GeoMeter: A System For Modeling and Algebraic Manipulation

C. Connolly, D. Kapur,
J. Mundy and R. Weiss

COINS TR 89-99

September 1989

GeoMeter: A System for Modeling and Algebraic Manipulation

C. I. Connolly* D. Kapur† J. L. Mundy‡

R. Weiss*

Computer and Information Science Department

University of Massachusetts at Amherst §

Artificial Intelligence Program

GE Corporate Research and Development Center ¶

September 25, 1989

Abstract

The *GeoMeter* modeling system is described. The system is designed to manipulate solid models for a variety of purposes. *GeoMeter* also supports polynomial and transcendental function manipulation, including methods for solving systems of polynomial equations. The applications for such methods in the context of solid modeling and computer vision are also discussed.

1 Introduction

GeoMeter [17] is a system written in Common Lisp for the purpose of modeling solid objects and providing tools for algebraic manipulation. The original motivation for *GeoMeter* was as a library to support experiments in Computer Vision, although its uses are by no means limited to that application. It is the result of several years of effort in both the Image Understanding Laboratory at the GE Research and Development Center, and more recently in the

*University of Massachusetts at Amherst

†State University of New York at Albany

‡GE Research and Development Center

§Supported in part by the following: NSF/CER Grant DCR8500332, DARPA Grant F30602-87-C-0140

¶Supported in part by the following: DARPA Contract DACA76-86-C-007, AFOSR Contract F49620-89-C-003

VISIONS Group at the University of Massachusetts in Amherst. Others who have been affiliated with this software are now at the Rensselaer Polytechnic Institute and the State University of New York at Albany.

Existing uses of *GeoMeter* include robot navigation [18,19], model matching [33], model construction from image data [12], proofs of geometry theorems using algebraic techniques [15], and computation of generic view information [20]. *GeoMeter* is written in Common Lisp, and has been compiled and tested on TI Explorers, Symbolics Lisp Machines, VAXLisp, and Suns under Lucid Lisp. Work is underway to allow *GeoMeter* to run on the Sequent Balance 2000 series computer.

Many of the functions and data structures in *GeoMeter* have close counterparts in mathematics. The implementors attempted to approach classical mathematical terminology in naming functions and data structures. The intent was to keep interested users from being bewildered by a deluge of nonstandard terminology. In addition, it allows users to resort to their own mathematical references for clarification of certain concepts, when desired.

1.1 Representations

There are many different representations for encoding the shape and three-dimensional structure of objects. All of them impose some type of restriction on the surfaces that can be described. For example, ACRONYM uses generalized cylinders as the basic primitive [5], SuperSketch uses superquadrics [30], some are specifically polyhedral [32] while others provide multiple primitives [6] For example, the Designer system [31] has rectangular blocks together with spheres, cylinders, and tori.

In *GeoMeter*, the language of simplicial complexes in algebraic topology [16,21] has been adopted for describing surfaces. It provides generality and an explicit representation of edges, vertices, and faces. Each of these serve as a type of geometric primitive, and can be parameterized as a smooth function from a point, unit interval, and triangle to $\mathbf{R}^3$, respectively. For example, a standard 0-simplex is a point, a 1-simplex is a straight line segment, and a 2-simplex is a triangle (see figure 1). In the usual mathematical approach, a smooth n-simplex is a differentiable map from the standard n-simplex to a subset of $\mathbf{R}^3$, and the images of these 0-, 1-, and 2-simplices correspond to the vertices, edges and faces of a surface. Surfaces are thus constructed as the union of these primitives, and are denoted by an algebraic sum of simplices. This representation produces a triangulation of the surface, where the triangles are not necessarily planar. Each smooth simplex determines an orientation on its image, i.e. a choice of the direction of the normal at each point. It is worth noting that the theory of Algebraic Topology provides operations for determining whether the triangles in a simplicial complex fit together to form a closed surface. This theory provides the foundation for many of *GeoMeter*'s operations.

