Interrogating witnesses for geometric constraint solving

Sebti Foufou[a,b,*], Dominique Michelucci[b]

[a] *CSE dept, CENG, Qatar University, PO Box 2317, Doha, Qatar*
[b] *Le2i, CNRS 5158, Université de Bourgogne, BP 47870, 21078 Dijon, France*

Abstract

Classically, geometric constraint solvers use graph-based methods to decompose systems of geometric constraints. These methods have intrinsic limitations, which the witness method overcomes; a witness is a solution of a variant of the system. This paper details the computation of a basis of the vector space of free infinitesimal motions of a typical witness, and explains how to use this basis to interrogate the witness for dependence detection. The paper shows that the witness method detects all kinds of dependences: structural dependences already detectable by graph-based methods, but also non-structural dependences, due to known or unknown geometric theorems, which are undetectable by graph-based methods. It also discusses how to decide about the rigidity of a witness and how to decompose it.

Keywords: Geometric constraints, constraint solving, constraint decomposition, dependent and independent constraints, witness configuration, infinitesimal motions

1. Introduction

Shape modelling based on geometric constraints enables the designer to specify shapes as a set of geometric entities and their constraints and relationships. Geometric constraints are specifications of relations (*e.g.* distances, angles, incidences, tangencies, parallelisms, orthogonalities) between geometric elements such as points, lines, planes, conics, quadrics, or algebraic curves and surfaces of higher degree. Various problems in various domains can be formulated as geometric constraint systems that can be decomposed and solved using geometric constraint solving techniques. Big clients of geometric constraints are for example: robotics (*e.g.* generalized Stewart platform), molecular chemistry (*e.g.* the molecule problem which consists in finding the configurations of a molecule from interatomic distances), and geometric modelling for CAD-CAM (dimensioning mechanical parts) and virtual reality (*e.g.* blending surfaces) [1, 2, 3, 4, 5, 6].

*Corresponding author

Email addresses: sfoufou@u-bourgogne.fr, sfoufou@qu.edu.qa (Sebti Foufou), dmichel@u-bourgogne.fr (Dominique Michelucci)

The Systems of geometric constraints found in industry are increasingly larger. Decomposing such large systems into smaller subsystems is essential. Graph-based methods have been extensively used to perform this decomposition, to plan the resolution of subsystems and to merge their solutions [7, 1]. These methods rely more or less on a combinatorial count of degrees of freedom; they use graph flow computations, maximum matching or k-connectedness properties; they are polynomial time [8, 9, 10, 11].

Graph-based methods work very well for correct systems of constraints, and they indeed make it possible to solve systems which are intractable otherwise. These methods are even able to detect simple mistakes in systems of constraints, namely structural dependences, which may occur when a subset of unknowns is constrained by too many constraints. However, non-structural dependences, due to geometric theorems, cannot be detected with pure graph-based methods. Missing such dependences makes the solver fail to solve the system, and to give a relevant explanation to the designer. This is a serious drawback as the probability of existence of such dependences increases with the size of the system to be solved.

The difficulties in detecting dependences between geometric constraints are due to the fact that every system of algebraic equations is translatable into a system of point-line incidences in the projective plane with a size of the same order of magnitude. Because of this universal property of systems of point-line incidences [12], detecting dependences between such incidence constraints, or a superset of these constraints, is as hard as detecting dependences between algebraic equations; the latter, known in computer algebra as the ideal or radical membership problem, is decidable, for instance with standard bases (also known as Gröbner bases), but not practicable. Such seemingly trivial incidence constraints between flats (points, lines, planes) are essential in real-world problems, the molecule problem being an exception. Thus there is no reasonable hope to make pure graph-based methods, or other polynomial time methods robust against non-structural dependences due to geometric theorems and incidence constraints.

A set of geometric constraints gives a system of algebraic non- linear equations to solve. For CAD-CAM problems, a witness (a solution of a variant of the system) is usually available and can be used to check the independence between the geometric constraints, to decompose them, or to check that a decomposition proposed by any other method is correct. The concept of witness is defined in section 2.

The witness method was proposed in [13, 12], it is intended to help the designer build correct systems of geometric constraints. The current paper introduces the vector space of the free infinitesimal motions of the witness and shows how this vector space is computed and used to answer questions such as: Are the constraints coordinate-independent? Are constraints dependent on, or independent of each other? Is the witness flexible? Is the witness decomposable? Is a flexion generic or degenerate? Is the witness typical or not? This paper proves that the witness method detects all kinds of dependences, including non-structural dependences, due to known or unknown geometric theorems, which can not be detected with graph-based methods. It gives first ideas to detect atypical witnesses.

The paper is structured as follows. Section 2 presents the principle of the witness method and discusses the difficulty of finding a witness. Section 3 presents the free infinitesimal motions and how they are computed with rank considerations. Section 4 explains how to interrogate the witness for testing flexibility and rigidity. Section 5 presents a witness-based method to decompose geometric constraint systems. Section 6 deals with typicality issues. Section 7 concludes.

2. The witness method for systems of geometric constraints

Definition 1. *A System of geometric constraints is a system of algebraic non-linear equations $F(U, X) = 0$ to solve, where U is the vector of parameters and X is the vector of unknowns. Parameters may be geometric entities such as distances, angles, and/or non-geometric entities such as weights, forces, costs. Unknowns are coordinates of points, components of vectors, coefficients of lines or plane equations, etc. Equations are independent of the used Cartesian coordinate system.*

Definition 2. *A witness is a solution of a variant of the system to solve. It is a couple (U_W, X_W) such that $F(U_W, X_W) = 0$, where vectors U_W and X_W are respectively the numerical values of parameters U, and unknowns X at the witness. We refer to the unknown/searched configuration as the target (U_T, X_T), where vectors U_T and X_T are respectively the numerical values of parameters U, and unknowns X at the target.*

Figure 1 shows a target configuration and a possible witness configuration. Depending on the set of geometric constraints a witness may be degenerate, typical or atypical.

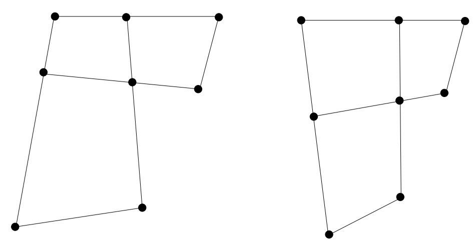

Figure 1: A target *(left)* and a witness *(right)* configurations in 2D. Constraints are collinearities, and some edge lengths or some angles.

Definition 3. *Degeneracy: in 2D, a witness is degenerate if it has 2 equal vertices, 3 aligned vertices, 4 cocyclic vertices, or 6 vertices on the same conic. In 3D, a witness is degenerate if it has 4 coplanar vertices, 5 cospherical vertices, or 10 vertices on the same quadric. A non-degenerate witness is also called a generic witness.*

Definition 4. *Typical witness: a witness is typical if it is generic, or it is degenerate but all its degeneracies are due to constraints, in which case all possible witnesses are degenerate.*

Definition 5. *Atypical witness: a witness is atypical if it contains a degeneracy and there exist other witnesses without this degeneracy.*

The witness method assumes that the witness is typical. Intuitively, a witness is typical if all its degeneracies, such as collinearity of three vertices or cocyclicity of four vertices in 2D, are due directly or indirectly to the constraints, and not to some numerical accidents. Thus all degeneracies of a typical witness also occur in the target (under mild assumptions discussed in 6.4). The word 'typical' is needed, and 'generic' can not be used instead, since a typical witness can be degenerate. Typicality is discussed in more details in Section 6.

2.1. Generating a witness: how difficult is it?

The sketch interactively provided by the user is assumed to be a typical witness, *e.g.* points which must be aligned or coplanar in the target are aligned or coplanar in the sketch, and only the generic angles and distances in the sketch need to be corrected by the solver. When no witness is available it is possible to automatize its computation by considering U, the vector of parameters, as unknowns, and by solving the very under-constrained $F(U, X) = 0$ system. Here are two examples of witness computations in 2D and in 3D.

