Sharir and Leven review some of the most important applications of this technique to robotics and describe many of the most efficient algorithmic techniques known for calculating and for applying Voronoi diagrams.

Markowsky and Wesley (*Fleshing Out Wire Frames and Projections*) treat an intriguing class of geometric "reconstruction" problems having immediate application to computer-aided geometric design systems. Given any polyhedral object, we can easily derive a "wire frame" representation for it by dropping all its faces and retaining only its edges and vertices. (When one needs to enter the object into a mechanical design system, it may be significantly easier to supply just its edges than to define the object's faces explicitly.) The first problem considered by Markowsky and Wesley is that of reconstructing the object's surfaces, given only its wire frame. Since there can exist intriguing ambiguities of reconstruction, it is best to require the algorithm to compute all possible reconstructions. A second related problem is to compute the same representation given only several two-dimensional projections of an object's wire frame. The two Markowsky–Wesley chapters give elegant, general, and complete solutions for these problems; the algorithms that they describe have already begun to find practical application.

A Brief Review of Standard Terminology

Readers may find this brief review of concepts of complexity theory helpful. By a *complexity function* we mean a real function f of a real variable, where the range of f may include ∞.

Definition. (*Big-oh notation*) If f, g are two complexity functions, we say that f *dominates* g if for some n_0, for all $x \geq n_0$, $g(x) \leq f(x)$. We write $O(f)$ for the set of all complexity functions g such that for some constant $C > 0$, g is dominated by $C \cdot f$.

'$O(f)$' is read 'big-oh of f' or 'order of f'. A typical idiom using this notation is 'O(1)' which denotes the set of functions that are bounded by some constant, e.g., $f(n) = 1/n$ is in $O(1)$. Two conventions customarily govern the use of this notation:

(i) The sum $F + G$ of two sets F, G of complexity functions is defined as the set of functions of the form $f + g$ where $f \in F$, $g \in G$. Similarly for the expressions $F - G$, $F \cdot G$, F^G and $F \circ G = \{f \circ g : f \in F, g \circ G\}$ where $f \circ g(n) = f(g(n))$ denotes functional composition. If F consists of a single function f then we use 'f' rather than the correct '$\{f\}$' in such expressions; for instance, $O(n^2) + O(n \log n)$. Another example is $n^{O(1)}$, denoting the set of functions g that is dominated by the function n^k for some $k \geq 1$.

(ii) One often writes equations in which the O-notation appears on one side or on both sides of the equality symbol. Examples are $n = O(n^2)$ and $O(n \log_2 n) = O(n^2)$. The expressions on *both* sides of the equation should be interpreted as denoting sets of functions and the 'equality' sign should actually be interpreted as set inclusion '$\subseteq$'. This so-called 'one-way equality' is clearly transitive but not reflexive. For example, $n = O(n \log_2 n)$ is true even though $n \log_2 n = O(n)$ is not.

If $f = O(g)$ and $g = O(f)$ then we write $f = \Theta(g)$ and read "f is big-omega of g". Clearly the big-omega relation is an equivalence relation. We also say f and g are *Θ-equivalent* if $f = \Theta(g)$. In algorithmic analysis, it is usual to distinguish complexity functions only up to Θ-equivalence. It is this convention that allows us, for instance, to use logarithms without regard for the base of the logarithm. The rationale for this convention is that the complexity of a program required to realize a given algorithm can often be changed up to some small constant factor by rather trivial modifications in the programming language used; however such the changes remain within the Θ-equivalence class of the original function. An example of a trivial modification is where we allow one instruction in the new language to stand for a fixed sequence of instructions of the original language. Thus, up to Θ-equivalence, the complexity of realizing a given algorithm is typically independent of the details of the machine on which is run or the actual programming language involved.

We say a programming language L can *O-simulate* another programming language L' if for any program with complexity f in language L' there is another equivalent program in language L' with complexity $O(f)$. Say L and L' are *Θ-equivalent* if they can O-simulate each other. Even if two languages are not Θ-equivalent, it often turns out that they can simulate each other up to a polynomial