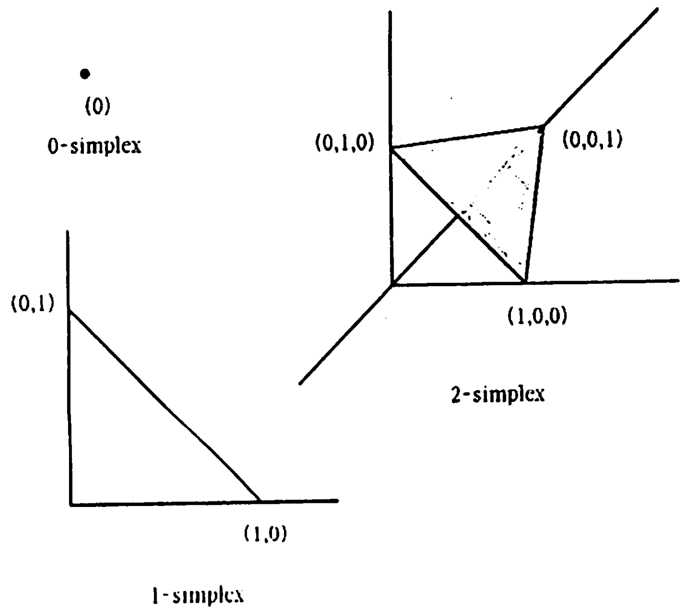

Figure 1: 0-, 1-, and 2-simplices

2 GeoMeter Structure

GeoMeter has two basic parts: a geometric section, and an analytic section. The geometric section consists of those functions and data structures which are used to describe physical objects. The analytic section consists of functions and data structures used in manipulating polynomials and transcendental functions.

2.1 Geometric section

The three basic entities which *GeoMeter* uses to represent sets are the vertex, the edge, and the face. These entities are composed to represent solid objects. The vertex is a 0-dimensional primitive which has an x, y, z position in space. An edge is a 1-dimensional set defined by two vertices (if linear). Edges can also be defined by three bounded univariate functions (if parametric) or as the planar zero set of a bivariate polynomial (if implicit). Linear edges have a direction vector and a normal vector (defined with respect to the origin). A face is a 2-dimensional set defined by a collection of edges. In *GeoMeter*, faces can also be defined parametrically and implicitly. Planar faces have a normal vector and a transformation matrix to define their coordinate system.

Interleaved with the faces, edges, and vertices are topological structures which are used to define the connectivity of sets in the model. A 0-chain is a set of vertices from which an edge can be defined. A 1-chain is a set of edges from which a face can be defined. A 2-chain is a set of faces which can be used to define a surface. An important concept in forming closed surfaces, i.e. objects, is the definition of the boundary. Every 2-chain has a boundary which is a 1-chain. If the boundary of a 2-chain is 0, then the 2-chain is said to be a 2-cycle. Similarly, if the boundary of a 1-chain or 0-chain is 0, it is a cycle. Each of *GeoMeter*'s chain structures can be used to represent cycles. A 1-cycle, i.e., a chain in which every vertex is used on exactly two edges, forms one or more polygons in the plane. [1] Likewise, a 2-cycle defines one or more polytopes.

Objects are built hierarchically starting with vertices. Vertices can be added together to form a 0-chain, a 0-chain with two points can be used to create an edge, edges can be added together to form a 1-chain, 1-chains can be used to create faces, etc. Usually, for computational simplicity, straight lines are used for edges and planes for faces, so that curved surfaces are approximated by polyhedra. Models can also be built by joining faces along common edges. A hinge function enforces the constraints that the edges must be aligned. Due to the generality of the mathematical framework, it is possible to represent semi-algebraic curves and surfaces, and there are some procedures for manipulating these objects. There are plans for the future to expand this capability. In addition, *GeoMeter* is capable of representing superquadrics and generalized cylinders.

2.2 Analytic section

A major section of *GeoMeter* is devoted to the manipulation of polynomials and transcendental functions. The motivation for these functions is twofold. They permit the exact description of curved surfaces. They also provide the mechanism for performing algebraic deduction, which is useful in reasoning about geometric relations. *GeoMeter* provides polynomial arithmetic and various ordering and testing predicates for polynomials. A full set of utilities for printing and evaluating polynomials and transcendental functions is also available.