- Find in 2D a triangle specified by its 3 lengths, and contains an incircle (an inscribed circle tangent to the 3 sides of the triangle). To find a witness the user (or the solver) starts from any circle, choose any 3 distinct points on the circle, and then trace the tangent lines at these points. The witness is the triangle defined by these tangents and their intersection points. If the three points are rationale the witness is rational, *e.g.* for the circle $x^2 + y^2 = 1$ the rational points are $((1-t^2)/(1+t^2), 2t/(1+t^2))$ for $t \in Q$.

- Find the 3D lines tangent to 4 given spheres (the 4P1L problem [14]). To find the witness choose any line, choose any 4 centers for the spheres, then compute the radius for the spheres: they are the distances between the line and the centres. This witness is rational. Again parameters in the target system become unknowns in the system characterizing witnesses, and their values are deduced by propagation. This also holds for all problems (the 6P or octahedral problem, and the 5P1L problem) mentioned in Hoffmann and Yuan's paper: it is easy to find a rational witness and the values of parameters in the target system are deduced by propagation. We conjecture that it holds for all problems soluble with the locus method by Gao, Hoffmann and Yang [15].

Finding the witness may be quite hard for some systems, but to our experience such systems are not relevant to CAD-CAM. Here is a summary of some cases where the difficulty is known.

Solving the molecule problem (given some interatomic distances, find the configuration of the molecule) is difficult [16, 17, 18], since it means solving an algebraic system, but finding a witness is completely trivial: it is enough to generate random points in 2D or in 3D according to the nature of the problem. This easiness suggests that finding a witness remains easy even when some incidence constraints are added.

After Steinitz's theorem [19], each 3D Eulerian polyhedron (satisfying Euler formula $V - E + F = 2$) is realizable with a 3D convex polytope with integer coordinates only. A constructive proof of Steinitz's theorem relies on the Tutte barycentric embedding of the planar graph of the polyhedron in the plane, then brings this embedding in 3D; it provides a cubic time algorithm [19]. This gives a cubic time method for computing a generic witness of Eulerian polyhedra. It is worth mentioning that, in contrast, 4D polytopes are not all realizable with integer coordinates only [19], and that all algebraic numbers are necessary for realizing 4D polytopes.

Much less seems to be known for 3D polyhedra with non-zero genus (*i.e.* with through holes). Finally, after the universality theorem, constrained arrangements (constrained configurations where geometric constrains are only incidences without any parameter) can be arbitrarily difficult to solve. Anyway, in spite of their aesthetic appeal, or the fact that some of them have known complexity, these problems are not relevant to geometric constraints in CAD-CAM.

2.2. Principle of the witness method

When a system is correct, the numeric solver in use can reliably solve it in a numerically stable way. A numerical solver can reliably compute a root in $\mathbb{R}^N$, as the intersection point between the N hypersurfaces described by the N equations of the system, only when the hypersurfaces cut *transversely*, *i.e.* when the tangent hyperplanes at the root cut *transversely*: it means that the N normal vectors to the tangent hyperplanes are linearly independent, *i.e.* the Jacobian has full rank at the solution point.

The witness method basically computes the Jacobian structure at the witness; it detects subsets of hypersurfaces which do not cut transversely, *i.e.* subsets of equations having dependent gradient vectors. We think that transversality is definitely the good criterion. It has very convenient features.

- Transversality of the witness (thus of the target) is decidable in polynomial time, it requires only standard tools of linear algebra.

- Transversality guarantees the convergence of the numerical solver in some neighborhood of the root (for the witness, and thus for target). Then classical methods from interval analysis compute such a neighborhood (a box, that is a vector of intervals), and provide guarantees.

- Transversality also guarantees that the root (the witness, or the target) is stable against small perturbations of the values of parameters U in the system $F(U, X) = 0$, more precisely it guarantees that the root is locally an implicit, continuous, and differentiable function of parameters U; interval analysis can compute and guarantee such a neighborhood for U and X.

- When the equations are transversal at the witness, but there is no root for the parameter values of the target, there is certainly something wrong with these parameters (*e.g.* a triangular inequality is violated).

- Finally, transversality is stable against the variations in the formulation of constraints. There are numerous ways to translate constraints into equations, and to choose unknowns. For instance, distances may be possible unknowns or parameters in a first solver, and they may be forbidden in a second solver which requires to square all distances to accept them as unknowns or parameters; this second choice makes sense for rational witnesses (when all vertices have rational coordinates) which are very frequent. This choice avoids square roots of rational numbers, so that all unknowns and parameters have rational values, which are exactly representable. Similarly, an angle θ can be represented by several kinds of variables: simply a variable θ, but also more algebraically $c_\theta = \cos\theta$, or $s_\theta = \sin\theta$, or $t_\theta = \tan\theta$, or $C_\theta = \cos^2\theta$, or $S_\theta = \sin^2\theta$, or $T_\theta = \tan^2\theta$. But for all these ways of formulating constraints or choosing unknowns, the mapping between two distinct formulations is locally (*i.e.* in the neighborhood of the witness, and thus in the neighborhood of the target) a diffeomorphism, that is a bijective, continuous and differentiable mapping. Transversality is preserved by diffeomorphisms, thus it is preserved through variations in the translation of constraints into equations, therefore results of the witness method are easily reproducible, and this paper does not need to be unduly precise about the translation of constraints into equations since, as far as the witness is concerned, all reasonable formulations are equivalent.

The witness is not assumed to lie in a neighborhood of the target; for instance, the target and the witness can lie on two distinct connected components of the real solution set of $F(U, X) = 0$.

2.3. Inaccuracy issue, and rational witnesses

With the witness method, problems related to geometric constraints will be reduced to computing the rank of a set of vectors, or deciding if vectors are independent or not. These vectors have known numerical coordinates. When vectors are independent, approximate computations are sufficient to reliably prove their independence, for instance with interval arithmetics, assuming of course that intervals are sharp enough.

Unfortunately numerical inaccuracy prevents the correct computation of the rank of dependent vectors; for instance, vectors $(1, 2)$ and $(1/\sqrt{5}, 2/\sqrt{5})$ are no longer proportional after floating-point rounding. To avoid these difficulties, this paper assumes, for simplicity, that the witness has rational coordinates, exactly represented. To avoid square roots of rational numbers, each distance, cosine or sine is represented by a variable equal to its square [12].

It seems that, for CAD-CAM, numerous problems have rational witnesses. However, it is fair to mention exceptions. An exception is provided by regular platonic solids: *e.g.* up to scaling, the regular dodecahedron and icosahedron have only finite set of witnesses, *e.g.* the convex usual embedding, and the symmetrical concave star shaped one; no realization is rational. Converting the metric constraints (equality of edges lengths) into projective ones yields constrained arrangements with no rational realization: a 2D example is the pentagonal star, equivalent to the regular pentagon [12].

2.4. Forerunners of the witness approach

The principle of the witness approach is not new:

- Classical artificial intelligence uses reasoning on examples for proving or guessing properties.
- In computer algebra, the cylindrical algebraic decomposition by Collins represents each 2D region, where a set of polynomials and all their derivatives have constant signs, by a typical point in this region: this point is a witness.
- The rigidity theory probabilistically decides the rigidity of graphs in any dimension in polynomial time using a typical example, called a structure or a framework: this is a witness.
- In computer algebra, algebraic identities or the nullity of a black box polynomial are tested probabilistically in polynomial time: *e.g.* a multivariate polynomial $f(x)$ is identically zero if it vanishes for a random point x (*e.g.* Schwartz-Zippel theorem, [20, 21]). This random point is a witness.
- Even in geometric constraint solving, the sketch interactively provided by the designer is already used by solvers, as an initial approximation of the root for iterative numerical methods, such as Newton-Raphson iteration, homotopy (or continuation), and gradient descent.

