Another Paradigm for Geometric Constraints Solving

Dominique Michelucci Sebti Foufou* Loïc Lamarque
David Ménegaux

Lab. Le2i, UMR CNRS 5158, Université de Bourgogne
BP 47870, 21078 Dijon, France
{*dominique.michelucci, sebti.foufou, loic.lamarque, david.menegaux*}*@u-bourgogne.fr*

Abstract

Geometric constraints solving often relies on graph-based methods to decompose systems of geometric constraints. These methods have intrinsic and unavoidable limitations which are overcome by the witness method presented here.

1 Introduction

Geometric constraints solving tools [1, 3] support geometric modelers and other applications in CAD-CAM, chemistry, robotics and virtual reality. In a geometric constraints-based modeling tool, the designer has the ability to describe shapes by specifying a set of geometric constraints that the tool solves to provide the intended shape. Constraint specifications describe relations such as distance, angle, incidence, tangency, parallelism, and orthogonality between geometric entities such as points, lines, planes, conics, quadrics, or even higher degree algebraic curves and surfaces. Many problems in robotics (*e.g.* generalized Stewart platform), in molecular chemistry (*e.g.* finding the configurations of a molecule from inter-atomic distances; also known as the molecule problem), in geometric modeling (*e.g.* blending surfaces) can be formulated as systems of geometric constraints. Systems met in industry are composed of increasingly large sets of equations and unknowns. For example, geometric constraints become more and more used to define control points of parameterized patches in 3D; there are 48 unknown coordinates per bicubic patch, and the simplest shape requires a dozen of such patches. Thus decomposing such huge systems of geometric constraints into smaller ones is essential for finding a solution to those systems.

Earlier decomposition/solving methods were based on geometric rules and worked with 2D or 2.5D problems. Ideally, they produce a construction plan chaining simple geometric constructions (*e.g.* computing lines intersections, circumscribed circle to a triangle, etc). Graph-based methods are currently used to decompose a system of geometric constraints into a set of subsystems, find a solution for each subsystem, and merge these partial solutions. A large number of graph-based methods have been proposed [4, 9]. Most of these methods rely on a combinatorial count of degrees of freedom (DoF). They use graph flow computations, maximum matchings, k-connectedness properties.

One of the major drawbacks of graph-based methods is the fact that they often fail to detect subtle dependencies resulting from redundant or contradictory constraints in the system. The witness method, presented by Foufou et al. in [2] overcomes these limitations. The authors studied and extended the classical Numerical Probabilistic Method (NPM). The classical NPM, well known in rigidity theory to decide about the rigidity of graphs in any dimension, studies the structure of the Jacobian at a random (or generic) configuration. It only works for point-point distances constraints (*i.e.* the so called molecule problem) [8]. The extended NPM does not consider a random configuration, but a configuration similar to the unknown one. Similar means the configuration fulfills the same set of incidence constraints, such as collinearities and coplanarities. The extended NPM can be used to decompose more general geometric constraint systems into rigid subsystems.

This paper simplifies the NPM-based method for decomposing geometric constraint systems, and broadens its scope. Section 2 presents the principle of the witness with basic notions such as free infinitesimal motions and degrees of displacements. Section 3 shows how the rank is computed. Section 4 presents a first decomposition method which relies only on the witness. Section 5 illustrates the witness method on a 2D example, while section 6 concludes.

*Corresponding author. Currently guest researcher at the National Institute of Standards and Technology, Gaithersburg, MD, USA. sfoufou@cme.nist.gov

2 Witness principle

Let $F(U, X) = 0$ be a system of equations representing a set of geometric constraints. U is the vector of parameters composed of geometric (*e.g.* distances, angles) and/or non geometric values (*e.g.* weights, forces, costs). A witness (also called prototype, example, realization, structure, framework, etc.) is a couple (V, X_V) such that $F(V, X_V) = 0$ and $V \neq U$. In other words, a witness is a solution to a variant of the system to solve. The witness fulfils the incidence constraints of the problem; thus, it also fulfils the incidences due to geometric theorems (Pascal, Desargues, Pappus), and can be used to detect the dependencies between constraints into a geometric constraints system. The witness and the target (*i.e.* the unknown configuration we are looking for) have the same properties. Typically, the sketch interactively provided by the user is often a witness; the geometric relations are verified: the points are already aligned or coplanar as in the solution, the solver only has to correct angles and distances. Up to now in the CAD-CAM domain, we were always able to find a witness with rational coordinates (the counter-example in [2] is not relevant for CAD-CAM). For the molecule problem, points can be randomly chosen and the distances easily deduced; for the dodecahedron or the hexahedron problem, with planes of the faces randomly chosen the vertices and distances can be found.

