ON SOLVING VARIATIONAL GEOMETRY BY AUXILIARY CONSTRUCTIONS§

H. Lipson[1]*, M. Shpitalni[1], F. Kimura[2]

[1] Laboratory for Computer Graphics & CAD, Faculty of Mechanical Engineering
Technion-Israel Institute of Technology, Haifa, Israel 32000

[2] Maintenance Laboratory, Department of Precision Engineering
University of Tokyo, Japan

*Conference Lecturer

ABSTRACT

This paper discusses an approach for converting variational geometry constraints into parametric geometry, by means of creating auxiliary constraints and finding a sequential construction. The solution strategy is based on creation of new constraint chains yielding new paths in the constraint graph. Use of auxiliary constraints can thus decompose strongly connected components in the original constraint graph, and assist traditional constraint solution techniques.

INTRODUCTION

Constraint based geometric design refers to specification of shape by means of geometrical constraints relating shape *features* to shape *parameters*. Constraint specification is a basic computer aided design tool. It permits both flexibility in the geometric specification and in the conveyance of design intent. The constraints are typically specified on a basis of a predefined *topological* arrangement of the features, which provides a context for the problem. In a typical 'simplified' conceptual design process, the designer starts with a sketched concept, then constrains the concept according to engineering specification. The computer then combines the concept and the constraints to form an accurate instance of the geometry, suitable for detailed design and production.

There is an extensive amount of literature on constraint solution approaches and applications from various fields of interest, such as geometry, kinematics, engineering constraints, theorem proving, and programming languages. An extensive survey of general constraint solution approaches can be found in [1] and [2]. Although there are various types of engineering constraints, many higher-level engineering constraints such as manufacturability, functionality and even aesthetic constraints, can often be *reduced* to geometric constraints. Geometric constraints are therefore a *fundamental* form of constraint; learning how to reason and solve geometric constraint sets is hence of elementary importance. Reviews of constraint solution methods in the *geometric* context can be found in [3-6]. We focus this paper on a new insight into relationships among geometric constraints.

PROBLEM STATEMENT

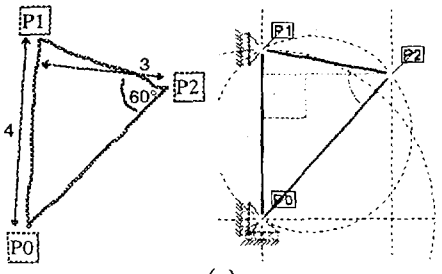
When solving geometrical constraints, the solver must provide an instance of the given topology which satisfies the given constraints *exactly*. Figure 1(a) shows a simple two-dimensional geometrical constraint problem, consisting of a triangle specified by two lengths and an apex angle. The problem consist of three components: (a) the triangle sketch denoting the *concept* of the design or the *topology* of the corresponding mathematical problem, (b) the dimensions (two lengths and an angle) which are the dimensioning *scheme*, (c) and the dimension *parameters*, 3, 4 and 60°. Geometric constraint problems can be categorized depending on whether these components are given or unknown, and whether they are fixed or variable. In the most simple case, all four components are given, and all but the parameters are fixed. This case has been categorized as the *parametric geometry* case, and is supported by most commercial CAD system. Parametric geometry frees the user from having to commit to any specific values of the parameters, and offers a stable and controllable solution method. However, in many cases, the user cannot commit to a particular dimensioning scheme in advance.

Variational geometry refers to the case of constructing geometry from scratch, rather than just varying existing geometry according to a fixed dimensioning scheme. A typical variational geometry solver therefore confronts problems of deriving a solution sequence, of handling large sets of non-linear simultaneous equations, of managing multiple solutions and of determining the most plausible solution as the one intended by the user. Hence it is generally considered a difficult task, especially as it is required to function reliably on complex structures in interactive near-real-time environments.

This paper proposes an approach for solving variational geometry problems based on determining a sequential construction sequence using *auxiliary constraints*. Once a construction sequence is obtained, the problem can be considered parametric, and resolved for different parameters by re-executing the construction sequence.