2.5. Particular considerations for the witness

The witness approach only considers the tangent hyperplanes: when they are dependent, it cannot decide if the witness (U_W, X_W) is an isolated root of a well-constrained system $F(U, X) = 0$, or if the witness lies on a curve, a surface, etc. In other words, it cannot compute the dimension[1] of the solution set of $F(U, X) = 0$ through X_W. Here are two examples of systems with exactly the same degenerate Jacobian, but having two zero sets of different dimensions. The

[1] The dimension is 0 for a discrete set, 1 for a curve, 2 for a surface, etc.

first system is $y = (x-3)^2 - y = 0$, the witness is $(3, 0)$, the Jacobian has rank 1, the witness is the only root, thus the dimension of the solution set is 0. The second system is $y = xy = 0$, a witness is also $(3, 0)$, the Jacobian has rank 1, but the line $y = 0$ is solution, so the dimension of the solution set is 1. For the second system, the witness cannot be the origin $(0, 0)$, because this is a critical point of the curve $xy = 0$.

2.5.1. Managing parameters

There are two ways to manage parameters: (i) Parameters are replaced by their values at the witness, so the only remaining symbols are the names of the unknowns. (ii) Parameters are considered as unknowns and for each $u \in U$ an equation $u - u_W = 0$ is added, where u_W is the numerical value of the parameter u in the witness. For conciseness, this paper uses the first way; since parameters are eliminated, the system $F(U, X) = 0$ can be rewritten $F(X) = 0$ whenever it is convenient.

2.5.2. Critical points

Definition 6. *A point p on an hypersurface $f(x) = 0$, $f : \mathbb{R}^n \to \mathbb{R}$ is a critical point iff the gradient vector at p vanishes: $f'(p) = \overrightarrow{0}$, i.e. the tangent hyperplane is undefined. A non-critical point is a regular point.*

An example of a critical point is the apex $(0, 0, 0)$ of the cone: $x^2 + y^2 - z^2 = 0$. The witness approach considers hyperplanes through the witness which are tangent to hypersurfaces corresponding to equations. Thus the witness must be a regular point for each hypersurface.

A critical point can occur in a system of geometric constraints if the distance between two vertices of the configuration is constrained to be 0: $(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2 = 0$. The latter constraint is not generic, and is forbidden by the genericity hypothesis of parameters. In the context of geometric constraints, it seems to be the only way to create a critical point.

It is easy to compute the gradient vector at the witness, and to check that the witness is a regular point for every hypersurface.

2.5.3. Determinant polynomial equations

Not all polynomial equations are given by an explicit list of coefficients and monomials. Sometimes polynomials are determinants of a square matrix [22, 23]. Computing such a determinant polynomial symbolically is exponential time (an $n \times n$ determinant has $n!$ factors). Fortunately, for the witness approach there is no need to apply this symbolic definition. We only compute the determinant and its derivatives at a given point: the witness. This can be done in polynomial time by exploiting the multilinearity of the determinant:

$$\frac{\partial}{\partial x} det(M) = \sum_{i=1}^{n} det\left[C_1, \ldots C_i' \ldots C_n\right]$$

where C_i is the i^{th} column of the matrix M and C_i' its derivative with respect to x. Computing the determinant of a square matrix with only numerical entries is cubic time.

3. Free infinitesimal motions of a typical witness

Definition 7. *Free infinitesimal motions are motions that can be applied to transform a constrained configuration without violating the constraints [16]. They are usually classified into two types: (i) infinitesimal displacements, namely translations, rotations and their compositions, which never deform the configuration; (ii) infinitesimal flexions (sometimes called deformations).*

Proposition 1. *A typical witness is rigid if it only admits infinitesimal displacements. It is flexible if it admits an infinitesimal flexion. i.e. the system of geometric constraints does not determine completely the geometric configuration.*

Generic flexions deform the configuration. Degenerate flexions do not; they only occur with atypical witnesses. An example of an atypical witness is a triangle with collinear vertices, where the collinearity is not due to the constraints. The set of atypical witnesses has measure zero, in the set of possible witnesses. Thus they are dismissed from now on, until Section 6 which discusses typicality issues.

Assume that a typical witness (U_W, X_W) is known, *i.e.* $F(U_W, X_W) = 0$. In a nutshell, the main idea of the witness method is to compute the vector space of the free infinitesimal motions $\dot{X}$ of the witness, such that the perturbed witness $X_W + \epsilon\dot{X}$, where ϵ is an infinitesimally small number, still fulfils the constraints: $F(U_W, X_W + \epsilon\dot{X}) = 0$. Taylor expansion gives $F(U_W, X_W + \epsilon\dot{X}) = F(U_W, X_W) + \epsilon F'(U_W, X_W)\dot{X}^t + O(\epsilon^2)$. Thus, for $F(U_W, X_W + \epsilon\dot{X})$ to be $O(\epsilon^2)$, infinitesimally small compared to the perturbation ϵ, the term $F'(U_W, X_W)\dot{X}^t$ must vanish: the vector space of the free motions is the kernel of the Jacobian matrix $F'(U_W, X_W)$ at the witness.

A basis of the infinitesimal displacements is computable *a priori*: it does not depend on the constraints, but only on the variables. Such a basis is provided below in section 3.1. The following conventions are used to describe the unknowns. In 2D, a point has coordinates (x, y); a line with equation $ax + by + c = 0$ is represented by a vector (a, b, c); a vector is represented by its components (u, v); this distinction between points and vectors is due to the fact that a translation (including an infinitesimal translation) modifies the (x, y) of points, but it does not modify the (u, v) of vectors; similarly translations do not modify the a, b coefficients of lines, but they modify the c coefficient. Under displacements, the variables u, v and a, b behave in the same way. Other geometric unknowns (barycentric coordinates, scalar products, distances, squared distances, angle cosines or squared cosines, or other trigonometric functions, areas, volumes) are unchanged by infinitesimal displacements, so the corresponding entries in all vectors of the basis are 0. This holds also for all non-geometric unknowns (weights, costs, densities, temperatures...).

3.1. Basis of infinitesimal displacements

It is possible to compute an *a priori* basis of the infinitesimal displacements. The top part of Table 1 shows such a basis, in the 2D case, composed of t_x a translation in the x direction, t_y a translation in the y direction, and r_{xy} a rotation around the origin. (x_i, y_i) are coordinates of a point, (a_l, b_l, c_l) are coordinates of a line (*i.e.* the line has equation: $a_l x + b_l y + c_l = 0$), and (u_k, v_k) are coordinates of a vector (the difference between two points). Dotted variables $\dot{x_i}, \dot{y_i}, \dot{a_l}, \dot{b_l}, \dot{c_l}, \dot{u_k}$, and $\dot{v_k}$ are used to denote the values of the corresponding coordinates in the basis of infinitesimal displacements, *e.g.* the couple $(\dot{x_i}, \dot{y_i})$ represents the infinitesimal translation t_x along the x axis of a point (x_i, y_i), it is equal to $(1, 0)$ (see the top part of Table 1). The proof of this basis is given in Section 3.3. Note that the infinitesimal displacements for a point (x, y), a

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

	$\dot{x_i}$	$\dot{y_i}$	$\dot{z_i}$	$\dot{a_h}$	$\dot{b_h}$	$\dot{c_h}$	$\dot{d_h}$	$\dot{u_k}$	$\dot{v_k}$	$\dot{w_k}$
t_x	1	0	0	0	0	0	$-a_h$	1	0	0
t_y	0	1	0	0	0	0	$-b_h$	0	1	0
t_z	0	0	1	0	0	0	$-c_h$	0	0	1
r_{xy}	$-y_i$	x_i	0	$-b_h$	a_h	0	0	$-v_k$	u_k	0
r_{xz}	$-z_i$	0	x_i	$-c_h$	0	a_h	0	$-w_k$	0	u_k
r_{yz}	0	$-z_i$	y_i	0	$-c_h$	b_h	0	0	$-w_k$	v_k

Table 1: Bases of free displacements: for points, lines, and vectors in 2D *(top)*, for points, planes, and vectors in 3D *(bottom)*.

normal (a, b) to a line, and a vector (u, v) are different; *e.g.* translating a point modifies it, but translating a vector or a normal does not.

