# Detecting All Dependences in Systems of Geometric Constraints Using the Witness Method

Dominique Michelucci and Sebti Foufou

LE2I UMR CNRS 5158, UFR Sciences, Université de Bourgogne,
BP 47870, 21078 Dijon Cedex, France
dmichel@u-bourgogne.fr, sfoufou@u-bourgogne.fr

**Abstract.** In geometric constraints solving, the detection of dependences and the decomposition of the system into smaller subsystems are two important steps that characterize any solving process, but nowadays solvers, which are graph-based in most of the cases, fail to detect dependences due to geometric theorems and to decompose such systems. In this paper, we discuss why detecting all dependences between constraints is a hard problem and propose to use the witness method published recently to detect both structural and non structural dependences. We study various examples of constraints systems and show the promising results of the witness method in subtle dependences detection and systems decomposition.

## 1 Introduction

Today all CAD CAM geometric modelers provide a geometric solver that enables designers to define shapes (geometric configurations) as solutions of a set of geometric constraints [3,24,11,8]. Geometric constraints specify distances, angles, incidences, and tangencies between basic geometric elements such as points, lines, circles, conics or higher degree curves (*e.g.* Bézier curves) in 2D, and lines, planes, quadrics or higher degrees algebraic curves and surfaces in 3D. In practice, designers interactively specify constraints on an approximation of the wanted configuration (called a "sketch") – the solver is often called a sketcher. The solver operates in various steps: (i) reads the sketch; (ii) translates the system of constraints into some internal data structure (typically some graph, and a system of equations...); (iii) analyses and decomposes the system; (iv) solves the subsystems obtained from the decomposition either with some formula or with a numerical method; (v) and finally assembles solutions of subsystems and displays the corrected sketch.

As the system is typically non linear, there is usually more than one solution, and the solver is supposed to provide the solution that gives the closest configuration to the intention of the designer. It turns out that, in 90% of the cases, the Newton-Raphson method converges to this solution when it starts from the initial guess provided by the sketch. When the Newton-Raphson method fails, the designer can resort to another method, slower but safer, like homotopy: for

F. Botana and T. Recio (Eds.): ADG 2006, LNAI 4869, pp. 98–112, 2007.
© Springer-Verlag Berlin Heidelberg 2007

an algebraic system of equations, the attraction basins for homotopy are semi algebraic sets, the ones of Newton-Raphson method are fractals; this is an empirical argument that the homotopy method should converge to the closest root more often than the Newton-Raphson method.

Dimensioning a complex mechanical part involves hundreds or thousands of geometric constraints, that is why qualitative analysis of the system of constraints plays an essential role in preparing the resolution [21,15,20,24]. Today this analysis seems to be graph-based for all industrial solvers, as far as it is possible to know.

Graph-based methods develop some kinds of graph representing the system of constraints; they compute the so called degrees of freedom in this graph and its subgraphs. Technically, graph-based methods compute maximum matching [23,1,10], or maximum flows [13,12], or $k$-connected components [17,20,21]. These methods are polynomial time; they work very well for correct systems of constraints, *i.e.* when constraints are independent. Indeed, graph-based methods allow to solve systems of constraints which could not be solved otherwise. Graph-based methods can also detect the simplest dependences between constraints, called structural dependences which typically occur when a subset of unknowns is constrained by too much constraints, as in the system $f(x, y, z) = g(z) = h(z) = 0$ which over constrains $z$.

It is essential to detect dependences because numeric solvers typically fail, or get bogged down, when they are used to solve systems which are "wrongly" assumed to be well-constrained (to have a finite number of roots modulo the group of isometries). Moreover they do not give any useful explanation to help the users fix the problem. However, when the system of constraints is available without any further details, no polynomial time method can detect all dependences. Non structural dependences, which are due to geometric theorems, are not detectable by the previous methods. These dependences can occur in the seemingly simplest geometric constraints such as point-line incidences in 2D (in the projective plane, more technically). In CAD CAM, the major part of systems of geometric constraints involve such incidence constraints, *i.e.* incidence relations between points, lines, planes, circles/spheres, conics/quadrics. Of course, other constraints are also used to specify angles and lengths for dimensioning; these metric constraints involve parameters (values for lengths and angles) with generic values. Incidence constraints are especially relevant; Section 3 shows that detecting non structural dependences just amongst point-line incidences in 2D is as difficult as the ideal or radical membership problem of computer algebra. This problem is decidable with standard bases (also called Grobner bases) computable with Buchberger's method [25,4]. But no method to solve it is practicable, *i.e.* none scales to problems with industrial size.

