Automated Geometry Diagram Construction and Engineering Geometry¹⁾

Xiao-Shan Gao

Institute of Systems Sciences, Academia Sinica

Beijing 100080, P.R. China

e-mail: xgao@mmrc.iss.ac.cn

Abstract. This paper reviews and evaluates three main techniques for automated geometry diagram construction: synthetic methods, numerical computation methods, and symbolic computation methods. We also show how to use these techniques to: parametric mechanical CAD, linkage design, computer vision, dynamic geometry, and CAI (computer aided instruction). The methods and the applications reviewed in this paper are closely connected and could be appropriately named as *engineering geometry*.

1. Introduction

In the area of geometry automated reasoning, most of the efforts have been concentrated on automated geometry theorem proving and discovering. These efforts lead to the classic work of Tarski [?], Gerlertner [?], and Wu [?]. On the other hand, automated geometry diagram construction (AGDC) is more or less overlooked. Actually, automated geometry diagram construction has been studied in the CAD community under a different name: geometric constraint solving (GCS) and with a different perspective: engineering diagram drawing. GCS is the central topic in much of the current work of developing intelligent or parametric CAD systems [?, ?, ?, ?, ?, ?, ?, ?] and interactive constraint-based graphic systems [?, ?, ?]. The main advantage of using the constraint approach is that the resulting systems accept declarative descriptions of diagrams or engineering drawings, while for conventional CAD systems the users need to specify how to draw the diagrams. As a result, the parametric systems are more powerful and user friendly.

Three main approaches have been developed for automated diagram construction: the synthetic approach, the numerical approach, and the symbolic approach.

In the *synthetic approach*, a pre-treatment is carried out to transform the constraint problem into a step-by-step constructive form which is easy to draw. A majority of the work is to transform the constraint problem to a constructive form with ruler and compass as construction tools. Once a constraint diagram is transformed into constructive form, all its solutions can be computed efficiently. Two main techniques used in the synthetic approach are the *rule-based search* and the *graph analysis*. The basic idea of the rule-based search [?, ?, ?, ?, ?] is to represent the geometric construction knowledge as deduction rules and use various search techniques from AI to find a step-by-step construction procedure for a

¹⁾This work was supported in part by an Outstanding Youth Grant from the Chinese NSF and the National Climbing Project.

constrained diagram. The basic idea of the graph analysis [?, ?, ?, ?, ?] is to represent the geometric conditions as a graph and use various techniques from graph theory to decompose the graph into smaller parts and the construction procedure for the diagram is to assemble the small parts into larger ones.

In the *numerical approach*, geometric constraints are translated into algebraic equations and various numerical techniques are used to solve these equations. The first generation constraint-based systems, such as Sketchpad [?] and ThingLab [?], used numerical relaxation as the last resort of solving constraint problems. Newton iteration methods were used in Juno-2 [?] and in the Variational Geometry [?]. The main advantage of numerical methods is their fast speed and generality. On the other hand, it also needs further improvements in several aspects: like finding stable solutions during animation and finding all solutions of equation systems.

In the *symbolic approach* [?, ?, ?, ?], we also transform the geometric constraints into algebraic equations. Instead of using numerical methods to solve the algebraic equations directly, in the symbolic approach, we first use general symbolic methods such as Wu-Ritt's characteristic method [?], the Groebner basis method [?], the Dixon Resultant method [?], and Collins' cylindrical algebraic decomposition method [?] to change the equation set to new forms which are easy to solve, and then solve the new equations numerically. The advantage of the symbolic computational approach is that it can provide complete methods for many problems in AGDC, such as detecting ruler and compass construction, the well-constraintness and conflict constraints. On the other hand, current symbolic methods are still too slow for real time computation.

The above AGDC techniques can be used to a variety of engineering problems: parametric mechanical design, linkage design, robotics, computer vision, and geometric modeling. They can also be used to: dynamic geometry, physics, and CAI (computer aided instruction). All these applications have the same character: they can be reduced to AGDC problems and can be solved with the three methods introduced above. We call the methods and the applications reviewed in this paper *engineering geometry*.

This paper is an attempt to give an overview of the research on engineering geometry with emphasis on the work done at Mathematics Mechanization Research Center (MMRC) in Beijing and Chou's group in Wichita State University. In the next section, we introduce two synthetic approaches: the global propagation [?] and the graph decomposition approach of Owen [?]. In Section 3, we introduce the application of the numerical optimization methods to AGDC [?]. In Section 4, we introduce the symbolic approach based on Wu-Ritt's characteristic method [?, ?, ?]. In Section 5, we discuss applications of these techniques.

2. Synthetic Approaches of AGDC

Synthetic approaches have many advantages. It can generate all solutions of a diagram efficiently and with high precision. The solutions provided by this kind of approaches are stable and have geometric meanings. So in most drawing systems, synthetic approaches are usually first used to solve the diagram construction problem. If synthetic approaches failed, other approaches will be used.