---

§ Excerpts from H. Lipson, F. Kimura, M. Shpitalni, 1998, "Converting Variational Geometry to Parametric Geometry by Auxiliary Constructions", submitted to the *Journal of Computer Aided Design*



(a)

Given:
C0:P0.X=6.0 C3:D(P2,(P0;P1))=-3.00
C1:P0.Y=8.0 C4:D(P0,P1)=4.00
C2:P1.X=6.0 C5:A(P1,P2,P0)=60°
Solving:
Using ground-x constraint C0:
Locus P0L0: Point P0 is on line 1.00x+0.00y+-6.00=0
Using ground-y constraint C1:
Locus P0L1: Point P0 is on line 0.00x+1.00y+-8.00=0
Combining locus P0L0 and P0L1, point P0 is at 6.00;8.00
Using ground-x constraint C2:
Locus P1L0: Point P1 is on line 1.00x+0.00y+-6.00=0
Using point-point distance constraint C4 and point P0:
Locus P1L1: Point P1 is on circle at 6.00;8.00 with R=4.00
Combining locus P1L0 and P1L1, point P1 is at 6.00;12.00 or 6.00;4.00 (Selected 1st)
Using angle constraint C5 and pointsP1, P0:
Locus P2L0: Point P2 is on circle at 7.15;10.00 with R=2.31
Using point-line distance constraint C3 and points P0, P1:
Locus P2L1: Point P2 is on line 1.00x+0.00y+-9.00=0
Combining locus P2L0 and P2L1, point P2 is at 9.00;11.39 or 9.00;8.61 (Selected 1st)

(b)

Fig 1. *(a) Simple constructive variational geometry problem and solution, (b) Transcript of automated solution process.*

SOLVING GEOMETRIC CONSTRAINTS

There are several approaches to solving geometric constraints, as illustrated in Fig 3. In general, solution techniques can be categorized into algebraic solvers, constructive solvers and non-deterministic solvers. In practice, most solvers are hybrid. A review of these techniques can be found in [7]. We describe them here briefly.

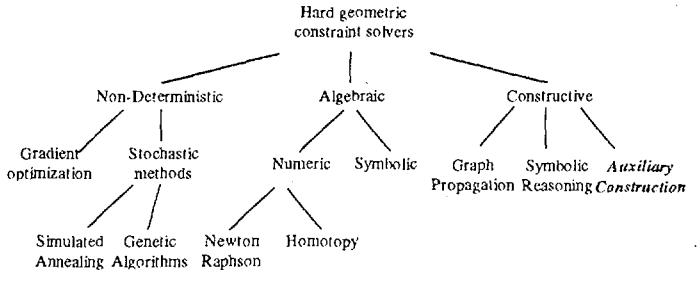


Fig 2. *A taxonomy of geometric constraint solvers.*

According to the *algebraic* approach, the geometrical constraints are translated into algebraic expressions, which must be solved simultaneously to yield the solution. The set of equations can be solved using iterative *numeric* methods, such as Newton-Raphson or Homotopy, as well as *symbolic* algebra methods. *Constructive* constraint solvers satisfy constraints using a sequence of *elementary constructive steps*, similar to building a shape by means of a compass and protractor. For example, the transcript of the automated solution sequence for the problem shown in Figure 1(a) is provided in Figure 1(b). The constructive method is based on the logic inference approach of artificial intelligence. See, for example, the systems described in [8-10]. A third class of solvers incorporates non-deterministic techniques, such as stochastic reasoning, genetic algorithms and simulated annealing. A qualitative comparison of some prevailing factors of the two main approaches is provided in Table 1.

Factor	*Algebraic*	*Constructive*
Stability and convergence of the solution process	-	+
Generality of supported constraint types	+	-
Independence of initial guess (sketch)	-	+
Traceability of solution sequence	-	+
Explorability for multiple solutions	-	+
Unnecessary to determine a solution strategy	+	-
Provision of Information upon failure/conflict	-	+
Well established techniques in the literature	+	-
Ability to use physical domain information	-	+

Table 1. *A qualitative comparison of the algebraic and constructive methods by some differentiating positive qualities*

AUXILIARY CONSTRUCTIONS

The method of auxiliary constructions is a *constraint rewriting rule*. Constructive constraint solvers use constraint rewriting rules for generating new constraints based on the given constraints, in order to facilitate finding a construction sequence in the constraint graph. For example, two common rewriting rules are (a) *trigonometric rules* such as the Law of cosines and the Law of Sines, used to derive new constraints based on known angles, distances or relationships among them within triangles; Alternatively, (b) *accumulation rules* derive new constraints by simple arithmetic on existing constraints.

