Computational Geometry and Geographical Information Systems

Jörg-Rüdiger Sack
School of Computer Science
Carleton University
1125 Colonel By Drive
Ottawa, ON., Canada K1S 5B6
E-mail : sack@scs.carleton.ca

Abstract

In this report I have taken an applied area, namely Geographic Information Systems (GIS), and compared the research work in this area with the field of Computational Geometry. A number of suggestions are made which might help to enhance the impact of Computational Geometry on GIS and other areas, as well as enrich Computational Geometry with some fundamental research problems.

1 Comparison Between GIS and CG

A Geographic Information System is a computerized system for gathering, storing, presenting, manipulating, and interpreting geographic data for phenomena below, on or above (but close to) the earth surface. Computational Geometry is the study of geometric algorithms under the framework of the design and analysis of algorithms.

Since both areas deal with geometric computation, it is natural to expect many common features. Despite this expectation, the respective research communities have mostly developed their understanding of geometric computation independently. This could be largely attributed to the different ways of looking at geometric computation.

In GIS, mostly geometric data is large, inaccurate, and often degenerate. GIS algorithms typically concentrate on reducing the space complexity, should have good average case performance, their output is often allowed to have errors within some tolerance limit, and algorithms should avoid complex data structures whenever possible. Most GIS literature is aimed at implementation-oriented researchers, presents details on experimental set-up (e.g., the hardware and GIS data-files used), and avoids technical details (positively lemmas and theorems). Much of the research work reported in the GIS literature comes from ideas which have been implemented.

Whereas in Computational Geometry, traditionally, the data is not that large, is accurate, and usually the assumptions have been made that there are no degeneracies in the data. In computational geometry algorithms, the geometric structures, the worst-case complexity, the correctness and the analysis are central issues; the space complexity is not a major issue. Furthermore, most of the computational geometry literature is aimed at "theoreticians", with a strong emphasis on mathematical rigor and usually with little emphasis on implementation set-up and detail.

2 Suggestions

Here are my main suggestions.

- GIS: Having worked now for several years on GIS R&D projects, I feel that there are a number of GIS tasks which could benefit from the research done in Computational Geometry and that there are problems in GIS which open up new areas of research within Computational Geometry. The execution time of some important GIS tasks (such as overlays, kriging) is far from satisfactory. Similarly several central issues in GIS require better understanding, e.g., data structures, generalizations, spatial accuracy, time and spatial analysis.

- Integrated Environment: There is a need to develop an open integrated environment for developing Computational Geometric Applications. This environment should have relevant components from many areas in computer science, e.g. Data Structures, Data Bases, Graph Algorithms, and Parallel Computing. Users within the computational geometry community as well as the application developers will be able to benefit from an environment, by either exporting relevant pieces of code into their own systems, or by building application on top of this environment. There are several sequential C.G. environments, e.g., the LEDA Project, WorkBench for Computational Geometry, and XYX Geo-Bench. Recently the need for algorithmic toolkits has been advocated by many researchers (e.g., The Algorithm Tool Kit Project suggested in the report by Karp et al.).

- In-accessibility: Most of the "optimal" computational geometric algorithms are inaccessible to researchers from outside the computational geometry community. This is largely due to the way we present our results, complex techniques and data structures used, hiding the constants inside the O-notation, and usually not providing insight into implementation. Some of these issues can be addressed by

1. Providing expository articles also, describing the research papers from the implementors point of view.
2. Providing the code and integrating it into existing geometric libraries. Also setting up and maintaining a WWW-database of pointers to available geometric code (e.g., Nina Amento's page on geometric code).
3. Providing run-time comparisons with other algorithms.
4. Providing the constants.

- Dissemination of Knowledge: The field of Computational Geometry has matured, but unfortunately, in spite of its potential, may have gained more from other areas than it has provided. (The above may explain this to some degree.) I feel that there is a need for more dissemination of our knowledge. This could be done in several ways.

  1. Expository Articles: By providing expository articles for a broader audience, for example in the Handbook style (there are two Handbooks currently in preparation, CRC Handbook (Ed. J. Goodman and J. O'Rourke) and Elsevier Science, North Holland, (Ed. J. Urrutia and myself)). Similarly, we could publish "popular" articles on our research work (e.g., in Mathematics Intellegencier, American Mathematical Monthly, Programming Columns, Scientific American)).
  2. Education: There is a need for introducing courses and programs, which bridge the gap between theory and applied areas, within our university set-up. This could be achieved in several ways, e.g., theory students are exposed to applied area courses and vice-versa. As part of the degree requirements, theory students are required to complete projects in applied areas. Also, within the theory community, interaction with applied areas, should be encouraged in whatever possible manner. Joint degree problems may also be fostered.
  3. Publications: Conferences, Workshops and Journals should encourage research work dealing with the dissemination of theoretical knowledge to applied areas.

- Reward Structure: Clearly, providing implementation results and comparisons requires substantial research effort as well as time investment. So a suitable reward structure should be devised. There are several ways of achieving this.

  1. Publications: Journals could seek research articles, aimed at implementation, as well as issues arising from implementation. These would be refereed according to acceptable standards, and code could

be made available over the web or through a journal. Code developed receives a "seal of approval" by the journal and the article become citable in CVs. The difficulty lies in the refereeing process. Already several conferences, e.g. SODA, SPAA, ACM-CG, ESA, WADS, have started accepting papers of the above nature. Similarly journals, Intl. Journal on Comp. Geom. and Applications (Ed. D.T. Lee), Comp. Geometry: Theory and Applications (Ed. J. Urrutia and myself), and several special issues in other journals, are seeking such papers.

2. Geometric Library: There is no doubt that having good geometric code is invaluable. Very likely, in near future, one can think of commercially marketing implementation and geometric libraries. This may provide an incentive for developing high quality code and implementation oriented research, in general. There is a need for commercially, professionally managed geometric libraries. Some institutions and industry do not trust code developed in Universities; maintenance and continuity of the product are other important issues here.

- Benchmark Data and Test Data: There is a need for having a spectrum of standardized benchmark data. This will help evaluating implementations. Likewise, there is a need for standardized test data, having all possible degeneracies and which should reflect the data in "reality". This will help evaluating software from the application point of view. The creation of such a benchmark library is an interesting research topics in itself and will provide a better understanding of the applied areas.

Lastly, the discussion on implementation, applications, etc. should never overshadow the importance of fundamental, theoretical work in the field.

Acknowledgement: I thank Anil Maheshwari for his input into this document.