In 3D, a basis of the infinitesimal displacements is $t_x, t_y, t_z, r_{xy}, r_{xz}, r_{yz}$, where t_z is a translation along z, r_{yz}, r_{xz}, r_{xy} are rotations around the x, y, and z axes. Corresponding coordinates are given in the bottom part of Table 1 which has as many columns as unknowns.

3.2. Example of structurally under-constrained systems

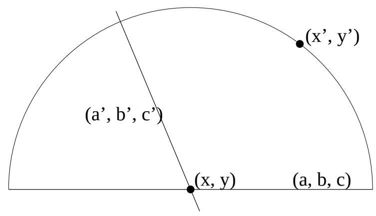

Figure 2: A 2D under-constrained system of geometric constraints.

A simple example in 2D is the typical system of six equations shown in (1) and represented by Figure 2, with generic parameters δ (a distance) and λ (a cosine). Point (x, y) lies on two lines (a, b, c) and (a', b', c'), with a specified

	x	y	x'	y'	a	b	c	a'	b'	c'
e'_1	a	b	0	0	x	y	1	0	0	0
e'_2	a'	b'	0	0	0	0	0	x	y	1
e'_3	$2(x-x')$	$2(y-y')$	$2(x'-x)$	$2(y'-y)$	0	0	0	0	0	0
e'_4	0	0	0	0	$2a$	$2b$	0	0	0	0
e'_5	0	0	0	0	0	0	0	$2a'$	$2b'$	0
e'_6	0	0	0	0	a'	b'	0	a	b	0
	$\dot{x}$	$\dot{y}$	$\dot{x'}$	$\dot{y'}$	$\dot{a}$	$\dot{b}$	$\dot{c}$	$\dot{a'}$	$\dot{b'}$	$\dot{c'}$
t_x	1	0	1	0	0	0	$-a$	0	0	$-a'$
t_y	0	1	0	1	0	0	$-b$	0	0	$-b'$
r_{xy}	$-y$	x	$-y'$	x'	$-b$	a	0	$-b'$	a'	0
flexion	0	0	$y-y'$	$x'-x$	0	0	0	0	0	0

Table 2: The Jacobian and a basis of infinitesimal motions: three displacements and a flexion for the system of equations (1). Variables are replaced by their values at the witness.

angle between them. Moreover the distance between points (x, y) and (x', y') is specified. Table 2 shows the Jacobian and a basis for the set of infinitesimal motions composed of three displacements and one flexion: the point (x', y') can rotate around the point (x, y). The reader can check that the vectors of infinitesimal motions are orthogonal to the gradient vectors (the derivatives) $e'_1, \dots e'_6$. A possible witness of this system is $(x = y = 0, x' = 3, y' = 4, \delta = 5, a = 1, b = 0, a' = 12/13, b' = 5/13, \text{and } \lambda = 12/13)$. To perform computations, the witness method replaces all variables $(x, y, x', y', a, b, c, a', b', c')$ with their numerical values at the witness in Table 2.

$$\begin{aligned}
e_1 &: ax + by + c = 0 \\
e_2 &: a'x + b'y + c' = 0 \\
e_3 &: (x - x')^2 + (y - y')^2 - \delta^2 = 0 \\
e_4 &: a^2 + b^2 - 1 = 0 \\
e_5 &: a'^2 + b'^2 - 1 = 0 \\
e_6 &: aa' + bb' - \lambda = 0
\end{aligned} \tag{1}$$

3.3. Proof of the basis of infinitesimal displacements

We only prove the basis for 2D infinitesimal displacements, the proof in 3D is similar. Let $P = (x, y, 1)$ be a point in homogeneous coordinates. P lies on a line $L = (a, b, c)$. A displacement represented by a matrix M maps the point P to the point $P' = (x', y', 1) = PM$, and the line L to the line $L' = (a', b', c')$, where $L'^t = M^{-1}L^t$ (Proof: $PL^t = 0$, then $P(MM^{-1})L^t = 0$, then $P'M^{-1}L^t = 0$, so $L'^t = M^{-1}L^t$).

For the infinitesimal translation t_x, the matrix M and its inverse M^{-1} are:

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \epsilon & 0 & 1 \end{pmatrix} \quad \text{and} \quad M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\epsilon & 0 & 1 \end{pmatrix}$$

and we have:

$$\begin{aligned}(x', y', 1) &= (x, y, 1)M &= (x + \epsilon, y, 1)\\ (a', b', c')^t &= M^{-1}(a, b, c)^t &= (a, b, -\epsilon a + c)^t\end{aligned}$$

thus: $\dot{x} = x' - x = \epsilon$, $\dot{y} = y' - y = 0$, $\dot{a} = a' - a = 0$, $\dot{b} = b' - b = 0$, $\dot{c} = c' - c = -\epsilon a$; dividing by ϵ gives t_x. Similarly, for the infinitesimal translation t_y along y.

For r_{xy}, the rotation around the origin with an infinitesimal angle, the matrix M and its inverse M^{-1} are (terms in ϵ^2 and higher degrees are ignored):

$$M = \begin{pmatrix} 1 & \epsilon & 0 \\ -\epsilon & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad M^{-1} = \begin{pmatrix} 1 & -\epsilon & 0 \\ \epsilon & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and thus:

$$\begin{aligned}(x', y', 1) &= (x, y, 1)M &= (x - \epsilon y, \epsilon x + y, 1)\\ (a', b', c')^t &= M^{-1}(a, b, c)^t &= (a - \epsilon b, b + \epsilon a, c)^t\end{aligned}$$

The difference between the identity matrix and the product of the two matrices is in $O(\epsilon^2)$, thus negligible compared to ϵ. Thus $\dot{x} = x' - x = -\epsilon y$, $\dot{y} = y' - y = \epsilon x$, $\dot{a} = a' - a = -\epsilon b$, $\dot{b} = b' - b = \epsilon a$, $\dot{c} = c' - c = 0$, and dividing by ϵ indeed gives r_{xy}. Vectors (u, v) are differences between two points, and thus $(\dot{u}, \dot{v})$ straightforwardly follows for all infinitesimal displacements.

3.4. Degrees of Displacements (DoD)

In an attempt to make graph-based methods more robust against dependences between constraints Jermann *et al* define degrees of rigidity [24]. We prefer to call them: degrees of displacements.

Definition 8. *The DoD of a rigid configuration (a set of points, lines, planes) is the number of equations needed to fix it in a Cartesian coordinate system.*

The DoD is difficult to compute with pure graph-based methods. Jermann *et al* mainly suggest formulas for big enough configurations and a tabulation for a finite set of small configurations; moreover the configurations need to be generic: incidence degeneracies (*e.g.* collinearities, coplanarities) due to geometric theorems are forbidden. This restriction is to prevent the universality theorem from confusing the graph-based methods.

The witness method computes straightforwardly the DoD by interrogating the typical witness, and requires no genericity hypothesis at all: for instance, the typical witness can contain three collinear points as long as this collinearity is a consequence of the system of constraints and holds for the target.

The witness method can determine which infinitesimal displacements are dependent. Let Y be a subset of X, the set of variables which describe the configuration, and D be a basis of the infinitesimal displacements at the witness. The DoD of Y is the rank of $D[Y]$, the subset of D relevant to Y.