GeoMeter allows several specialized operations on polynomials. Functions for performing polynomial arithmetic, GCD, univariate factoring, remainder sequences, decomposition, resultant computation, and Gröbner Basis [7] computation are incorporated into *GeoMeter*. Functions are available for bounding the roots of univariate polynomials and limited capabilities exist for performing Cylindrical Algebraic Decomposition [2]. Methods for triangulating sets of polynomials are available, as well as functions for testing the consistency of a polynomial with a triangulated set. Polynomials are represented in distributed form. Every polynomial is a list of monomials. In turn, each monomial is a cons pair consisting of a coefficient and a *term* representing a power product of the variables of the polynomial.

[1] Note that this differs slightly from the usual definition.

Transcendental functions are represented as a pair (p, s) where p is an arbitrary polynomial, and s is a set of substitutions mapping the variables of p to functions. For instance, a circular arc in *GeoMeter* is represented by a pair of transcendental functions:

$$p_1(x) = rx + x_0, \quad x \to \cos\theta$$

$$p_2(y) = ry + y_0, \quad y \to \sin\theta$$

where x_0, y_0 is the center of the arc and r is its radius.

3 Applications

3.1 Representing curves and surfaces

As mentioned above, *GeoMeter* has the capability to define parametric surfaces and curves. Both are defined using transcendental or polynomial functions in one or two variables over some interval. Parametric curves are defined using three transcendental functions in one variable, u: $x(u), y(u), z(u)$. The curve structure also has a slot for the bounds on u and the sampling rate for displaying the curve. Parametric faces are defined similarly, but the defining functions and interval are bivariate.

GeoMeter also supports algebraic curves and faces. Algebraic Faces are implicit surfaces defined by a polynomial $p(x, y, z) = 0$. In conjunction with algebraic faces, *GeoMeter* can also represent planar algebraic curves expressed with bivariate polynomials. Decomposition techniques [1,2] are used for display and analysis of such curves and surfaces. Figure 2 shows a sample *GeoMeter* frame displaying various objects.

3.2 Projection and Image Formation

There are a number of applications which require models of the imaging process. Modeling the projection process is not only useful for display of objects. It has been used (via *GeoMeter*) to model appearances for robot navigation [19,18]. The projection process within *GeoMeter* is central projection, also known as perspective projection. The projection is modeled in *GeoMeter* by a Camera entity, which contains the projection parameters. The projection is computed on points in $\mathbf{R}^3$, which are represented by homogeneous coordinates in $\mathbf{R}^4$. Each point is rotated and translated by a homogeneous 4x4 matrix that represents the transform from model coordinates to camera coordinates. Then each point is projected onto the image plane according to the camera parameters: the camera lens focal length, zoom, aspect ratio, and the image center.

