Computer-Aided Design 67–68 (2015) 38–47

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

h-graphs: A new representation for tree decompositions of graphs[☆]

Marta Hidalgo [a], Robert Joan-Arinyo [b,*]

[a] Systems Genomics Lab, Centro de Investigación Príncipe Felipe, València, Spain
[b] Grup d'Informàtica a l'Enginyeria, Universitat Politècnica de Catalunya, Av Diagonal 647, 8è, Barcelona, Catalonia

HIGHLIGHTS

- h-graphs, a new representation for tree decompositions of constraints graph is presented.
- h-graphs explicitly capture construction steps dependencies in a tree decomposition.
- An application to speed up computing feasibility ranges for constraint parameters is described.

ARTICLE INFO

Article history:
Received 14 November 2014
Accepted 5 May 2015

Keywords:
Parametric solid modeling
Geometric constraint solving
Constraint graphs
Tree-decompositions
Construction steps dependencies
Parameter ranges

ABSTRACT

In geometric constraint solving, 2D well constrained geometric problems can be abstracted as Laman graphs. If the graph is tree decomposable, the constraint-based geometric problem can be solved by a Decomposition–Recombination planner based solver. In general decomposition and recombination steps can be completed only when steps on which they are dependent have already been completed. This fact naturally defines a hierarchy in the decomposition–recombination steps that traditional tree decomposition representations do not capture explicitly.

In this work we introduce h-graphs, a new representation for decompositions of tree decomposable Laman graphs, which captures dependence relations between different tree decomposition steps. We show how h-graphs help in efficiently computing parameter ranges for which solution instances to well constrained, tree decomposable geometric constraint problems with one degree of freedom can actually be constructed.

1. Introduction

Many applications in computer-aided design, computer-aided manufacturing, kinematics, robotics or dynamic geometry are conveniently modeled by geometric problems defined by geometric constraints with parameters, some of them representing dimensions. These generic models allow the user to easily generate specific instances for various parameter and constraint values.

When parametric models are used in real applications, it is often found that instantiation may fail for some parameter values. Assuming that failures are not due to bugs in the system, they should be attributed to a more basic problem, that is, a certain combination of constraints in the model and values of parameters do not define a valid shape.

The failure to instantiate the model poses naturally the question of how to compute ranges for parameters such that model instantiation is feasible. This problem or restricted versions of it have been addressed in the literature. Shapiro and Vossler, [1], and Raghothama and Shapiro, [2–4], developed a theory on validity of parametric family of solids by investigating the relationship between Brep and CSG schemas in systems with dual representations for solid modeling. The formulation is built on formalisms of algebraic topology. Unfortunately, it seems a rather difficult problem transforming these formalisms into effective algorithms.

Joan-Arinyo and Mata [5] reported a method to compute feasible ranges for parameters in geometric constraint solving under the assumption that values assigned to parameters are non-trivial-width intervals. The method applies to complex systems of geometric constraints in both 2D and 3D and has been successfully applied in the dynamic geometry field, [6]. It is a general method, the main drawback, however, is that it is based on numerical sampling.

Hoffmann and Kim [7] developed a constructive approach to calculate parameter ranges for systems of geometric constraints

☆ This paper has been recommended for acceptance by Christoph Hoffmann.
* Corresponding author.
E-mail addresses: mhidalgoga@gamil.com (M. Hidalgo), robert@cs.upc.edu (R. Joan-Arinyo).

http://dx.doi.org/10.1016/j.cad.2015.05.003
0010-4485/

that include sets of isothetic line segments and distance constraints between them. Model instantiation for distance parameters within the ranges output by the method preserve the topology of the set of isothetic lines.

Sitharam et al. in [8,9] reported recent theoretical results concerning the computation of intervals of realizable solutions to linkages, that is, 2D geometric constraint problems with one degree of freedom where the parameter is a distance value.

For the first time, van Der Meiden and Bronsvoort [10] described a method to directly figure out the allowable range for a single parameter in the problem, called *variant parameter*, such that an actual solution exists for any value in the range. The method was formalized by Hidalgo and Joan-Arinyo in [11,12] where a correctness proof along with specific implementation details were given. This approach heavily relies on identifying the set of construction steps in the solution to the constraint problem the actual execution of which depends on the current value assigned to the variant parameter. So far identifying the dependencies concerning the construction step under consideration required a specific computation, [10]. Hence devising a method able to efficiently identify beforehand the whole set of dependencies existing between the construction steps of a constraint problem would be a valuable accomplishment.

In this work we introduce h-graphs, a new representation for graph based constructive solutions to the geometric constraint problem. The h-graph captures the naturally occurring dependency relationships between construction steps in the solution of a geometric constraint problem. As well, computing parameter ranges where the solution is feasible using h-graphs improves over the method described by Hidalgo and Joan-Arinyo in [11,12].