Definition 9. *A part Y has full DoD if it has DoD 3 in 2D, or 6 in 3D.*

Let us consider the computation of the DoD in the following cases:

- For a line in 2D, $Y = \{a, b, c\}$, and $D[Y] = D[a, b, c]$ the basis of infinitesimal displacements for the line is extracted from the top part of Table 1 by keeping only variables relevant to the line. It is shown on the top left part of Table 3. $D[Y]$ has rank 2. We can even notice the two dependent translations t_x and t_y, which is correct as a translation of a line along another line leaves the translated line globally unchanged.
- For a segment in 3D, $Y = \{x, y, z, x', y', z'\}$, we just consider $D[Y]$ in the witness as it is shown in the top right part of Table 3. In this case $D[Y]$ has rank 5; the three translations are independent; the three rotations are dependent, they have rank 2; which is correct as the rotation around the line supporting the segment leaves it unchanged.
- For two secant planes in 3D, $Y = \{a, b, c, d, a', b', c', d'\}$, we also consider $D[Y]$ at the witness as it is shown in the bottom part of Table 3. It has rank 5; precisely, the three translations have rank 2, the three rotations are independent. In the same way, we can compute the DoD of two parallel planes, which is 4. The three translations have rank 2. Thus the DoD of two planes depends on the configuration (are they secant or parallel); it cannot be computed reliably with graph-based methods, which have no way to decide correctly if the two planes are secant or parallel.
- Similarly, the DoD of three collinear points in 3D is 5 (as for a segment), though the DoD of three non-collinear points is 6. Again, the interrogation of the typical witness gives the correct answer, while graph-based methods have no way to decide if the three points are collinear or not. Note that the three points may be collinear, not because of an explicit collinearity constraint, but because of a geometric theorem.

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

	$\dot{x}$	$\dot{y}$	$\dot{z}$	$\dot{x}'$	$\dot{y}'$	$\dot{z}'$
t_x	1	0	0	1	0	0
t_y	0	1	0	0	1	0
t_z	0	0	1	0	0	1
r_{xy}	$-y$	x	0	$-y'$	x'	0
r_{xz}	$-z$	0	x	$-z'$	0	x'
r_{yz}	0	$-z$	y	0	$-z'$	y'

	$\dot{a}$	$\dot{b}$	$\dot{c}$	$\dot{d}$	$\dot{a}'$	$\dot{b}'$	$\dot{c}'$	$\dot{d}'$
t_x	0	0	0	$-a$	0	0	0	$-a'$
t_y	0	0	0	$-b$	0	0	0	$-b'$
t_y	0	0	0	$-c$	0	0	0	$-c'$
r_{xy}	$-b$	a	0	0	$-b'$	a'	0	0
r_{xz}	$-c$	0	a	0	$-c'$	0	a'	0
r_{yz}	0	$-c$	b	0	0	$-c'$	b'	0

Table 3: Bases of infinitesimal displacements, for a line in 2D *(top left)*, for a segment in 3D *(top right)*, and for two secant or parallel planes in 3D *(bottom)*. Variables are replaced with their values at the witness.

4. Interrogations of a typical witness

4.1. Are constraints coordinate-independent?

Usually correct geometric constraints are coordinate-independent. However, coordinate-dependent constraints such as $x_p = 0$ are sometimes needed, to pin the configuration in the plane or in the 3D space, because numerical solvers expect systems with as many unknowns as equations.

Proposition 2. *A constraint is coordinate-dependent if its gradient vector is not orthogonal to at least one of the vectors in the basis of infinitesimal displacements.*

For instance, the constraint $x_p = 0$ is orthogonal to the vectors of the translation in y and of the rotation around the origin, but not to the vector of the translation in x. All equations can be tested this way. These tests are only numerical: the witness and the basis of infinitesimal displacements at the witness are numerical vectors. In the following sections of this paper the equations are assumed to be coordinate-independent.

	x_O	y_O	x_A	y_A
e_1'	2	0	-1	0
e_2'	0	2	0	-1
e_3'	$2x_A - 2x_C$	$2y_A - 2y_C$	$2x_O - 2x_A$	$2y_O - 2y_A$
e_4'	0	0	$x_B - x_A$	$y_B - y_C$
e_5'	$2x_O - 2x_A$	$2y_O - 2y_A$	$2x_A - 2x_O$	$2y_A - 2y_O$
	$\dot{x_O}$	$\dot{y_O}$	$\dot{x_A}$	$\dot{y_A}$
t_x	1	0	1	0
t_y	0	1	0	1
r_{xy}	$-y_O$	x_O	$-y_A$	x_A
flexion	0	0	0	0

	x_B	y_B	x_C	y_C
e_1'	-1	0	0	0
e_2'	0	-1	0	0
e_3'	0	0	$2x_C - 2x_O$	$2y_C - 2y_O$
e_4'	$x_A - x_C$	$y_A - y_C$	$2x_C - x_A - x_B$	$2y_C - y_A - y_B$
e_5'	0	0	0	0
	$\dot{x_B}$	$\dot{y_B}$	$\dot{x_C}$	$\dot{y_C}$
t_x	1	0	1	0
t_y	0	1	0	1
r_{xy}	$-y_B$	x_B	$-y_C$	x_C
flexion	0	0	$y_O - y_C$	$x_C - x_O$

Table 4: The Jacobian, and a basis of four free infinitesimal motions for the dependent system given in (2). The fourth motion is a flexion: point C can rotate around O. Variables are replaced with their values at the witness.

4.2. Are constraints dependent or independent?

Graph-based methods can detect only structural dependences, as in the system: $f(x, y, z) = g(z) = h(z) = 0$ which over-constrains the unknown z. The interrogation of the witness makes it possible to detect non-structural as well as structural dependences.

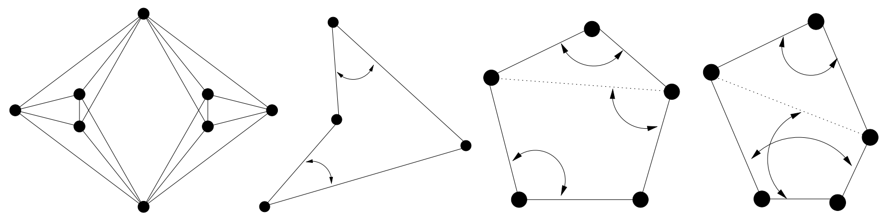

Figure 3: In 3D, the double banana *(left)*, and three configurations due to Ortuzar (personal communication). Edges denote specified distances, arrows denote specified angles. No four vertices are coplanar.

Proposition 3. *The constraints are dependent if the gradient vectors of the equations at the witness, i.e. the Jacobian matrix at the witness, are dependent. Computing a basis of this Jacobian is enough.*

Some simple examples of 3D configurations where the witness method detects dependences (unlike graph-based methods) are given in Figure 3. The leftmost configuration is classical, and is known as the double banana. The dependence in the double banana was already detected by a classical numerical probabilistic method [16], which the witness method encompasses. The systems of geometric constraints resulting from Ortuzar's three configurations are dependent; their dependence is not structural, graph-based methods can not detect it.
It is possible to tune pure graph-based methods to make them detect the simplest dependences (*e.g.* [1, 24, 25, 26]). But, the universal theorem turns the problem of detecting all kinds of dependences into an intractable one.

4.3. Example of non-structural dependence

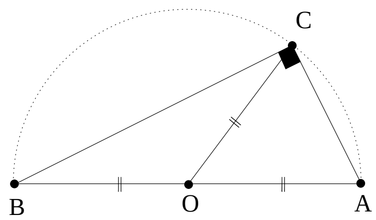

Figure 4: Example of dependent constraints.

