Preface

The advent of fast and inexpensive consumer graphics hardware has led to an increased demand for knowledge of how to program various geometric tasks for applications including computer games, scientific visualization, medical image analysis, simulation, and virtual worlds. The types of applications are themselves evolving to take advantage of the technology (Crawford 2002) and even include 3D environments for the purposes of code analysis and visual debugging, and analysis of coalition formation of political parties by representing the party beliefs as convex objects whose intersections indicate a potential coalition.

It is possible to find much of the graphics knowledge in resources that are scattered about, whether it be books, Web sites, newsgroups, journal articles, or trade magazines. Sometimes the resources are easy to comprehend, but just as often not. Sometimes they are presented with enough detail to illustrate underlying principles, sometimes not. Sometimes a concept is presented with an eye toward numerical issues that arise when using floating-point arithmetic, yet in other cases the concept is presented only in its full theoretical glory. Correctness of the presentation can even be an issue, especially with online articles. The time spent in locating the resources; evaluating their relevance, effectiveness, and correctness; and adapting them to your own needs is time better spent on other tasks. The book is designed with this in mind. It provides you with a comprehensive collection of many of the two- and three-dimensional geometric algorithms that you will encounter in practical applications. We call these *geometric tools* since, after all, the algorithms and concepts really are tools that allow you to accomplish your application's goals.

The level of difficulty of the topics in this book falls within a wide range. The problem can be as simple as computing the distance from a point to a line segment, or it can be as complicated as computing the intersection of two nonconvex, simple, closed polyhedra. Some of the tools require only a few simple concepts from vector algebra. Others require more advanced concepts from calculus such as derivatives of functions, level sets, or constrained minimization using Lagrange multipliers. Generally a book will focus on one end of the spectrum; ours does not. We intend that this book will be used by newcomers to the field of graphics and by experienced practitioners. For those readers who require a refresher on vector and matrix algebra, we have provided three gentle chapters on the topics. Various appendices are available, including one summarizing basic formulas from trigonometry and one covering various numerical methods that are used by the tools.

The book may be used in two ways. The first use is as a teaching tool. The material is presented in a manner to convey the important ideas in the algorithms, thus making the book suitable for a textbook in a college course on geometric algorithms for graphics. Although the book comes without exercises at the end of the sections, it does come with a lot of pseudocode. An appropriate set of assignments for the course could very well be to implement the pseudocode in a real programming language. To quote a famous phrase: the proof is in the pudding.

The second use for the book is as a reference guide. The algorithms chapters are organized by dimension, the two-dimensional material occurring first, the three-dimensional second. The chapter on computational geometry is a mixture of dimensions, but is better grouped that way as a single chapter. The organization makes it easy to locate an algorithm of interest. The attempt at separation by dimension comes at a slight cost. Some of the discussions that can normally be written once and apply to arbitrary dimensions are duplicated. For example, distance from a point to a line segment can be described in a dimensionless and coordinate-free manner, but we have chosen to discuss the problem both in two dimensions and in three dimensions. We believe this choice makes the sections relatively self-contained, thereby avoiding the usual reader's syndrome of having multiple pieces of paper or pens stuck in various locations in a book just to be able to navigate quickly to all the sections relevant to the problem at hand!

Inclusion of working source code in a computer science book has become common practice in the industry. In most cases, the code to illustrate the book concepts can be written in a reasonable amount of time. For a book of this magnitude that covers an enormous collection of algorithms, a full set of code to illustrate all the algorithms is simply not feasible. This is a difficult task even for a commercial venture. As an alternative, we have tried to add as much pseudocode as possible. The bibliography contains many references to Web sites (valid links as of the first printing of the book) that have implementations of algorithms or links to implementations. One site that has many of the algorithms implemented is *www.magic-software.com*, hosted by Magic Software, Inc. and maintained by Dave Eberly. The source code from this site may be freely downloaded. This site also hosts a Web page for the book, *www.magic-software.com/GeometricTools.html*, that contains information about the book, book corrections, and an update history with notifications about new source code and about bug fixes to old source code. Resources associated with the book are also available at *www.mkp.com/gtcg.*

We want to thank the book reviewers, Tomas Akenine-Möller (Chalmers University of Technology), Ian Ashdown (byHeart Consultants Limited), Eric Haines (Autodesk, Inc.), George Innis (Magic Software, Inc.), Peter Lipson (Toys for Bob, Inc.), John Stone (University of Illinois), Dan Sunday (Johns Hopkins University), and Dennis Wenzel (True Matrix Software), and the technical editor, Parveen Kaler (Simon Fraser University). A book of this size and scope is difficult to review, but their diligence paid off. The reviewers' comments and criticisms have helped to improve many aspects of the book. The input from Peter and Dennis is especially appreciated since they took on the formidable task of reading the entire book and provided detailed comments about nearly every aspect of the book, both at a low and a high level. David M. Eberle (Walt Disney Feature Animation) provided much of the pseudocode for several chapters and some additional technical reviewing; his help is greatly appreciated. We also want to thank our editor, Diane Cerra, and her assistant, Belinda Breyer, for the time they spent in helping us to assemble such a large tome and for their patience in understanding that authors need frequent encouragement to complete a work of this magnitude. The success of this book is due to the efforts of all these folks as well to ours. Enjoy!