In what follows we first give a short intuitive introduction to the geometric constraint solving problem to motivate the need for computing parameter ranges. Then we recall the graph tree decomposition, we give technical definitions related to dependence, define h-graphs, show some properties of h-graphs and, we describe an algorithm to compute h-graphs from a tree decomposition which is a solution to a geometric constraint problem. Finally we illustrate how actual dependencies are computed applying the h-graph to a geometric constraint problem with one degree of freedom.

2. Geometric constraint-based problems

Assume that we want to build a triangle the vertices of which are the points a, b, c like those shown in Fig. 1(a). We want point a to be placed at a distance d_1 from point b and point c to be placed at a distance d_2 from point b. Moreover we want that the edge bounded by points a, b makes an angle λ with respect to the edge bounded by points a, c. It is well known that this description properly defines a triangle in the Euclidean space. A ruler-and-compass procedure to build the triangle is illustrated in Fig. 1(b) and can be described as follows,

1. Draw an arbitrary straight line, say X.
2. On line X mark an arbitrary point a.
3. On line X mark a point b at a distance d_1 from a.
4. Draw a line L through point a and at an angle λ with line X.
5. Draw a circle C with center b and radius d_2.
6. Intersections of circle C and line L yield points c and c' that along with points a and b define triangles which fulfill the requirements described.

If the procedure is applied after assigning specific values to d_1, d_2 and λ, the geometric construction can be carried out depending on the specific assignment of values.

Many techniques have been reported in the literature that provide powerful and efficient methods for solving geometric

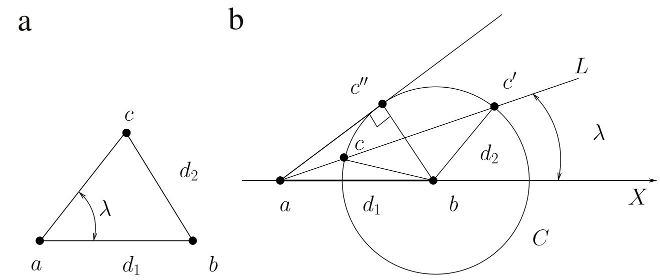

Fig. 1. (a) Triangle defined by three points, two point–point distances and an angle between two straight lines. (b) A ruler-and-compass constructive solution.

problems defined by constraints. For a review, see Hoffmann et al. [13]. Computer programs that solve geometric problems defined by constraints are called *solvers*. Among all the geometric constraint solving techniques, our interest here focuses on the one known as *constructive*. See [14–18] and the references there in for an in depth discussion on this topic.

Constructive solvers belong to the Decomposition–Recombination solvers, in short DR-solvers, class [15] and have two main components: the *analyzer* and the *constructor*. Given the geometric elements and the constraints defined on them, the analyzer figures out a description of how geometric elements are placed with respect to each other in such a way that the constraints are fulfilled. This description is called *construction plan*.

If the analyzer succeeds, actual values are assigned to the parameters and the constructor builds an instance of a placement for the geometric objects, provided that no numerical incompatibility arises due to geometric degeneracy.

In the example described above and illustrated in Fig. 1, the set of geometric elements includes the points $\{a, b, c\}$ while the constraints are the distances d_1, d_2 and the angle λ.

In this scenario asking for the set of values of λ for which the construction is actually feasible seems natural. To answer this question, efficiently computing dependencies between construction steps in the construction plan plays a central role.

In this work we capture a 2D geometric constraint problem as a graph $G = (V, E)$ where V is a finite set of nodes or vertices which stand for the geometric elements in the problem and E is a collection of edges. An edge is an unordered pair (u, v) of distinct vertices $u, v \in V(G)$. In general $V(G)$ and $E(G)$ will denote respectively the set of vertices and edges of the graph G.

In what follows we only consider 2D well constrained geometric constraint problems, that is, problems with a finite number of solution instances. In this work, these problems are abstracted as Laman graphs, [19], $G = (V, E)$ with $|V| \geq 3$ and such that

1. $|E| = 2|V| - 3$.
2. For every subgraph $G' = (V', E')$ with $V' \subset V$ and $E' \subset E$, $|E'| \leq 2|V'| - 3$.

3. Tree decomposition of a graph

Tree decompositions, also known as *triangular decompositions*, are a tool widely used in geometric constraint solving mainly when the underlying solving technique belongs to the DR solvers class. The resulting decomposition describes the solution to the geometric constraint problem by fixing how geometric elements are placed with respect to each other to fulfill the constraints. In this section we recall the concept of tree decomposition of a graph, we formalize the tree decomposition as a rewrite system and show some properties which will be used later on.