This difficulty explains why the GCS (Geometric Constraints Solving) community usually assumes that constraints are either independent, or structurally dependent. This inability to detect and treat non structural dependences clearly restricts the use of GCS. This paper shows that at least for CAD CAM problems, an alternative method is indeed able to detect all dependences, structural

or non structural, in polynomial time, assuming that a witness (to be defined below in Section 2) is available. Moreover, this method can also be used to decompose a well-constrained system into smaller well constrained parts, also in polynomial time. Here, well-constrained means: has a finite number of solutions modulo some group; the most relevant group for dimensioning a part in CAD CAM is the group of isometries: all compositions of translations, rotations and symmetries. For short, a system or a subsystem which is "well constrained modulo the isometry group" is said to be rigid in the GCS community (the meaning of "rigid" is a bit different in the rigidity theory community which has inspired several graph-based methods used in GCS).

Section 2 defines the witness configuration, discusses the witness computation, and presents the probabilistic graph rigidity test of which the witness method is an offspring. Section 3 explains why detecting all dependences between constraints is as hard as the ideal or radical membership problem. Section 4 explains the rigidity test of the witness method. Section 5 gives the proof that the witness method detects all dependences, structural as well as non structural. Section 6 presents a possible method to decompose a rigid systems into rigid subsystems. Section 7 concludes.

## 2 The Witness Method

### 2.1 The Witness: Definition and Computation

A witness is defined as follows: let $F(U, X) = 0$ be the system to be solved; $X$ is the vector of unknowns, and $U$ is the vector of parameters (lengths, cosines, or sines, or tangents of angles, etc) or non geometric parameters (Young elasticity modules, weights, costs, densities, temperatures, forces, etc) [19]. By definition, the values of parameters are known just before the resolution. The goal is to find the roots $X_T$ of the target system: $F(U_T, X_T) = 0$, where $U_T$ are the specified values for the parameters $U$. Thus the target is a couple $(U_T, X_T)$ so that $F(U_T, X_T) = 0$. A witness is just another couple $(U_W, X_W)$ such that $F(U_W, X_W) = 0$ as well. Usually, $U_W$ and $U_T$ are different, so a witness root is likely not a target root; nevertheless, the witness and the target share essential combinatorial properties *e.g.* they share the same jacobian rank and structure. Actually, the witness method assumes that the target and the witness have the same combinatorial properties, in other words, only their numeric values are different. This is a probabilistic assumption in the following sense: among all possible witnesses, *i.e.* solutions of the system $F(U, X) = 0$, the set of wrong witnesses has measure (or probability) zero. A witness is wrong when its combinatorial properties (rank and structure of its jacobian) are different from those of the target. The principle of the witness method is then straightforward: it studies the combinatorial properties at the witness and transfers them to the target.

In CAD CAM, a witness is usually available, or easy to find; often the sketch is a possible witness. A witness is a configuration which fulfils all "constraints without parameters". These constraints are projective constraints of linear incidences

(collinearities, coplanarities) or non linear incidences ("co-conicity" of 6 points in 2D), parallelisms or orthogonalities (or other non generic angles). In passing, parallelisms and orthogonalities (and non generic angles) can all be replaced by projective constraints, see Fig. 1, 2, and 3 for an intuitive account. The set of possible witnesses is very large: typically it is a continuum with dimension $|U|$ where $|U|$ is the number of parameters. This explains why, in the CAD CAM context it is generally easy to find a witness: the sketch, or the solution to a previously solved system is usually a good witness. However, the witness computation may also be arbitrarily difficult, We summarize here some cases where the difficulty is known. For the molecule problem (given some inter atomic distances, find the configuration of the molecule) [9,10,6], finding a witness is completely trivial: just generate random points in 2D or in 3D according to the nature of the problem. For Eulerian polyhedra with specified coplanarities constraints (metric constraints such as angles between planes, distances between points, are dismissed), finding a witness is cubic time; it results from a constructive proof of Steinitz's theorem [22], which states that each Eulerian polyhedron is realizable in $\mathbb{Z}^3$ with some convex polyhedron, *i.e.* all vertices coordinates are integers (Steinitz's property is remarkable, since it does not hold in 4D [22]). Fewer details are known for non Eulerian polyhedra, *i.e.* with one or several handles.

For general geometric constraints (including incidences), we can often use a dual method for generating a witness, for instance when the unknown configuration is a 3D polyhedron described by the length of its edges. The octahedron problem, solved by Durand and Hoffmann [7], also known as the Stewart platform, and the icosahedron problem are just molecule problems: they have triangular faces, so generating random vertices is sufficient; the fact that the generated polyhedron is likely concave and even self intersecting does not matter as far as the distance and coplanarity conditions are satisfied. The hexahedron (6 quadrangular planar faces) or the dodecahedron (12 pentagonal planar faces) are examples of systems of geometric constraints where the dual method for generating the witness works: generate random planes in 3D, one random plane per face, before computing the resulting vertices as intersection points of the supporting planes. This dual method clearly relies on the fact that each vertex of the hexahedron and of the dodecahedron is degree 3. For vertices with greater degree, the method will not work because there is a null probability for four (or more) random planes to meet in a common point.