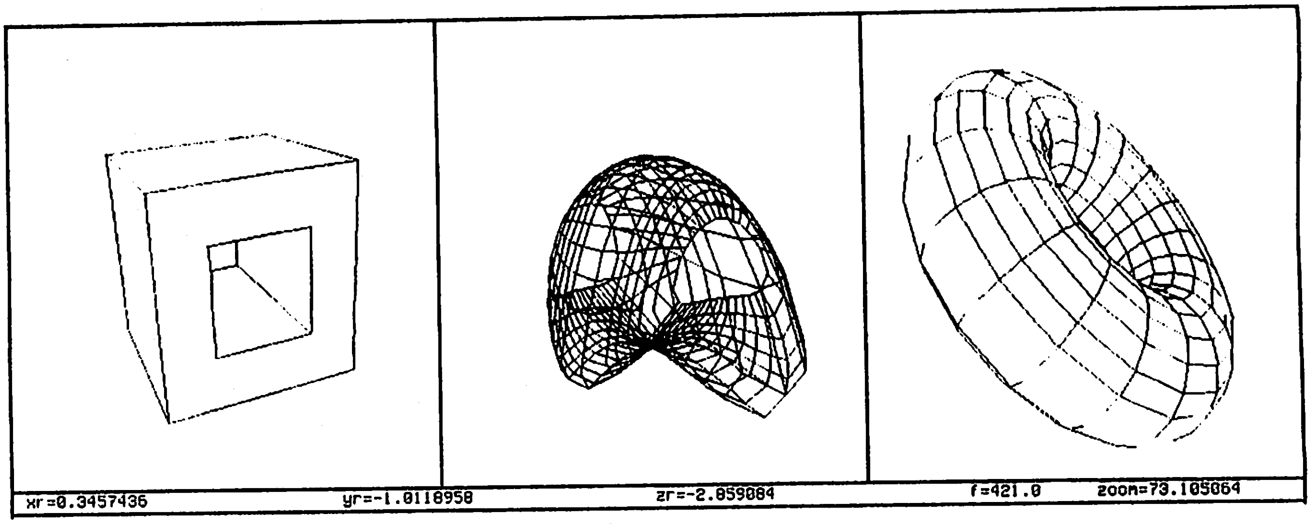

Figure 2: A Planar, algebraic, and parametric surface, respectively

In implementing the projection operation, we are not only interested in point sets, but also in the edges and possibly the faces that constitute the model as well as their visibility. *GeoMeter* contains a Viewer entity, which stores information about what is to be projected and the camera entity parameters for performing the projection. In addition, the Viewer contains the global, local, and projected coordinates of the points of the model, along with incidence information that allows faces and edges to be selected and drawn. The global coordinates and the incidence information are obtained from the faces, edges, and vertices of the model (or part thereof) that is being projected.

3.3 Model construction

GeoMeter has been used for experiments in model construction at GE. Stenstrom, et al. [12] describe a method for constructing volume models from image data. The technique involves the formation of volume sets from different views and performing boolean intersection of the sets obtained. Most of the implementation of this technique is carried out in *GeoMeter*. Other techniques have been developed [3] which use algebraic constraints to construct "parameterized models". Extensions of this work are reported later in section 4.3.2, and elsewhere in these proceedings [28].

3.4 Robot Navigation

GeoMeter is also being used for robot navigation experiments at the University of Massachusetts [18,19]. In order to meet the demands of a robot navigation system, *GeoMeter* had to satisfy several requirements. Geometric modeling must be easily interfaced with other modules needed for specific tasks in navigation. In our work it is interfaced directly with the high-level model matching functions used during model-to-image and image-to-model matching [4]. Components of the models are annotated with visual characteristics. A multilevel representation scheme is required so that parts of objects can be isolated and named as landmarks for recognition.

For navigation through a complex domain, one needs to model the world in which objects can be located. In our environment, buildings, lamp posts, and telephone poles must be modeled. Buildings have sub-objects such as windows and doors. Elements at all levels may be annotated with information relevant to visual tasks. In order to navigate from known landmarks, the environment must be modeled accurately. For the University of Massachusetts campus, this was done using information obtained from a careful survey of the environment, building plans, and direct measurements. Once the landmarks are identified, *GeoMeter* is used for pose refinement [26] to obtain robot bearings from visual information and information provided by the campus model.

3.5 Matching

Both at GE and at the University of Massachusetts, experiments in object recognition are underway [8,33]. These experiments use *GeoMeter* for some of the geometric operations and data structures required for correspondence and the computation of transformations. In both schemes, models are created, stored and displayed using *GeoMeter*. Images are processed to obtain edge information which is then compared to the model data to identify objects and their poses. In work described elsewhere in these proceedings, a method has also been devised for selecting and verifying the best matches out of a finite set of possibilities [22].

4 Solution Techniques

Solution of nonlinear systems of equations and optimization are two functions which are central to some of the aforementioned application areas. *GeoMeter* supports methods for solving such problems. Much of this machinery is oriented toward exact solution methods such as Wu's Method [34], the Gröbner Basis method [7,23] or algebraic decomposition [2,9]. There are also functions for computing approximate solutions.

4.1 Numerical methods

GeoMeter uses numerical methods for obtaining approximate solutions to nonlinear systems of equations. Functions exist for exact computation of the Jacobian, and for Newton's method for nonlinear systems. Morgan's Continuation method [27] has also been implemented in *GeoMeter*. This is a numerical method which can find all isolated point solutions to a system of nonlinear equations. It is a promising method that avoids many of the convergence problems to which Newton's method is susceptible. In addition, most of the computation used for modeling purposes in *GeoMeter* is numerical in nature (e.g., curve and surface intersection, boolean operations, transformation, etc.).

4.2 Support for Geometric Reasoning

GeoMeter supports two different but related approaches to reasoning in algebraic geometry: a refutational method based on the Gröbner basis algorithm [24] and a direct method by Wu based on the Ritt principle. [2] Both methods take polynomial equations as input. It is assumed that geometric relations have already been transformed into polynomial equations.