Let us consider an example of dependence detection by witness interrogation. Suppose that we have four 2D points A, B, C, and O with the following constraints: (i) the distance OA is specified by a parameter u, (ii) O is the middle of the segment AB, (iii) distances OC and OA are equal, and (iv) AC and BC are orthogonal (see Figure 4). This last constraint results from the previous ones, this is due to a geometric theorem: if C lies on the circle with

diameter AB, then AC and BC are orthogonal.
These constraints result into the system of equations:

$$\begin{aligned}
&e_1 : 2x_O - x_A - x_B = 0\\
&e_2 : 2y_O - y_A - y_B = 0\\
&e_3 : (x_C - x_O)^2 + (y_C - y_O)^2 - (x_A - x_O)^2 - (y_A - y_O)^2 = 0\\
&e_4 : (x_C - x_A)(x_C - x_B) + (y_C - y_A)(y_C - y_B) = 0\\
&e_5 : (x_A - x_O)^2 + (y_A - y_O)^2 - u^2 = 0
\end{aligned} \tag{2}$$

A possible witness for this system of constraints is: $O = (0, 0), A = (-10, 0), B = (10, 0), C = (6, 8), u = 10$. Table 4 displays the Jacobian and a basis of the free infinitesimal motions: three displacements and a flexion, point C can rotate around point O. The rank of $e'_1, \dots e'_5$ computed at the witness is 4, thus equations are dependent. We point out that this system is structurally well-constrained as graph-based methods cannot detect the dependence. Table 4 is divided in two parts only for the sake of presentation.

Every geometric theorem can be used to generate a system of geometric constraints which contains non-structural dependences: translating into constraints the hypotheses and the conclusion (or a negation of the conclusion) of the theorem gives a dependent system. Section 4.7 shows that all kinds of dependences can be detected, as far as a typical witness is available. However, if the conclusion is denied, and if the conclusion is a collinearity or another incidence (more generally, a constraint that the witness must fulfil), then no witness can exist due to the contradiction, and the witness method does not apply.

4.4. Minimal dependent set of constraints

If the constraints are dependent, then the interrogation of the witness permits to find the smallest dependent set of constraints: this information is relevant to the user who can fix the mistake more easily even in a large system of constraints. This problem reduces to finding the minimal dependent set in a dependent set of vectors (they are the gradient vectors of the equations at the witness). We assume that the rank of the dependent set is its cardinal minus one: the last vector we try to add in the basis reduces to the null vector. In such a case, the minimal dependent set is unique; to find it, just try to remove each vector in the dependent set; if the set minus this vector is still dependent, then remove this vector. The remaining set of vectors is then the minimal dependent set. This greedy method can be proved with the matroid theory.

4.5. Flexibility test: is the system flexible?

Proposition 4. *A system of geometric constraints is flexible iff the basis of the kernel of the Jacobian at the typical witness (i.e. the basis of infinitesimal motions) contains vectors outside the vector space generated by the basis of the infinitesimal displacements.*

For instance, in the classical configuration of the double banana, the two bananas can rotate around the axis through their two common vertices; the corresponding infinitesimal flexion is detected by the method. If the system is flexible, then the witness method can provide a basis of the infinitesimal deformations, and the set of maximal rigid subparts.

4.6. Rigidity test: is a part rigid?

A flexible system can contain rigid parts. A part is described by a subset Y of the unknowns. On Table 1, each variable corresponds to a column, and a part Y is thus a subset of columns.

Proposition 5. *A part Y is rigid iff the vector space $M[Y]$, the free infinitesimal displacements in the columns Y, is equal to the vector space $D[Y]$, the free infinitesimal displacements in the columns Y. Components of $M[Y]$ are obtained by taking only the columns Y in the vectors of the basis of M. Similarly, components of $D[Y]$ are obtained by taking only the columns Y in the vectors of the basis of D.*

For instance, in the system defined by equations (1) and Table 2, $Y = \{x, y, a, b, c, a', b', c'\}$ is rigid, but $Y \cup \{x', y'\}$ is flexible: it does not depend on the basis chosen for M and D. In the system defined by (2) with the Jacobian and a basis of infinitesimal motions shown on Table 4, the part $Y = \{x_O, y_O, x_A, y_A, x_B, y_B\}$ is rigid, while $Y \cup \{x_C\}$, $Y \cup \{y_C\}$, and $Y \cup \{x_C, y_C\}$ are not rigid. Again the rigidity is independent of the chosen basis.

4.6.1. Are A and B relatively fixed?

A flexible system can fix some pairs of geometric elements (two points, two lines, one point and one line, etc) relatively to each other. Actually, the previous section already provides a decision procedure.

Proposition 6. *Two geometric elements A and B are relatively fixed by the (possibly flexible) system if the part $Y = A \cup B$ is rigid.*

4.7. All kinds of dependences are detected

This section proves that the witness method detects all dependences in algebraic systems, including non-structural dependences due to known or unknown geometric theorems. Structural dependences are due to trivial theorems, they are detected as well; an example of a structural dependence is the over-constrainedness in $f(x, y, z) = g(z) = h(z) = 0$, where $(x, y, z) \in \mathbb{R}^3$. All geometric theorems (Pappus, Pascal, Desargues, their duals, etc.) relevant to geometric constraint solving are algebraically expressed by the fact that $f_1(x) = \dots f_n(x) = 0 \Rightarrow g(x) = 0$; here the $f_i(x) = 0$ express the hypothesis of the theorem, and $g(x) = 0$ is its conclusion.

Algebraically, there are two possibilities for an algebraic equation $g(x) = 0$ to be a consequence of other algebraic equations $f_1(x) = \dots f_n(x) = 0$. This depends on whether g is in the ideal or in the radical of $(f_1, f_2, \dots f_n)$. In both cases, the witness interrogation detects some linear dependence in the Jacobian of the typical witness. These two possibilities are defined by the following two theorems.

Theorem 1. *if g is in the ideal of $(f_1, f_2, \dots f_n)$, then the gradient vector of g at every common root x of $f_1, f_2, \dots f_n$ is dependent on the gradient vectors $f_1'(x), \dots f_n'(x)$.*

Proof. Let $F = (f_1, f_2, \dots f_n)$ be the polynomials of some algebraic system $F(x) = 0$. Let g be a polynomial lying in the ideal generated by $f_1, f_2, \dots f_n$. Then, by definition, there are polynomials λ_i such that $g = \lambda_1 f_1 + \dots + \lambda_n f_n$. Let x be a root of F; then x is also a root of g: $f_1(x) = \dots = f_n(x) = 0 \Rightarrow g(x) = \lambda_1(x)f_1(x) + \dots + \lambda_n(x)f_n(x) = 0$. After deriving we get $g'(x) = \lambda_1'(x)f_1(x) + \lambda_1(x)f_1'(x) + \dots \lambda_n'(x)f_n(x) + \lambda_n(x)f_n'(x) = \lambda_1(x)f_1'(x) + \dots \lambda_n(x)f_n'(x)$. The gradient vector of g at x lies in the vector space spanned by the gradient vectors $f_1', \dots f_n'$ of F (in other words, $g'(x)$ does not cut $F'(x)$) transversally. $\square$

Theorem 2. *if g is in the radical of $(f_1, f_2, \dots f_n)$, but not in the ideal, then gradient vectors $f_1'(x), \dots f_n'(x)$ are linearly dependent at every common root x.*

Proof. The other possibility for the vanishing of g to be a consequence of the vanishing of $f_1, \dots f_n$, is that g lies in the radical generated by $(f_1, \dots f_n)$, *i.e.* there is an integer $k \geq 2$ such that g^k lies in the ideal generated by $(f_1, \dots f_n)$. Here, the fact that g is in the radical of $(f_1, \dots f_n)$ does not imply that the gradient vector of g at a root x of $(f_1, \dots f_n)$ lies in the vector space spanned by the gradient vectors F' of F (*e.g.* $g(x, y) = y$, $k = 2$, $f_1 = x^2 + y^2 - 1$, $f_2 = x^2 - 1$, so $g^k = f_1 - f_2$). But it implies that the gradient vectors of $f_1, \dots f_n$ at a common root x are linearly dependent: deriving $-g^k + \sum \lambda_i f_i = 0$ yields $-kg^{k-1}g' + \sum \lambda_i' f_i + \sum \lambda_i f_i' = 0$. If x is a common root of $(f_1, \dots f_n)$, it is also a root of g. Accounting for the fact that $k \geq 2$ (*i.e.* g is in the radical and not in the ideal), we obtain $\sum \lambda_i(x) f_i'(x) = 0$. Thus the $f_i'(x)$ are dependent. $\square$

5. Decomposition based on a typical witness

When combined with graph-based methods, the witness decision procedures of section 4 are sufficient to decompose systems of geometric constraints. Actually it is even possible to rely only on the witness to define a new decomposition method, such a new method uses the notions of anchors and maximal rigid parts presented below.