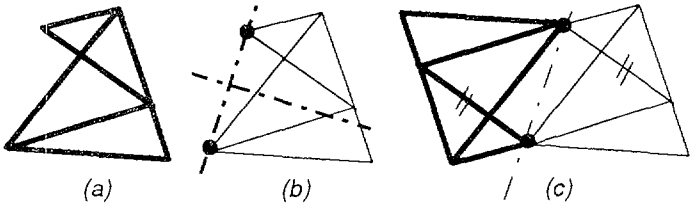
The generation of auxiliary constructions is based on generation of *new geometry* as well as new constraints, both based on existing constraints and geometry. The new geometry and new constraints are generated by duplicating and applying a *transformation* on the existing geometry (or part of it) and on the relative constraints in the corresponding part of the constraint graph. Additional constraints are then added into the constraint graph depending on the kind of transformation applied, linking the new (transformed) part to the old part. In this paper we discuss the *reflection* transformation. Other transformations such as translation or rotation can also be used with the same procedures.

Given an arbitrary geometrical construction of dimension d, a sequence of d reflections about perpendicular axes, as shown in Figure 3, has the following properties:

1. Relative constraints within the original geometry hold

among the corresponding entities of the reflected geometry as well, because the transformation is *linear and non scaling*/skewing.

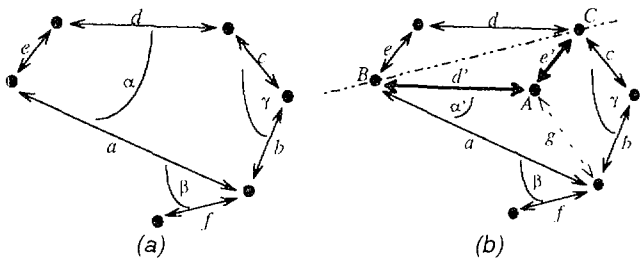
2. Absolute angles and lengths are maintained, i.e. corresponding entities are parallel and equally sized, because the transformation is *linear and non-orienting*.
3. If the reflection axes pass through some points of the original geometry (Figure 3b), the new and old geometry will be linked at these points, and therefore the constraint graph will be linked too.



(a) (b) (c)

Fig 3 *(a) The original geometry, (b) two perpendicular reflection axes and two link points, and (c) the reflected geometry, linked to the original geometry. Note that in 2D this transformation is equivalent to a rotation of 180°.*

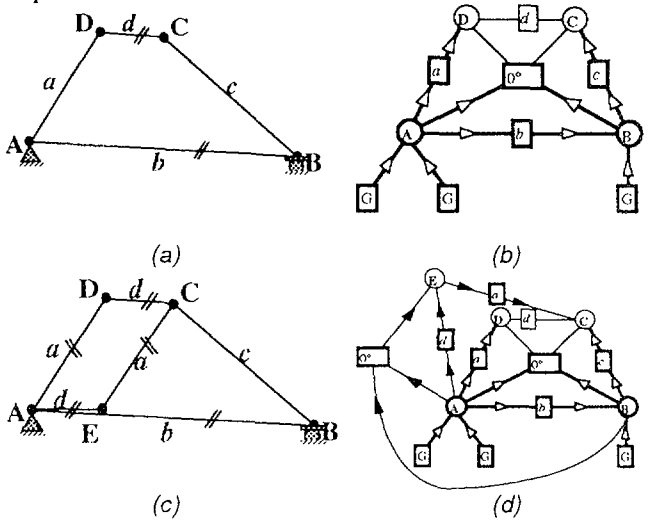
Due to the properties listed above, performing a reflection on the complete geometry or on part of it will create new constraints and paths in the constraint graph. To demonstrate how this reflection is used as a 'building block' to create an entirely sequential constructive solution, examine the following problem. An arbitrary set of n points well defined by distances and angles among arbitrary pairs of points is given, such as the set shown in Figure 4(a). In this geometry, consider any set of three constraints. If the three constraints pertain to three common points, (e.g. the set $\{c, b, \gamma\}$), then a trigonometric rule can be used to resolve required information to construct that local geometry. If the set does not pertain to three common points (e.g. set $\{a, d, \alpha\}$), we chose a reflection that makes two constraints adjacent, and thus reduces the total number of common points in the set (e.g. the reflection along the dashed line in Figure 4b). Such a reflection always passes through two points in the original set. The newly created geometry is now one step closer top being solvable by, say, a trigonometric or accumulation rule. In the example, the newly created set $\{a, d', \alpha'\}$ can be solved to yield g and A, and now there is one additional *usable* constraint for point C.