4.2.1 Refutational approach based on the Gröbner basis method

In the refutational method, the hypotheses of a geometry statement and the negation of the conjecture being proved are input and it is checked using the Gröbner basis algorithm that they are not satisfiable. If the algorithm detects that the Gröbner basis includes 1, it declares that the conjecture follows from the input. Otherwise, the Gröbner basis generated by *GeoMeter* can be used to extract out additional conditions that must be imposed on the input for the conjecture to follow from the input.

The refutational method has been shown to be complete for deciding whether a conjecture follows from the input or not [24]. In the case when the conjecture does not follow from the input, the method has also been shown to be complete for computing conditions under which the conjecture would follow from the input. The method has been successfully used to prove over a hundred geometry theorems including many nontrivial theorems which even humans find very difficult to prove. The method is fully described with examples and theoretical foundations in [24].

[2] In fact, this portion of the software used to be called GEOMETER [14] and the whole system used to be called GEOCALC until we discovered that there was a commerical product with the name GEOCALC. It was then decided to call the whole system *GeoMeter*.

4.2.2 The direct approach based on Wu's method

GeoMeter also supports Wu's method for geometric reasoning [34,36]. In contrast to the refutational approach based on the Gröbner basis algorithm, the method is direct. The hypotheses of a geometry statement are transformed into a triangular form using the Wu-Ritt method. *GeoMeter* expects the user to specify the independent variables as well as a total order on dependent variables; it currently does not provide any assistance in selecting dependent variables. Independent variables correspond to the degree of freedom in a geometric configuration defined by a geometry statement. Intuitively, independent variables are those variables which can be assigned arbitrary values and which determine the values of dependent variables.

Once a triangular system of polynomials is computed, the polynomial corresponding to the conjecture is pseudo-divided by each of the polynomial equations in the triangular form to successively eliminate each dependent variable in the conjecture. If the remainder is 0, then conjecture follows from the hypotheses. In this case, the method also identifies subsidiary conditions ruling out degenerate cases for the conjecture to follow from the hypotheses.

If the remainder is not 0, then it is still possible that the conjecture follows from the hypotheses. The triangular form of the hypotheses must be checked for irreducibility. If polynomials in the triangular form cannot be factored over successive extension fields, then the triangular form is irreducible. If the remainder of a conjecture with respect to an irreducible triangular form obtained from the hypotheses is not 0, then the conjecture does not follow from the hypotheses. Otherwise, the polynomials in the triangular form must be factored generating a set of irreducible triangular forms; the conjecture must then be checked over each of these irreducible triangular forms. *GeoMeter* does not provide algorithms for checking the irreducibility of a triangular form, nor does it provide any algorithms for factoring over extension fields. Theoretical foundations of Wu's method are discussed in [35]. An excellent implementation of Wu's method and its success in proving nontrivial geometry theorems are discussed in [10,11]. An informal discussion of Wu's method and its application to problems in perspective viewing is described in [25].

4.3 Hybrid Approaches

4.3.1 Hybrid Solution Methods

The power of purely exact methods for geometric reasoning and representation is limited to relatively small problems (see [9] for an analysis). By contrast, large model specifications consisting of thousands of entities can be successfully processed quickly if numerical methods are used. While they are capable of fast solution of such systems, numerical methods can be plagued with error accumulation. More importantly, Newton's method is only guaranteed to converge under very strict conditions [29].

Using tools in *GeoMeter*, an approach is being pursued where exact methods are used to improve the convergence properties of numerical methods. The basic idea is to determine a set of independent model parameters which can be freely varied with respect to the model constraints. Exact methods are then used to triangulate the constraint equations, as in Wu's method described earlier. The triangulated constraints can then be easily differentiated to determine the singularities of the Jacobian matrix. Thus, algebraic techniques can be used to implement restrained versions of Newton's method which only operate in "safe" regions where gradients are always uniquely defined. Experiments with this basic technique have been successful on systems with up to eight parameters. Other experiments are underway to examine the possibility of reduction and decomposition of the original system. This involves decomposing the system into individual subproblems which can be solved more easily than the original problem.