### 2.2 A Forerunner of the Witness Method

The witness method extends a probabilistic test used in rigidity theory [9]. Rigidity theory searches a combinatorial characterization for the rigidity of graphs: for a given dimension, does a non oriented graph with edges labelled with generic lengths has a rigid realization in $d$ dimensions? For instance two triangles sharing a common edge are a rigid graph in 2D, but not in 3D where they can fold along their common edge. The genericity assumption forbids collinearities, coplanarities in 3D, and non linear incidences (points on conics, or quadrics) which are essential in CAD CAM.

The characterization of graph rigidity is known in 2D, after Laman's theorem: a graph with $v \geq 2$ vertices and $e$ edges is minimally rigid (or isostatic: it is rigid, and removing one edge makes it flexible) iff $e = 2v - 3$ and if for all subgraphs induced by $v'$ vertices and having $e'$ edges, $e' \leq 2v' - 3$. There is an exponential number of subgraphs, but several polynomial time algorithms have been proposed to test graph rigidity [18,10].

The intuitive extension to 3D of Laman's theorem (where $2v - 3$ is replaced by $3v - 6$, and $2v' - 3$ is replaced by $3v' - 6$) is unfortunately wrong; the double banana is the most famous example. Up to now, the combinatorial characterization of graph rigidity in 3D and beyond is unknown. For GCS, it is convenient to extend Laman's theorem to other kinds of constraints in 2D, and in 3D. It gives an *approximate* but essential characterization of rigidity on which graph-based decomposition methods rely.

Though the combinatorial characterization of graph rigidity is unknown in 3D and beyond, there is a probabilistic and polynomial time algorithm to decide the rigidity of a given graph for any given dimension. It relies on Gluck's theorem: a graph is rigid if a generic realization of it is. So compute the rank of the jacobian (called the rigidity matrix) for a random realization. The witness method is an offspring of this rigidity test; in order to account for any kind of constraints (and not only point-point distances), the realization can no more be random: a random realization has probability 0 to fulfil "constraints without parameters": parallelism, orthogonalities, incidences (collinearities, coplanarities).

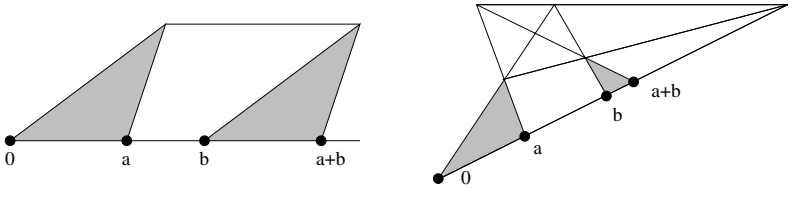


**Fig. 1.** Affine and projective construction of $a + b$. In the affine construction (left), the two shaded triangles are congruent and have parallel sides.

From a theoretical viewpoint, the witness method brings nothing new: it does not give a combinatorial characterization for well-constrainedness modulo the isometries group. It is a rather straightforward extension of the probabilistic test for graph rigidity which relies on Gluck theorem.

## 3 Why Detecting All Dependences Is Difficult

This section explains why a polynomial time method can not detect all dependences in seemingly simple systems of constraints containing only point-line incidences in 2D.

All systems of algebraic equations with coefficients in $\mathbb{Z}$ reduce, in polynomial time, to a system of point-line incidences in the projective plane [5,2]. The idea

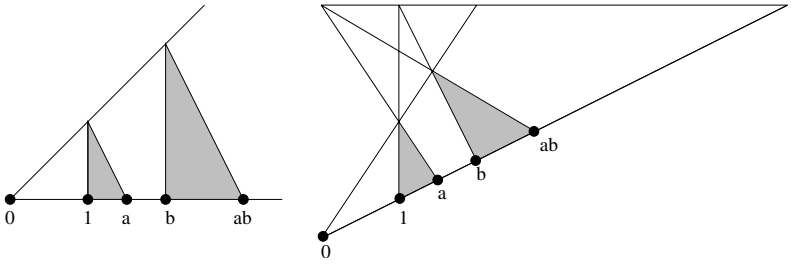


**Fig. 2.** Affine and projective construction of $a \times b$. In the affine construction, the two shaded triangles are similar (proportional) and have parallel sides.