5.1. Anchors: rigid and full DoD parts

Definition 10. *An anchor $Y \subset X$ is a part which is rigid and has full DoD: the vector space of its free motions is equal to the vector space of the infinitesimal displacements; moreover the anchor has minimal cardinality, either in the geometric sense: the anchor is a set of geometric elements (points, lines, planes), or in the algebraic sense: the anchor is a set of variables.*

Anchors are used as seeds when computing the maximal rigid part containing an anchor (see Section 5.2). In 2D, a geometric anchor can be two points within a distance fixed by the system (directly or not), but an anchor cannot be only one point, *e.g.* in Figure 5 the set {A,B} is an anchor. An anchor can be a line and a non-incident point. It can also be made of three secant non-concurrent lines, etc. In 3D, a geometric anchor can be three points positioned within three distances fixed by the system. It cannot be a segment as we have seen that the DoD of a segment in 3D is not 6 but 5. Clearly a configuration contains only a polynomial number of geometric anchors. Every geometric anchor contains an algebraic anchor.

An algebraic anchor contains $r = 3$ variables in 2D, and $r = 6$ in 3D. There is a polynomial number $O(|X|^r)$ of potential algebraic anchors in the system with variables X ($|X|$ is the cardinal of X). To find algebraic anchors, just check for all $O(|X|^r)$ potential anchors that they are rigid and have full DoD. We have no space to mention optimizations, but this is polynomial time anyway.

For example, in 2D, a possible algebraic anchor is the set of variables x_1, y_1, y_2, iff the distance between the points (x_1, y_1) and (x_2, y_2) is directly or indirectly fixed by the system; this anchor does not use x_2. Note that variables in an *algebraic* anchor can be assigned null values. Similarly, in 3D, a possible anchor is the set of variables $x_1, y_1, z_1, y_2, z_2, z_3$ if the three distances are fixed (directly or indirectly) by the system. The previous definition of algebraic anchors is geometrically counter intuitive, since it breaks geometric elements into variables.

5.2. MRPs: maximal rigid parts

Definition 11. *A part $Y \subset X$ is an MRP iff Y is rigid and there is no $Y' \subset X$ such that $Y \subsetneq Y'$ and Y' is rigid.*

For instance, $Y = \{x, y, a, b, c, a', b', c'\}$ is an MRP for the configuration given by Figure 2, system (1), and Table 2. We denote MRP(E) the maximal rigid part that includes E. This definition makes sense only if E is an anchor (*i.e.* E is rigid and has full DoD).

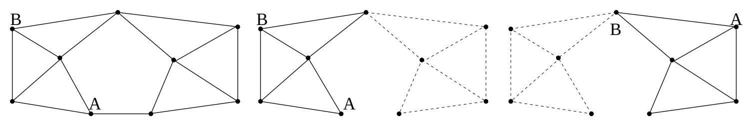

Figure 5: A 2D rigid system of constraints. Removing a constraint creates a flexible system with two Maximal Rigid Parts (MRP). In the three figures, the MRP(A, B) is in solid lines.

Figure 5 gives examples of MRPs and their computation: starting from the rigid configuration on the left subfigure, removing one constraint will give the MRP that includes the anchor defined by points A and B, so depending on the position of these points, the $MRP(A, B)$ will be the one in the middle or the one in the right subfigures.

If a part Y is rigid and has full DoD, then Y is an anchor, it is contained in a unique (MRP) defined by a set of variables R. R is computed using the following greedy method: initialize R with Y, and for each variable $x \in X - Y$, if $Y \cup x$ is rigid, then insert x in R (the test: if $Y \cup x$ is rigid, can be replaced by the test: if $R \cup x$ is rigid).

The set Ω of all maximal rigid parts is initialized to $\emptyset$. For all potential anchors A, if A is not already included in an MRP in Ω, then insert MRP(A) in Ω. The number of MRP is polynomial: there is only one MRP per anchor, and the number of potential anchors is polynomial. Thus this method is polynomial time. Usually the number of $MRPs$ is much smaller than the number of potential anchors.

5.3. The decomposition method

A decomposition method, which relies only on the witness without any combination with graph-based methods, considers the array of the Jacobian and the basis of free infinitesimal motions, and works as follows: if the configuration is flexible, the method finds its maximal rigid parts (MRP), if the configuration is rigid, the method removes each constraint in turn in order to make it flexible, and then computes the $MRPs$. Some book-keeping may be needed to avoid finding the same MRP several times over, but even without book-keeping the method is polynomial time. Once the decomposition is available, any classical method can be used to plan the resolution, *e.g.* the C-tree method [6].

6. Typicality of the witness

6.1. Detecting Degeneracies in the witness

Degeneracies are detectable automatically or by the user's visual inspection. The method to detect degeneracies in the witness is straightforward: for instance, to detect collinearities of 3 vertices, it is enough to test the collinearity of all triples of vertices in the witness. In practice, we dot not search the 6 points on the same conic and the 10 points on the same quadric degeneracies: the test is terribly costly and useless.

6.2. Atypicality test

The witness method relies on the assumption that the witness is typical. This section discusses typicality. Examples of 2D atypical witnesses are witnesses with three collinear points, or four cocyclic points, or six points on the same conic, etc, when these features are not due to the constraints, and thus do not occur in the target. In practice, atypical witnesses make the witness method fail.

Consider the simplest case of a triangle specified by its three lengths in 2D. In an atypical witness, the three vertices P_0, P_1, P_2 defining this triangle are collinear. The Jacobian of the atypical witness has rank 2, while the Jacobian of the typical witness has rank 3. The collinearity of P_0, P_1, P_2 is expressed by the vanishing of the polynomial $p(X_W)$ where

$$p(x) = \begin{vmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix} = x_1y_2 - x_2y_1 - x_0y_2 + x_2y_0 + x_0y_1 - x_1y_0$$

The gradient vector $p'(X_W)$ of the equation $p(x) = 0$ at the witness is given in Table 5. This gradient does not lie in the vector space generated by the

	x_0	y_0	x_1	y_1	x_2	y_2
p'	$y_1 - y_2$	$x_2 - x_1$	$y_2 - y_0$	$x_0 - x_2$	$y_0 - y_1$	$x_1 - x_0$

Table 5: The gradient vector of the equation expressing the collinearity of three points $P_i = (x_i, y_i)$, $i = 0 \dots 2$. Variables are replaced by their values in the witness.

gradient vectors of the Jacobian of the system, whatever the witness (except if $P_0 = P_1 = P_2$), thus the collinearity is not due to the constraints, but to the atypicality of the witness. For the very atypical witness with three equal vertices $P_0 = P_1 = P_2$, the same test applies and shows that these degeneracies, for instance $p(x) = x_0 - x_1 = 0$, are not due to the constraints but to the atypicality of the witness.

The atypicality test applies for all degeneracies in the witness, *e.g.* equality of two vertices, collinearity of three vertices in 2D or 3D, coplanarity of four vertices in 3D, cocyclicity of four coplanar points, etc. Each kind of degeneracy is expressed with a specific coordinate-independent polynomial $p(X_W) = 0$. Assume that the Jacobian $F'(X_W)$ has full rank, *i.e.* redundant equations have been removed: if $p'(X_W)$ does not lie in the vector space of $F'(X_W)$, then the degeneracy $p'(X_W) = 0$ is surely due to the witness, and not to the constraints: the witness is atypical beyond doubt, this follows from the proof in Section 4.7, first case: p does not lie in the ideal of F; thus the vanishing of $p(X_W)$ is accidental and so the witness is atypical. On the other hand, if $p'(X_W)$ lies in the vector space of $F'(X_W)$, then the degeneracy $p'(X_W) = 0$ is *probably* due to the constraints and not to the witness, so the witness is *probably* typical. In this case, the confidence and the witness method are only probabilistic. This probabilistic confidence costs polynomial computational time: determinist proofs cost at least exponential computational time. Schwartz' method [20, 21] uses the same kind of probabilistic argument and features the same asymmetry: a given polynomial is *probably* identically zero when it vanishes at a random value chosen independently of the polynomial, and it is surely not identically zero when it does not vanish.