4.3.2 Constraint-based modeling

One approach to constraint-based modeling is to represent geometric constraints in a relational database, along with a numerical specification [13]. Known relationships can be retrieved using standard database techniques. Additional relationships can be derived by computation on the numerical specification of the objects and object relationships, or by logical inferences on the known relations. The power of automated logical inference techniques is limited to relatively simple deductions. On the other hand, the inference of relationships by numerical processes alone has limited robustness, particularly in the case of empirical data with significant errors. The numerically derived relationships can be easily inconsistent with logically derived relations. SRI's CKS (Core Knowledge System) deals with this uncertainty by providing a logic of belief which can handle multiple agents with various levels of reliability.

Another approach is to maintain a consistent set of geometric relationships that are maintained in a relational network, but specified algebraically [31]. The algebraic equations and inequalities provide a parametric specification of the objects and object relationships. The interaction with empirical data is taken as a problem in error minimization. That is, the parameters of the model specification are to be adjusted such that the distance between model predictions and actual image features, and other empirical data, is a minimum. The resulting model configuration maintains the consistency of *a priori* constraints while accommodating empirical relationships as closely as possible. We refer to this approach as constraint-based modeling [28].

The next major development of the constraint-based modeling technique in *GeoMeter* will be the integration of the algebraically derived convergence strategy with classical nonlinear programming methods.

5 Conclusion

GeoMeter is a versatile tool for solid modeling and algebraic manipulation. It is written in Common Lisp, and is portable. The source code is available via anonymous FTP from Internet host VAX1.CS.UMASS.EDU (128.119.40.1). To obtain *GeoMeter*, contact GEOMETER@CS.UMASS.EDU via electronic mail.

The authors would like to acknowledge the many people who have contributed to *GeoMeter*, especially Michele Barry, Brian Burns, Dave Cyrluk, Antonio Delgado, Rick Harris, Aaron Heller, Tim Kelliher, Ross Stenstrom, Dan Thompson, and Patricia Vrobel.

References

[1] Dennis S. Arnon. Topologically reliable display of algebraic curves. *Computer Graphics*, 17(3):219–227, July 1983.

[2] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition. *SIAM Journal of Computing*, 13:865–877,878–889, 1984.

[3] Michele Barry, David Cyrluk, Deepak Kapur, Joseph Mundy, and Van-Duc Nguyen. A multi-level geometric reasoning system for vision. *Artificial Intelligence*, 37:291–332, 1988.

[4] J. Ross Beveridge, R. Weiss, and E. Riseman. Matching and fitting sets of 2d line segments to broken and skewed data. Technical Report in preparation, COINS department, University of Massachusetts at Amherst, 1989.

[5] Rodney A. Brooks. *Symbolic Reasoning Among 3-D Models and 2-D Images*. PhD thesis, Stanford University, June 1981.

[6] C. Brown. PADL2: A Technical Summary. *Proc. IEEE Computer Graphics Applications*, 2(2):69–84, March 1982.

[7] Bruno Buchberger. *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Polynomial Ideal*. PhD thesis, Universität Innsbruck, Innsbruck, Austria, 1965. in German.

[8] J. Brian Burns and Leslie J. Kitchen. Rapid object recognition from a large model base using prediction hierarchies. In *Proceedings: Image Understanding Workshop*, pages 711–719. DARPA, Morgan-Kaufman, Inc., 1988.

[9] John Francis Canny. *The Complexity of Robot Motion Planning.* PhD thesis, Massachusetts Institute of Technology, 1987.

[10] S.-C. Chou. Proving elementary geometry theorems using Wu's algorithm. In W. W. Bledsoe and D. W. Loveland, editors, *Contemporary Mathematics*, volume 29, pages 235–241. 1984.

[11] S.-C. Chou. *Proving and discovering theorems in elementary geometry using Wu's method.* PhD thesis, University of Texas, Austin, TX, 1985.

[12] C. I. Connolly and J. R. Stenstrom. 3D scene reconstruction from multiple sensory images. In *Proceedings of the IEEE Computer Society Workshop on Interpretation of 3D Scenes.* IEEE Computer Society, November 1989. to appear.

[13] N.R. Corby, J.L. Mundy, P.A. Vrobel, A.J. Hanson, L.H. Quam, G.B. Smith, and T.M. Strat. PACE: An environment for intelligence analysis. In *Proceedings: Image Understanding Workshop.* DARPA, Morgan-Kaufman, Inc., 1988.