In geometric constraints-based modeling, the constraints control the shape of the configuration, and then the only permitted actions are motions that one can apply to displace the modeled configuration. These motions are usually classified in two classes: the infinitesimal displacements, namely translations, rotations and their compositions, and the infinitesimal flexions, which deform the configuration. The essential idea of the proposed witness method is to compute the $\dot{X}$ vectors of the witness free infinitesimal motions $\epsilon \times \dot{X}$, such that the disturbed witness $X_V + \epsilon\dot{X}$ still fulfils the specified constraints: $F(V, X_V + \epsilon\dot{X}) = 0$. With a Taylor expansion: $F(V, X_V + \epsilon\dot{X}) = F(V, X_V) + \epsilon F'(V, X_V)\dot{X}^t + O(\epsilon^2)$. Thus $F'(V, X_V)\dot{X}^t$ must vanish: the free motions are given by the kernel of the witness jacobian matrix $F'(V, X_V)$.

It is always possible to compute a base, independent of the geometric constraints, for the free infinitesimal displacements in 2D. Table 1 shows an example of such base in the case of a point (x_i, y_i), a vector (u_k, v_k), and a line (a_l, b_l, c_l) of equation $a_l x + b_l y + c_l = 0$. This base contains three components: a translation t_x on the x axis, a translation t_y on the y axis, and a rotation r_{xy} around the origin. Dotted variables represent the values of the infinitesimal displacements, *e.g.* the values of $(\dot{x_i}, \dot{y_i})$, the infinitesimal translation t_x on the x axis of a point (x_i, y_i) are $(1, 0)$. Other geometric unknowns (barycentric coordinates, scalar products, areas...) and non geometric (costs, forces...) are unchanged by infinitesimal displacements, therefore the corresponding value in all vectors of the base is zero. The last line of table 1 describes an infinitesimal scaling s, which is not a displacement. If such vector s is orthogonal to the gradient vectors of all the equations, then the system is unchanged by scaling [7]. Sometimes, it may be interesting to reduce into well-constrained subsystems modulo the scaling transformations. Similarly, a base of six infinitesimal displacements can be defined in 3D, with three translations along x, y and z and three rotations in the planes Oxy, Oxz and Oyz.

	$\dot{x_i}$	$\dot{y_i}$	$\dot{a_l}$	$\dot{b_l}$	$\dot{c_l}$	$\dot{u_k}$	$\dot{v_k}$
t_y	1	0	0	0	$-a_l$	0	0
t_x	0	1	0	0	$-b_l$	0	0
r_{xy}	$-y_i$	x_i	$-b_l$	a_l	0	$-v_k$	u_k
s	x_i	y_i	$-a_l$	$-b_l$	0	u_k	v_k

Table 1: Example of base for free infinitesimal displacements in 2D.

For a given configuration (*i.e.* a figure) identified by a set of variables X, a subfigure is described by a subset of variables $Y \subset X$, each member of Y corresponds to a column in table 1. Let us denote by $D[Y]$ and $M[Y]$ the vectorial spaces of infinitesimal displacements and infinitesimal motions of Y respectively. For instance, for a line described in 2D by $Y = (a_L, b_L, c_L)$, the $D[Y]$ is given in table 2. The degree of displacements (the rank) of a figure Y is the number of independent infinitesimal displacements in $D[Y]$.

	$\dot{a}$	$\dot{b}$	$\dot{c}$
t_x	0	0	$-a_L$
t_y	0	0	$-b_L$
r_{xy}	$-b_L$	a_L	0

Table 2: The vectorial space of infinitesimal displacements of a line in 2D.

In the previous case, $D[Y]$ has rank 2, and the two translations t_x and t_y are clearly detected as dependent. Interrogating witness gives the degree of displacements of any figure, with no genericity condition required.

The witness also permits to achieve the following tests:

- A geometrically correct equation must be independent of the coordinate system. The equation: $(x_A - x_B)^2 + (x_A - x_B)^2 - D^2_{AB} = 0$ is correct whereas the equation $x_A = 0$ is not. An equation is independent of the coordinate system iff its gradient vector to the witness is orthogonal to all the displacement base vectors.