is to represent each number (coefficient or root) by a point on an arbitrary line; this line passes through two arbitrary distinct points, representing the number 0 and the number 1 (later, the projective construction will need a third arbitrary point on this line, the point at infinity ). Then a first geometric construction in 2D constructs the point representing $a + b$ from the point representing $a$ and the point representing $b$ , see Fig. 1. A second geometric construction constructs the point representing $a \times b$ from the points representing $a$ and representing $b$, see Fig. 2. Actually, a construction in affine geometry is first proposed; it uses parallelism constraints, which are removed in the projective construction, using the classical idea of projective geometry: parallel lines are replaced by lines concurrent to a special arbitrary line, the Desargues line at infinity. It is a classical result that, if the projective plane satisfies Desargues and Pappus properties, then these two geometric constructions indeed define a field of numbers, *i.e.* associativity, commutativity, distributivity, etc hold; for instance the fact that the point representing $a \times b$ is equal to the point $b \times a$ relies on Pappus property of the projective plane [5].

These two constructions permit to translate the algebraic system into a set of point-line incidences. Remark that the ruler construction of an integer coefficient $n$ of the system of equations needs $O(\log_2 n)$ incidence constraints, using iterated

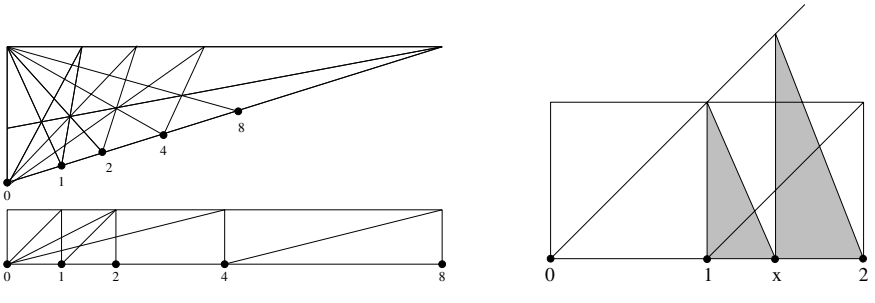


**Fig. 3.** Left: geometric construction of powers of 2. Right: affine constraints equivalent to the equation $x^2 - 2 = 0$.

squaring and additions (Fig. 3), thus it has the same size as the usual binary representation of integers. The right part of Fig. 3 shows the set of incidence and parallelism constraints for the equation $x^2 - 2 = 0$.

Thus solving algebraic systems and solving point-line incidences constraints are in principle the same problem. So detecting dependences in point-line incidences constraints is as difficult as detecting dependences in algebraic systems. The latter problem is equivalent to solving the ideal membership problem, and the radical membership problem (more on this question in section 3). Both problems are decidable, for instance with standard bases computable with Buchberger's algorithm. Unfortunately, due to the high algorithmic complexity of the problem, these methods are practicable only for small instances. Another direct consequence is that, in principle, finding a witness can be arbitrarily difficult: it suffices to translate a difficult system of equations (*e.g.* arising from a system of geometric constraints with specified numerical values for parameters) into a system of point-line incidences in 2D.

## 4 The Study of a Witness

This section details the study of the witness. The main idea is to compute the structure of the jacobian of the system at the witness. For example, if the jacobian has full rank, then the witness system contains no dependence, and this property is transferred to the target system.

The kernel of the jacobian at the witness is a vector space of the so called infinitesimal motions. Classically, there are two kinds of motions: displacements (also called rigid motions, or isometries), and flexions. Displacements are compositions of translations, symmetries, rotations; they constitute the isometry group; they do not alter distances and angles. On the contrary, flexions do – at least in the generic case (degenerate cases are not considered in this paper, for concision).

### 4.1 Computing a Basis of Infinitesimal Displacements

A system of geometric constraints is rigid iff the kernel of its jacobian contains only displacements. It is possible to compute an *a priori* basis of the infinitesimal displacements. 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 2 points). $q_j$ represents an unknown which is independent of the cartesian frame: it is either a geometric unknown such as a length, a radius, an area, a scalar product, or a non geometric unknown. Dotted variables $\dot{x_i}, \dot{y_i}, \dot{a_l}, \dot{b_l}, \dot{c_l}, \dot{u_k}, \dot{v_k}$ and $\dot{q_j}$ are used to denote the values of the corresponding coordinates in the basis of infinitesimal displacements, *e.g.* the couple

**Table 1.** A basis for the free displacements in 2D for points, lines, and vectors

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

$(\dot{x_i}, \dot{y_i})$ representing the infinitesimal translation $t_x$ along the $x$ axis of a point $(x_i, y_i)$ is equal to $(1, 0)$. Note that the infinitesimal displacements for a point $(x, y)$, a normal $(a, b)$ to a line, and a vector $(u, v)$ are different; *e.g.* translating a point modify it, but translating a vector or a normal does not.