[14] D. Cyrluk, R. Harris, and D. Kapur. GEOMETER: A theorem prover for algebraic geometry. In *Proceedings of the 9th International Conference on Automated Deduction (CADE-9)*, Argonne, IL, May 1988.

[15] David A. Cyrluk, Deepak Kapur, and Joseph L. Mundy. Algebraic reasoning in view consistency and parameterized model matching problems. In *Proceedings: Image Understanding Workshop*, pages 731–739. DARPA, Morgan-Kaufman, Inc., April 1988.

[16] S. Eilenberg and N. Steenrod. *Foundations of Algebraic Topology.* Princeton University Press, Princeton, NJ, 1952.

[17] C. Connolly et. al. *GeoMeter: Solid Modelling and Algebraic Manipulation.* COINS Department, University of Massachusetts at Amherst, 1989.

[18] Claude Fennema, Allen Hanson, and Edward Riseman. Towards autonomous mobile robot navigation. In *Proceedings: Image Understanding Workshop.* DARPA, Morgan-Kaufman, Inc., 1989.

[19] Claude Fennema, Edward Riseman, and Allen Hanson. Planning with perceptual milestones to control uncertainty in robot navigation. In *Proc. of SPIE*, Cambridge, MA, November 1988. International Society for Photographic and Industrial Engineering.

[20] Peter Giblin and Richard Weiss. Reconstruction of surfaces from profiles. In *Proceedings: Image Understanding Workshop*, pages 900–908. DARPA, Morgan-Kaufman, Inc., February 1987.

[21] M. Greenberg and J. Harper. *Algebraic Topology - A First Course.* Benjamin, Reading, MA, 1981.

[22] A. J. Heller and J. R. Stenstrom. Verification of recognition and alignment hypotheses by means of edge verification statistics. In *Proceedings: Image Understanding Workshop.* DARPA, Morgan-Kaufman, Inc., 1989.

[23] A. Kandri-Rody and D. Kapur. Algorithms for computing the Gröbner bases of polynomial ideals over various Euclidean rings. In *Lecture Notes in Computer Science*, volume 174. Springer-Verlag, New York, NY, 1984.

[24] D. Kapur. A refutational approach to geometry theorem proving. *Artificial Intelligence*, 37:61–93, 1988.

[25] Deepak Kapur and Joseph L. Mundy. Wu's method and its application to perspective viewing. *Artificial Intelligence*, 37:15–36, 1988.

[26] Rakesh Kumar. Determination of camera location and orientation. In *Proceedings: Image Understanding Workshop.* DARPA, Morgan-Kaufman, Inc., 1989. also COINS Technical Report (in preparation), University of Massachusetts at Amherst.

[27] Alexander Morgan. *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[28] J. L. Mundy, P. A. Vrobel, and R. E. Joynson. Constraint-based modeling. In *Proceedings: Image Understanding Workshop.* DARPA, Morgan-Kaufman, Inc., 1989.

[29] J. M. Ortega. *Numerical analysis - A second course.* Academic Press, New York, NY, 1972.

[30] Alex Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[31] Robin Popplestone. The Edinburgh Designer System as a Framework for Robotics: The Design of Behavior. *AI EDAM*, 1(1), 1988.

[32] A. Requicha and H. B. Voelker. Solid modeling: A historical summary and contemporary assessment. *IEEE Computer Graphics and Applications*, 2(2):9–24, March 1982.

[33] Daniel W. Thompson and Joseph L. Mundy. Three dimensional model matching from an unconstrained viewpoint. Computer Science Branch Internal Report, 1986.

[34] Wu Wen-tsün. On decision problem and mechanization of theorem proving in elementary geometry. *Scientia Sinica*, 21:159--172, 1978. also in Bledsoe and Loveland, eds., *Theorem Proving: After 25 Years, Contemporary Mathematics.*

[35] Wu Wen-tsün. Basic principles of mechanical theorem proving in geometries. *J. Syst. Sci. Math. Sci.*, 4(3):207–235, 1984.

[36] Wu Wen-tsün. Some recent advances in mechanical theorem proving of geometries. In W. W. Bledsoe and D. W. Loveland, editors, *Contemporary Mathematics*, volume 29, pages 235–241. 1984.