(a) (b)

Fig 4. *(a) The original geometry, (b) new point and constraints generated by reflection.*

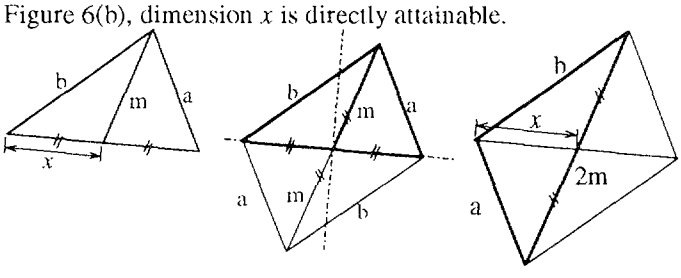
To see the effect of such steps on the constraint graph, consider the example shown in Figure 5. This is a simple problem, which can be solved using only one reflection. The example consists of a trapezoid constrained by four side lengths, a, b, c and d. Starting from point A its is easy to determine point B. But in order to determine either point C or D, more than one constraint must be applied simultaneously. Hence a solution sequence propagates to the state highlighted in Figure 5(b), when a further sequential (or *parametric*) construction cannot be obtained. However, reflecting on axis A-C to create an auxiliary construction as shown in Figure 5(c). This has created a new path in the constraint graph, Figure 5(d), permitting a sequential solution.



(a) (b) (c) (d)

Fig 5 *(a) A variational problem which cannot be solved using a sequential procedure, (b) the associated constraint graph, with the solution propagated (bright arrows) to a stagnation state, (c) two constraints (lengths* d *and* a*) swapped to create an auxiliary construction with new vertex* E*, and (d) the new graph now permits a sequential solution (dark arrows).*

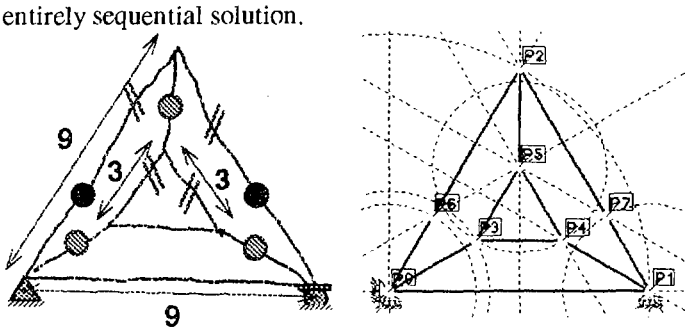
More complex conditions can be manipulated by reflecting about axes which do not pass only through two points. For example, Figure 6(a) shows a variational problem requiring the construction of a triangle given lengths of two sides and the length of a median. The key dimension sought is that of the base, labeled x. By reflecting the geometry twice about the base, as shown in Figure 6(b), dimension x is directly attainable.



(a) (b) (c)

Fig 6 *(a) The original geometry, (b) two perpendicular reflection axes (c) the reflected geometry*

We also note that constraint transformation can be executed multiple times on a series or set of constraints, thereby effectively operating on complete *features*. Simiarly, Figure 7 shows a two-dimensional problem exhibiting triple symmetry, which requires application of two consecutive constraint transformations to reach an entirely sequential solution.



(a) (b)

Fig 7 (a) A constraint problem with triple symmetry, (b) The solution requiring two consecutive constraint transformations.

The same process can be applied in three dimensions. Figure 8(a) shows a sketch of a constrained pyramid. The sequence of solution is shown in the following images. The same principles used in solving two-dimensional problems has been applied here, with the following modifications (a) since the problem is three dimensional, the system now intersects planes and spheres instead of lines and circles, and (b) a reflection about three axes (rather than two) is performed. Note that the reflection can be performed on the sketched geometry as is, even before it is accurate and even while it is still two dimensional.