Table 2 shows a possible basis for the infinitesimal displacements in 3D; it contains three translations $t_x$, $t_y$, and $t_z$, and three rotations $r_{xy}, r_{xz}$ and $r_{yz}$. Points have coordinates $(x, y, z)$, planes have coordinates $(a, b, c, d)$ (their equation is: $ax + by + cz + d = 0$), vectors have coordinates $(u, v, w)$; $q$ represents an unknown independent of the cartesian frame.

**Table 2.** A basis for the free displacements in 3D for points, planes, vectors, and unknowns independent of the cartesian frame

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

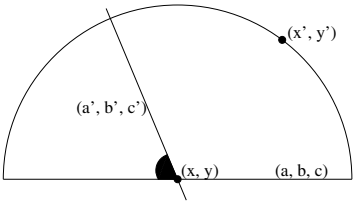


**Fig. 4.** A 2D under-constrained system of geometric constraints

### 4.2 A Structurally Under-Constrained Example in 2D

Fig. 4 shows a simple under-constrained example in 2D. A possible witness is $(x = y = 0, x' = 3, y' = 4, \delta = 5, a = 1, b = 0, a' = 12/13, b' = 5/13,$ and $\lambda = 12/13)$. All graph-based methods give correct results when considering this system,

**Table 3.** The jacobian and a basis of infinitesimal motions: three displacements and a flexion for the system given in (1)

| | $x$ | $y$ | $x'$ | $y'$ | $a$ | $b$ | $c$ | $a'$ | $b'$ | $c'$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $e'_1$ | $a$ | $b$ | 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 |

because the under-constrainedness is structural. This system is composed of the following six equations:

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

The jacobian and a basis of its kernel are given Table 3 where all symbols are replaced by their values at the witness. This basis contains the 3 displacements of the plane, plus a flexion vector: indeed point $(x', y')$ can rotate around point $(x, y)$. If columns $x', y'$ are removed, the flexion vector clearly becomes, in the remaining columns: $x, y, a, b, c, a', b', c'$, a linear combination of the 3 displacement vectors (the fact that it vanishes is basis dependent; the fact that it is a linear combination is not). This shows that the part obtained after the removal of the point $(x', y')$ is rigid. This is the test that the witness method uses to decide the rigidity of a part.

### 4.3 A Dependence due to a Theorem

Let us now consider the 2D system of geometric constraints of Fig. 5. This system is structurally correct, but contains a non structural dependence due to a (simple) geometric theorem, so it will be difficult (at least) for graph-based methods to detect this dependence, but the witness method detects it.

Constraints are as follows; point $O$ is the middle of $AB$; distance $OC$ equals distance $OA$; the angle between $AC$ and $BC$ is right. Actually this constraint is

a consequence of the other constraints. Finally distance $OA$ is specified (just to have the "good" number of constraints). The system of equations is:

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

A possible witness is: $O = (0,0), A = (-10,0), B = (10,0), C = (6,8), u = 10$. The jacobian and a basis of the free infinitesimal motions (three displacements and a flexion: point $C$ can rotate around point $O$) are given in Table 4, where again, all symbols are replaced by their value at the witness. The rank of $e'_1, \ldots, e'_5$ computed at the witness is 4, thus equations are dependent.

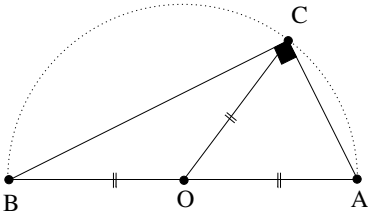


**Fig. 5.** Example of dependent constraints

**Table 4.** The jacobian, and a basis of 4 free infinitesimal motions for the dependent system given in (2). The fourth motion is a flexion: point $C$ can rotate around $O$.

| | $x_O$ | $y_O$ | $x_A$ | $y_A$ | $x_B$ | $y_B$ | $x_C$ | $y_C$ |
|---|---|---|---|---|---|---|---|---|
| $e'_1$ | 2 | 0 | $-1$ | 0 | $-1$ | 0 | 0 | 0 |
| $e'_2$ | 0 | 2 | 0 | $-1$ | 0 | $-1$ | 0 | 0 |
| $e'_3$ | $2x_A - 2x_C$ | $2y_A - 2y_C$ | $2x_O - 2x_A$ | $2y_O - 2y_A$ | 0 | 0 | $2x_C - 2x_O$ | $2y_C - 2y_O$ |
| $e'_4$ | 0 | 0 | $x_B - x_A$ | $y_B - y_C$ | $x_A - x_C$ | $y_A - y_C$ | $2x_C - x_A - x_B$ | $2y_C - y_A - y_B$ |
| $e'_5$ | $2x_O - 2x_A$ | $2y_O - 2y_A$ | $2x_A - 2x_O$ | $2y_A - 2y_O$ | 0 | 0 | 0 | 0 |