- The equations are independent to each other (neither redundant nor contradictory) iff their gradient vectors to the witness are independent.

- A figure, or a subfigure described by a subset Y of the variables, is well-constrained, or rigid, iff the vectorial space of its infinitesimal motions $M[Y]$ is identical to the vectorial space $D[Y]$ of its infinitesimal displacements, and the full degree of displacements (the full rank) of Y is 3 in 2D, and 6 in 3D.

Graph-based methods can only detect structural dependencies such as in the system $f(x, y, z) = g(z) = h(z) = 0$ where the unknown z is over-constrained. The interrogation of the witness enables the detection of more subtle dependencies (actually every dependencies). The constraints are dependent if the gradient vectors of the witness equations, *i.e.* the jacobian matrix of the witness, are dependent. The witness method can also detect if the witness (V, X_V) is generic or not. The details of this are left out for the sake of brevity.

3 Rank computations

The only calculations necessary to detect the dependencies between constraints, into a geometric constraints system, are those needed to obtain the ranks of a set of vectors for which all coordinates are known numbers. The ranks are computed by a Gauss triangulation or a LU decomposition. Due to numerical imprecision in computers arithmetic, the rank computation is rather awkward. For example, with floating-point arithmetic, vectors $(1, 1)$ and $(\sqrt{2}/2, \sqrt{2}/2)$ are not completely collinear. When the witness has rational coordinates, the Gauss triangulation and the LU decomposition are calculable exactly and quickly, *e.g.* calculations are carried out modulo a prime number close to one billion in a probabilistic way [5, 6].

We consider that a witness is rational if all its coordinates are rational numbers. Although some systems of geometric constraints, such as regular polygons in 2D and Platonician polytopes (icosahedron, dodecahedron) in 3D, have no rational witness, most of the CAD/CAM geometric constraint systems have rational witnesses. The advantage of rational witnesses is that it is possible to use exact arithmetics to compute ranks of vectors. CAD practioners prefer the SVD and the epsilon heuristic to deal with floating point arithmetics difficulties.

4 Witness-based decomposition

We introduce in this section a new term: an anchor is a rigid subset Y of three variables in 2D (respectively six in 3D) with full degrees of displacements. For instance, in 2D if the system directly or indirectly sets the distance between points A and B, then $\{x_A, y_A, x_B\}$ is an anchor. A system has a polynomial number of anchors. Each anchor A is contained in a unique maximal rigid set $Y = A \cup \{x | \{x\} \cup A \ is \ rigid\}$. We can then introduce the following decomposition method: if the system is flexible, the maximal rigid parts are determined from all the anchors. If the system is rigid, it becomes flexible by removing each constraint; then find the maximal rigid parts (cf. Figure 1).

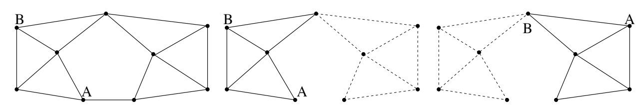

Figure 1: A 2D rigid system of constraints. Removing a constraint creates a flexible system with two maximal rigid parts. In the three subfigures, the maximal rigid part containing A and B is in thick lines.

5 Example: a 2D dependent system, and a probabilistic proof

Let us consider three points A, B, and C with the following constraints: the distance OA is specified by a parameter u. O is the middle of the points A and B, and distance OC and OA are equal. AC and BC are orthogonal; this last constraint results from the others. This result into the system of equations:

$$\begin{cases} (x_A - x_O)^2 + (y_A - y_O)^2 - u^2 = 0 \\ 2x_O - x_A - x_B = 0 \\ 2y_O - y_A - y_B = 0 \\ (x_C - x_O)^2 + (y_C - y_O)^2 \\ \qquad\qquad -(x_A - x_O)^2 - (y_A - y_O)^2 = 0 \\ (x_C - x_A)(x_C - x_B) + (y_C - y_A)(y_C - y_B) = 0 \end{cases}$$

A possible witness for this system of constraints is: $O = (0, 0), A = (-10, 0), B = (10, 0), C = (6, 8), u = 10$. Table 3 shows a possible base for the infinitesimal motions (the jacobian kernel). This base has rank 4, three displacements and a flexion: C can turn around O. Therefore the figure is flexible. Consulting the witness also shows that $\{O, A, B\}$ is rigid, as for $\{O, C\}$, and that the last equation results from the others. Suppose it is conjectured that $C(X) = 0$ is a consequence of the system $F(X) = 0$. First check that the conjecture indeed holds in the witness, i.e. that $C(X_V) = 0$. If not, the conjecture is clearly wrong. If $C(X_V) = 0$, then