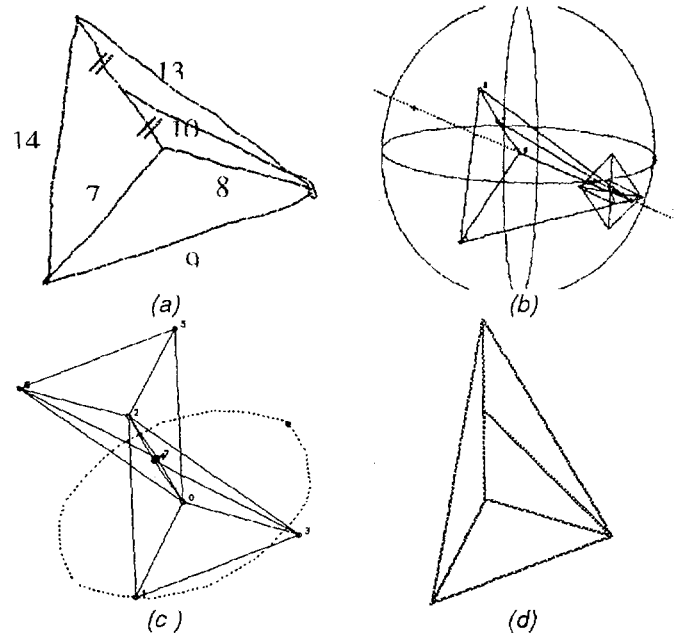


Fig 8 (a) The original sketch with specification, (b) the actual solver proceeds to solve the 3D geometry using planes and spheres, then, (c) performs a reflection, and obtains the final geometry. (d) The isolated solution

CONCLUSIONS AND FURTHER RESEARCH

This paper discussed an approach for converting variational geometry constraints into parametric geometry, by means of creating auxiliary constraints and finding a purely sequential construction. The solution strategy was based on decomposing the problem into a sequential process consisting of known elementary-cases, and on construction of auxiliary geometry by constraint transformation, resulting in creation of new constraint chains in the constraint graph. Further research is required towards determining when constraint reflection can and should be applied in search for a sequential construction, and which part of the geometry must be reflected. Similarly, it is necessary to determine where to place ground constraints, when none have been specified by the user. Both of these issues are resolvable by performing searches on the constraint graph, locating where additional paths and starting points are required for solution propagation.

ACKNOWLEDGMENT

This research has been supported in part by the Israeli Science Foundation (ISF) Grant No. 1380021 (Research Number 033-028).

REFERENCES

[1] Van Hentenryck P., Saraswat V. *et al*, 1996, "Strategic directions in constraint programming", *ACM computing surveys*, Vol. 28, No. 4

[2] Leler W., 1988, *Constraint programming languages*, Addison-Wesley

[3] Suzuki H., Kimura F., Sata T., 1985, "Treatment of dimensions on product modeling concept", *Design and Synthesis*, North Holland, pp. 491-496.

[4] Juster N. P., 1992, "Modeling and representation of dimensions and tolerances: a survay", *Computer Aided Design*, Vol. 24, No. 1, pp. 3-17.

[5] Nelson, G., 1985, "A constraint based graphical system", *ACM SIGGRAPH 85*, pp. 235-243.

[6] Kimura F., Suzuki H., Ando H., Sato T., Kinosada A., 1987, Variational geometry based on logical constraints and its applications to product modeling, Annals of the CIRP 36/1, pp. 75-78.

[7] Fudos I., 1995, "Constraint solving for computer Aided Design", Ph.D. Thesis, Purdue University

[8] Bouma W., Fudos I., Hoffmann C. M., Cai J., Paige R., 1995, "Geometric constraint solver", *Computer Aided Design*, Vol 27, No. 6, pp. 487-501.

[9] Kramer G., 1992, *Solving geometric constraint systems*, MIT Press

[10] Joan-Arinyo R., Soto A., 1996, "A ruler-and-compass geometric constraint solver", *5th IFIP Workshop on Geometric Modeling in CAD*, Airlie, VA, May 19-23.