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

### 4.4 Computing Degrees of Displacements

In an attempt to make graph-based methods more robust against non structural dependences, Jermann introduced the notion of DoD, Degrees of Displacements, in his PhD thesis (actually he called that the Degree of Rigidity) [16,14]. Consider

a subset of unknowns $Y \subset X$; extract the columns of $Y$ in the basis of the infinitesimal displacements; call it $D[Y]$; then the DoD of $Y$ is the rank of this subarray $D[Y]$. For instance, for a 2D line $Y = (a, b, c)$, the subarray $D[Y]$ is: $(\dot{a},\ \dot{b},\ \dot{c}) = (0,\ 0,\ -a)$ for the translation $t_x$, $(0,\ 0,\ -b)$ for the translation $t_y$, and $(-b,\ a,\ 0)$ for the rotation $r_{xy}$. The line has 2 DoD and the 2 translations are dependent. Similarly, $M[Y]$ will denote the content of the columns $Y$ in any basis of the free infinitesimal motions (displacements, and flexions) of the system at the witness.

Jermann understood that graph-based methods are fragile because they can not compute exact DoDs. It turns out that DoD is not only a syntactical or structural idea when accounting for incidence constraints. For instance, the DoD of two secant planes in 3D is 5 (more precisely the 3 rotations are independent, but the 3 translations are dependent; intuitively, and *a posteriori*, it is understandable, since translating the 2 planes along their intersection line leave them invariant) while the DoD of two parallel planes is 4. Similarly, the DoD of 3 collinear points is 2, while the DoD of 3 non collinear points is 3. A pure graph-based method has no mean to know if 3 points are collinear or not, or if two planes are parallel or not. Either it assumes the configuration is generic (and thus has maximal DoD), or it can try to look if the parallelism/collinearity is an explicit constraint of the system; but it may happen that the parallelism/collinearity is a remote consequence of a set of constraints, thanks to Desargues, or Pappus, or Pascal, or Miquel theorems: the incidence in the conclusion is a non trivial consequence of the hypothesis.

The witness method avoids this difficulty: it just computes the DoD of the part, at the witness, with standard method from linear algebra. If a parallelism/collinearity or any other feature holds because of some constraints and geometric theorems, then it holds in the witness.

A part with full DoD (3 in 2D, 6 in 3D) and with minimal cardinality: 3 in 2D, 6 in 3D, is called an anchor. Section 6 uses anchors for decomposing.

### 4.5 Interrogating a Witness

Geometric constraints are independent of the cartesian frame. But sometimes, some constraints such as: $x_1 = y_1 = y_2 = 0$ are used to "pin" the configuration in the plane and to make the system of equations well-constrained, which simplifies the work of the numerical solver, *e.g.* Newton-Raphson. The following test checks that a constraint is independent of the cartesian frame: it is iff its gradient vector is orthogonal to all basis vectors in the basis of infinitesimal displacements. From now on, constraints are assumed to be independent of the cartesian frame.

Are constraints independent? They are iff the jacobian at the witness has full rank. Is a part $Y$ rigid, *i.e.* well constrained modulo displacements? It is iff $D[Y] \equiv M[Y]$: the vector space of its infinitesimal motions is equal to the vector space of its infinitesimal displacements. In other words, since $D[Y] \subset M[Y]$, the rank of $D[Y]$ is equal to the rank of $M[Y]$.

### 4.6 Rank Computations Are All What You Need

All computations of ranks are performed on vectors with numerical entries provided by the coordinates of the witness and the gradient vectors at the witness.

Inaccuracy issue is detailed elsewhere. We just mention that, if the witness has rational coordinates, then all computations can be performed exactly, with the classical Gauss pivoting method; it is also possible to compute exactly modulo a prime close to $10^9$, which introduces another source of probabilisticity. Finally, if the witness method is used only to check the independence of vectors, then an interval arithmetic is sufficient, in the following sense: if a set of vectors is independent, then interval computations can guarantee it (assuming intervals are sharp enough); if vectors are dependent, then interval computations can not prove it. Note that when the interval analysis can not prove the independence of the gradient vectors at the witness, either vectors are dependent or the system at the witness is ill-conditioned.

## 5 The Proof That the Witness Method Detects All Dependences

An algebraic equation $g(x) = 0$ is a consequence of the other equations: $f_1(x) = f_2(x) = \ldots f_n(x_n) = 0$ in two cases: either $g$ lies in the ideal generated by the polynomials $f_1, \ldots f_n$, or $g$ lies in the radical generated by the polynomials $f_1, \ldots f_n$. This section proves that in both cases the witness method detects the dependence.