6.3. Perturbing the witness to make it typical?

Let X_W be the initial witness. Choose a random vector T with unit norm in the vector space of the free motions of the witness. Possibly replace T with its component perpendicular to the space of infinitesimal displacements. Then

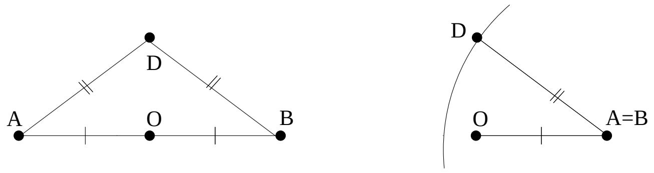

Figure 6: This system, due to C. Jermann [24], is built in order to have two kinds of solutions – thus two kinds of typical witnesses. Constraints are: points O, A, B are collinear, distances OA and OB are equal and specified, distances DA and DB are equal and specified. The left solution is rigid, the right one is flexible.

perturb X_W into $X = X_W + \rho T$ where ρ is a step size similar to the prediction step of prediction-correction methods. Finally correct X with some Newton-Raphson iterations until it lands on the solution set to get the new witness. Apply this prediction-correction steps several times to remove all degeneracies specific to the atypical witness.

This perturbation method is polynomial time and can be displayed, which permits users to see the free flexions of the configuration, and it gives them an opportunity to formulate new constraints. A drawback of the perturbation scheme is that it is intrinsically a numerical method that uses approximate computations: starting from a rational witness, the perturbed witness is no more rational. The possibility to deform an initial atypical witness randomly, continuously and successfully to make it typical is an open question.

6.4. Solution sets heterogeneous in dimension

Most of the time, degeneracies of a typical witness also occur in the target, since they are due to the constraints. However, this does not hold in some very exceptional cases, when the solution set of the system is heterogeneous in dimensions – like for instance the Whitney umbrella in 3D which contains one curve and one surface. Fig. 6, due to Christophe Jermann from Nantes University in France, gives an admittedly very artificial example of such a system for which both rigid, and flexible solutions are available. When the witness and the wanted target are not of the same type, though typical, the witness is not typical of the target. Detecting that a solution set is, exceptionally, heterogeneous in dimensions is a difficult task.

7. Conclusion

Classical graph-based methods for decomposing systems of equations or constraints have intrinsic limitations: they do not detect all dependences between constraints. This paper proves that the witness method overcomes this limitation if a typical witness is available. It shows how to interrogate the witness, and how all computations reduce to the polynomial time computation of the rank or a base of a set of numerical vectors. It proposes a test to detect atypical witnesses. The paper provides answers to some essential questions for geometric constraints, but other questions are still under investigation: *e.g.* which method can be used to generate a typical witness? Does it make sense to consider only rational witnesses? Is it possible to use approximations (*i.e.* floating-point coordinates, or intervals) to represent witnesses?

8. Acknowledgements

This research work has been funded in part by NPRP grant number NPRP-09-906- 1-137 from the Qatar National Research Fund (a member of the Qatar Foundation).

References

[1] J. Owen, Constraint on simple geometry in two and three dimensions., Int. J. Comput. Geometry Appl. 6 (4) (1996) 421–434.

[2] B. Bruderlin, D. Roller (Eds.), Geometric Constraint Solving and Applications, Springer-Verlag, 1998.

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

[4] C. Jermann, G. Trombettoni, B. Neveu, P. Mathis, Decomposition of geometric constraint systems: a survey, Internation Journal of Computational Geometry and Applications (IJCGA) 16 (5-6) (2006) 379–414.

[5] X.-S. Gao, C. Hoffmann, W. Yang, Solving spatial basic geometric constraint configurations with locus intersection, Computer Aided Design 36 (2) (2004) 111–122.

[6] X.-S. Gao, G. Zhang, Geometric constraint solving via c-tree decomposition, in: ACM Solid Modelling, ACM Press, New York, 2003, pp. 45–55.

[7] J. Owen, Algebraic solution for geometry from dimensional constraints, in: Proc. of the Symp. on Solid Modeling Foundations and CAD/CAM Applications, 1991, pp. 397–407.

[8] W. Bouma, I. Fudos, C. Hoffmann, J. Cai, R. Paige, Geometric constraint solver, Computer-Aided Design 27 (6) (1995) 487–501.

[9] I. Fudos, C. Hoffman, A graph constructive approach to solving systems of geometric constraints, ACM Trans. on Graphics 16 (2) (1997) 179–216.

[10] C. Hsu, Graph-based approach for solving geometric constraint problems, Ph.D. thesis, University of Utah, Dept. of Comp. Sci. (1996).

[11] M. Sitharam, J.-J. Oung, Y. Zhou, A. Arbree, Geometric constraints within feature hierarchies., Computer-Aided Design 38 (1) (2006) 22–38.

[12] D. Michelucci, S. Foufou, Geometric constraint solving: the witness configuration method, Computer Aided Design 38 (4) (2006) 284–299.

[13] S. Foufou, D. Michelucci, J.-P. Jurzak, Numerical decomposition of geometric constraints, in: Proc. of ACM Symp. on Solid and Physical Modelling, 2005, pp. 143–151.

[14] C. M. Hoffmann, B. Yuan, On spatial constraint solving approaches, in: Automated Deduction in Geometry, 2000, pp. 1–15.

[15] X.-S. Gao, C. M. Hoffmann, W.-Q. Yang, Solving spatial basic geometric constraint configurations with locus intersection, in: SMA '02: Proceedings of the seventh ACM symposium on Solid modeling and applications, ACM Press, New York, NY, USA, 2002, pp. 95–104. doi:http://doi.acm.org/10.1145/566282.566299.

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

[17] B. Hendrickson, Conditions for unique realizations, SIAM J. Computing 21 (1) (feb 1992) 65–84.

[18] G. M. Crippen, T. F. Havel, Distance Geometry and Molecular Conformation, Research Studies Press, Taunton, U.K., ISBN 0-86380-073-4, 1988.

[19] J. Richter-Gebert, Realization Spaces of Polytopes, Lecture Notes in Mathematics 1643, Springer, 1996.

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

[21] Zippel, Effective Polynomial Computation, Kluwer Academic Publishers, 1993.

[22] D. Michelucci, S. Foufou, L. Lamarque, P. Schreck, Geometric constraint solving: some tracks, in: Proc. of ACM Symp. on Solid and Physical Modelling, 2006, pp. 185–196.

[23] D. Michelucci, S. Foufou, Using Cayley Menger determinants, in: Proc. of ACM symposium on Solid modeling, 2004, pp. 285–290.

[24] C. Jermann, B. Neveu, G. Trombettoni, Algorithms for identifying rigid subsystems in geometric constraint systems, in: 18^{th} International Joint Conference in Artificial Intelligence (IJCAI-03), 2003, pp. 233–238.

[25] C. Hoffmann, A. Lomonosov, M. Sitharam, Decomposition plans for geometric constraint systems, Part I : Performance measures for cad, J. Symbolic Computation 31 (2001) 367–408.

[26] C. Hoffmann, A. Lomonosov, M. Sitharam, Decomposition plans for geometric constraint problems, Part II : New algorithms, J. Symbolic Computation 31 (2001) 409–427.