	$\dot{x_O}$	$\dot{y_O}$	$\dot{x_A}$	$\dot{y_A}$	$\dot{x_B}$	$\dot{y_B}$	$\dot{x_C}$	$\dot{y_C}$
t_x	1	0	1	0	1	0	1	0
t_y	0	1	0	1	0	1	0	1
r_{xy}	$-y_O$	x_O	$-y_A$	x_A	$-y_B$	x_B	$-y_C$	x_C
fl	0	0	0	0	0	0	$y_O - y_C$	$x_C - x_O$

Table 3: A possible base for the infinitesimal motions

check that $C(X_V + \epsilon\dot{X}) = 0$ is still true for all vectors $\dot{X}$ in the base of free motions of the witness X_V. Using Taylor, $C(X_V + \epsilon\dot{X}) = C(X_V) + \epsilon C'(X_V)\dot{X}^t + O(\epsilon^2)$, thus the gradient vector $C'(X_V)$ must be orthogonal to $\dot{X}$. In other words, $C'(X_V)$ must lie in the vectorial space spanned by the jacobian $F'(X_V)$. This procedure detects the false witnesses; for instance, if OC and OA are orthogonal in the witness of the last example, the orthogonality is detected as accidental and does not hold generically: the gradient vector of equation $\vec{OC}.\vec{OA} = 0$ is not orthogonal to the flexion vector.

Other typical and simple examples of 3D configurations, where the witness method detects dependencies (unlike graph-based methods), are given in Figure 2. The most left subfigure is classical and known as the double banana; the other three subfigures are from Ortuzar: four vertices are never coplanar.

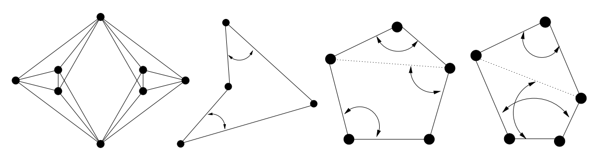

Figure 2: In 3D, the double banana, and three Ortuzar's configurations.

6 Conclusion

Graph-based methods for decomposing systems of equations or constraints have intrinsic and unavoidable limitations. The witness method successfully overcomes these limitations. This paper simplifies and generalizes the witness method; though several questions arise: is there always a rational witness? In the case of existing rational witness, is it possible to use only integer arithmetic to perform rank computations? Another interesting point is the degree of our comprehension of the system of equations and the ability to extract its relevant properties and use them during the decompose/solving process.

We feel the witness method can still be simplified and generalized to other kinds of invariance (*e.g.* modulo scaling or homography). Several clues (*e.g.* the duality between the maximal rigid part and the minimal dependent system) suggest that there is a deeper, simpler and more powerful theory for decomposition.

References

[1] B. Brunderlin, D. Roller. Eds. *Geometric Constraint Solving and Applications*, 1998.

[2] S. Foufou, D. Michelucci, J.-P. Jurzak. Numerical decomposition of geometric constraints. In *CAD Journal*, 2005.

[3] C. Hoffmann. Summary of basic 2D constraint solving. *International Journal of Product Lifecycle Management 1*, 143–149, 2006.

[4] C. Jermann, G. Trombettoni, B. Neveu, P. Mathis. Survey of decomposition methods. *International Journal of Computational Geometry and Applications*, Editors X.S. Gao and D. Michelucci, to be published.

[5] W. A. Martin. Determining the equivalence of algebraic expressions by hash coding. *J. ACM*, 18(4):549–558, 1971.

[6] J.T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 4(27):701–717, 1980.

[7] P. Schreck, P. Mathis. Geometrical constraint system decomposition: a multi-group approach. *International Journal of Computational Geometry and Applications*, Editors X.S. Gao and D. Michelucci, to be published.

[8] H. Servatius, J. Graver, B. Servatius. Combinatorial Rigidity. Graduate Studies in Mathematics. *American Mathematical Society*, 1993.

[9] G.F. Zhang, X.S. Gao. Spatial Geometric Constraint Solving Based on k-connected Graph Decomposition. *Proc. of The 21st Annual ACM Symposium on Applied Computing*, Dijon, France, ACM Press, 973–977, 2006.