Assume first that $g$ lies in the ideal of the polynomials $f_i, i = 1, \ldots n$. Then by definition there are polynomials $\lambda_1(x), \ldots \lambda_n(x)$ such that $g(x) = \lambda_1(x) \times f_1(x) + \lambda_2(x) \times f_2(x) + \ldots \lambda_n(x) \times f_n(x)$. A first consequence is that $g$ vanishes at a common root of the polynomials $f_i, i = 1, \ldots n$. A second consequence is obtained by deriving the previous equality: $\nabla g(x) = \nabla\lambda_1(x)f_1(x) + \lambda_1(x)\nabla f_1(x) + \ldots \nabla\lambda_n(x)f_n(x) + \lambda_n(x)\nabla f_n(x)$. At a common root $w$ of the $f_i$ polynomials, such as the witness, terms $f_i(w)$ vanish and it yields $\nabla g(w) = \lambda_1(w)\nabla f_1(w) + \ldots \lambda_n(w)\nabla f_n(w)$. In other words the gradient vector of $g(w)$ is a linear combination of the gradient vectors $f_1(w), \ldots f_n(w)$. But the witness method detects such dependences between the gradient vectors when it studies the jacobian at the witness $w$.

Assume now that $g$ does not lie in the ideal, but in the radical of $f_1, \ldots f_n$. By definition there is an integer $k > 1$ and polynomials $\lambda_1(x), \ldots \lambda_n(x)$ such that $g(x)^k = \lambda_1(x) \times f_1(x) + \lambda_2(x) \times f_2(x) + \ldots \lambda_n(x) \times f_n(x)$. A first consequence is that $g$ vanishes at a common root of the polynomials $f_i, i = 1, \ldots n$. A second consequence is that, by derivation: $kg(x)^{k-1}\nabla g(x) = \nabla\lambda_1(x)f_1(x) + \lambda_1(x)\nabla f_1(x) + \ldots \nabla\lambda_n(x)f_n(x) + \lambda_n(x)\nabla f_n(x)$. At a common root $w$ of the $f_i$ polynomials,terms $f_i(w)$ vanish and it yields $kg(w)^{k-1}\nabla g(w) = 0 = \lambda_1(w)\nabla f_1(w) + \ldots \lambda_n(w)\nabla f_n(w)$. It means that the gradient vectors of $f_1, \ldots f_n$ at $w$ are dependent (and $g$ does not matter in this case). But the witness method detects such dependences between the gradient vectors when it studies the jacobian at the witness $w$. Thus, in

both cases, the witness method detects a dependence between the gradient vectors of the equations at the witness. The previous proof applies to algebraic systems. Maybe it is possible to extend it to non algebraic equations (involving transcendentals) with some topological argument.

## 6 The Witness Method Decomposes into Rigid Parts

Every graph-based method for decomposing the system of geometric constraints and assembling the solutions of the parts, is two-folds: on one hand it proposes a strategy to decompose the system, and on the other hand it proposes a test to decide if a given part is well-, over-, or under-constrained modulo the isometries group. We saw that, contrary to the witness method, the test can be confused with some configurations and fail to detect some dependences when they are not structural. But the strategy can still be used. Thus for each graph-based method proposed so far to plan the resolution process, it is possible to keep its strategy, and to replace its rigidity test with the one provided by the witness method.

This section explains one of the possible strategies to decompose a rigid system into rigid subsystems, maybe the simplest strategy. If the system is flexible, its Maximal Rigid Parts (MRP) are computed. If it is rigid, each constraint is removed in turn; it provides a flexible system, the MRP of which are computed.

To find the MRP of a flexible system, its anchors are first determined; an anchor is a subset $Y$ of $d(d+1)/2$ unknowns which has full DoD (3 in 2D, 6 in 3D) and which is rigid, *i.e.* the vector space of its infinitesimal motions $M[Y]$ is equal to the vector space of its infinitesimal displacements $D[Y]$. Clearly there is a polynomial number of potential anchors; just test the rigidity of each potential anchor. Every anchor $Y$ belongs to exactly one MRP, noted MRP($Y$). MRP($Y$) is computed with the obvious greedy method: initialize MRP($Y$) with $Y$, consider every variable $x \in X - Y$ (in any order) and insert it in MRP($Y$) iff $Y \cup \{x\}$ is still rigid, *i.e.* iff $M[Y \cup \{x\}] \equiv D[Y \cup \{x\}]$.

Some book-keeping may speed up the method, and avoid to find several times the same maximal rigid parts. However the method is polynomial even without such optimizations: indeed there is a polynomial number of potential anchors, and each anchor is contained in a single MRP.

## 7 Conclusion

This paper has shown that the witness method detects all dependences: structural dependences which are already detected by graph-based methods, but also non structural dependences which are due to known or unknown geometric theorems, and may occur with incidence constraints. The witness method can also decompose a rigid system into rigid subsystems; actually it is possible to reuse the strategic part of every graph-based method proposed so far to decompose rigid systems into rigid irreducible parts with the rigidity test provided by the witness method. In practice, the witness method should widen the scope of geometric constraints solving.

## References


1. Ait-Aoudia, S., Jegou, R., Michelucci, D.: Reduction of constraint systems. In: Compugraphic, Alvor, Portugal, pp. 83–92 (1993)
2. Bonin, J.E.: Introduction to matroid theory. The George Washington University web site (2001)
3. Bruderlin, B., Roller, D. (eds.): Geometric Constraint Solving and Applications. Springer, Heidelberg (1998)
4. Cox, D., Little, J., O' Shea, D.: Ideals, varieties and algorithms, 2nd edn. Springer, Heidelberg (1996)
5. Coxeter, H.: Projective geometry. Springer, Heidelberg (1987)
6. Crippen, G.M., Havel, T.F.: Distance Geometry and Molecular Conformation, Taunton, U.K. Research Studies Press (1988) ISBN 0-86380-073-4
7. Durand, C., Hoffmann, C.M.: Variational constraints in 3D. In: Shape Modeling International, pp. 90–97 (1999)
8. Gao, X.-S., Zhang, G.: Geometric constraint solving via c-tree decomposition. In: ACM Solid Modelling, pp. 45–55. ACM Press, New York (2003)
9. Graver, J.E., Servatius, B., Servatius, H.: Combinatorial Rigidity. Graduate Studies in Math., AMS (1993)
10. Hendrickson, B.: Conditions for unique graph realizations. SIAM J. Comput. 21(1), 65–84 (1992)
11. Hoffmann, C.M.: Summary of basic 2D constraint solving. International Journal of Product Lifecycle Management 1(2), 143–149 (2006)
12. Hoffmann, C.M., Lomonosov, A., Sitharam, M.: Decomposition plans for geometric constraint problems, part II: New algorithms. J. Symbolic Computation 31, 409–427 (2001)
13. Hoffmann, C.M., Lomonosov, A., Sitharam, M.: Decomposition plans for geometric constraint systems, part I: Performance measures for CAD. J. Symbolic Computation 31, 367–408 (2001)
14. Jermann, C., Neveu, B., Trombettoni, G.: A new structural rigidity for geometric constraint systems. In: Winkler, F. (ed.) ADG 2002. LNCS (LNAI), vol. 2930, pp. 87–106. Springer, Heidelberg (2004)
15. Jermann, C., Trombettoni, G., Neveu, B., Mathis, P.: Decomposition of geometric constraint systems: a survey. Internation Journal of Computational Geometry and Applications (IJCGA) (to appear 2006)
16. Jermann, C., Trombettoni, G., Neveu, B., Rueher, M.: A new heuristic to identify rigid clusters. In: Richter-Gebert, J., Wang, D. (eds.) ADG 2000. LNCS (LNAI), vol. 2061, pp. 2–6. Springer, Heidelberg (2001)
17. Latham, R.S., Middletich, A.E.: Connectivity analysis: a tool for processing geometric constraints. Computer-Aided Design 28(11), 917–928 (1996)
18. Lovasz, L., Yemini, Y.: On generic rigidity in the plane. SIAM J. Algebraic Discrete Meth. 3(1), 91–98 (1982)
19. Michelucci, D., Foufou, S.: Geometric constraint solving: the witness configuration method. Computer Aided Design 38(4), 284–299 (2006)
20. Owen, J.C.: Algebraic solution for geometry from dimensional constraints. In: Proc. of the Symp. on Solid Modeling Foundations and CAD/CAM Applications, pp. 397–407 (1991)
21. Owen, J.C.: Constraint on simple geometry in two and three dimensions. Int. J. Comput. Geometry Appl. 6(4), 421–434 (1996)

22. Richter-Gebert, J.: Realization Spaces of Polytopes. LNCM, vol. 1643. Springer, Heidelberg (1996)
23. Serrano, D.: Automatic dimensioning in design for manufacturing. In: Sympos. on Solid Modeling Foundations and CAD/CAM Applications, pp. 379–386 (1991)
24. Sitharam, M.: Combinatorial approaches to geometric constraint solving: Problems, progress and directions. In: Dutta, D., Janardan, R., Smid, M. (eds.) AMS/DIMACS. Computer aided design and manufacturing (2005)
25. Wester, M.J.: Contents of Computer Algebra Systems: A Practical Guide. John Wiley and Sons, Chichester (1999)