In this section, we will give a brief introduction to two synthetic approaches: the global propagation approach [?] and Owen's graph decomposition approach [?]. Other important

synthetic approaches include: Hoffmann's graph analysis approach [?], Kramer's freedom of degree analysis approach [?], Latheam and A.E. Middleditch's approach based on weighted graphs [?], Lee's optimized rule-based method [?], and other rule-based methods [?, ?].

2.1. The Global Propagation Method

Local Propagation is a basic AI technique to solve constraint problems [?, ?]. It is to find unknown quantities one by one from the set of known quantities. For instance, from the relation $9c - 5F = 160$ of two kinds of temperature c and F , we may find the following propagation rules:

1. If c is known, we may compute F with the formula: $F = \frac{9}{5}c + 32$.
2. If F is known, we may compute c with the formula: $c = \frac{5(F-32)}{9}$.

For AGDC, local propagation is closely related to the *method of locus intersection*. For instance, to draw a triangle ABC with sides $AB = 4$, $BC = 5$, and $CA = 6$, we first draw an arbitrary point as point A . To draw point B , we first draw a circle (locus) with center A and radius 4, and point B is an arbitrary point on this circle. To draw point C , we first draw two circles (loci) with centers A and B and radii 6 and 5 respectively, and C is the intersection of these two circles.

It is clear that the power of local propagation is limited. It can not solve constraint problems with *loops*, i.e., more than one geometric objects are constrained or referred simultaneously in the problem. *Global propagation* is proposed to solve constraint problems with loops [?]. Like the *local propagation method*, it tries to determine the position of a geometric object from the set of already drawn geometric objects. But the global propagation uses not only the constraints involving this object but also *implicit information* derived from other constraints. The global information needed in the propagation comes from a *geometry information base (GIB)* built before the construction begins. The GIB for a configuration is actually a database containing all the properties of the configuration that can be deduced using a fixed set of geometric axioms [?].

Fig. 1. An example to show the propagation of line direction

Fig. 1 is an example from [?]. Starting from point B , it is easy to construct B, C, X . Next, we want to construct D . Since $|XD|$ is known, we need only to know the direction of XD which can be determined by the following *global propagation of line directions*: $XD \perp DE$, $\angle(DEA) = a_2$, $AE \perp AB$, and $\angle(ABC) = a_1$.

Generally speaking, to determine the direction of a line L , we first use some *basic situations*:

1. L passes through two known points.
2. L cuts a fixed length between a pair of parallel lines.
3. L cuts a pair of segments on the sides of a known angle such that the ratio of the segments is known.
4. L cuts a pair of segments between three known lines passing through a point such that the ratio of the segments is known.

Second, any line with a known direction will be transformed to a line with a known direction by the following *transformations*: (1) parallel, (2) perpendicular, and (3) forming a known angle. So the direction of a line is usually determined by a sequence of lines

$$U_1V_1 \tau_1 U_2V_2 \tau_2 \dots \tau_{s-1} U_sV_s,$$

where U_i, V_i are points and τ_i are transformations. If the direction of U_sV_s is determined by one of the basic situations, the direction of line U_1V_1 is known. This process clearly demonstrates the global feature of the propagation.

An important technique in the global propagation is to use *LC transformation*. An LC transformation is a one-to-one map from the Euclidean plane to itself which transforms a line to a line and a circle to a circle. We mainly use two kinds of LC transformations.

The first LC transformation is as follows: suppose that segment AB has a fixed direction and a fixed length. Then point A is on a line or a circle if and only if point B is on a line or a circle. An example is given in Figure 2. Suppose that points A and C have been constructed. Next, we construct point B . Since $\angle ABC = a_1$, B is on circle C_1 . Similarly, point D is on a circle C_2 . Since BD has a fixed direction and a fixed length, point B must be on another circle C_3 , which is the transformation of circle C_2 . B is the intersection of circles C_1 and C_3 . To construct C_3 , we first construct the center X_1 of circle C_2 and then move it for a distance of $|BD|$ along the direction of BD to obtain the center X_2 of C_3 . The radius of C_3 is the same as that of C_2 .

Fig. 2. An example to show the first kind of LC transformation

The second LC transformation is related to ratio constraints. If points P, Q, R are collinear, PQ/RQ is known, and one of them, say Q , is already known, then point P is on a line or a circle if and only if point R is on a line or a circle. Consider the constraint problem in Figure 3. Suppose that we have constructed points A and B . Since $\angle(ACB) = a_1$, point C is on a circle C_1 . We construct the center X_2 of circle C_1 . Since M is the midpoint of BC ,

Fig. 3. An example to show the second kind of LC transformation

by the second LC transformation, point M is on a circle C_2 . Hence M is the intersection of circle C_2 and the circle with A as center and with d_2 as radius.

On the top level, the method works as follows [?]:

1. For a constraint problem

$$[[Q_1, \dots, Q_m], [P_1, \dots, P_n], [C_1, \dots, C_m]],$$

let $CT = \{C_1, \dots, C_m\}$ be the constraint set, $CS = \emptyset$ the construction sequence, $QS = \{Q_1, \dots, Q_m\}$ the points with given construction order, and $PS = \{P_1, \dots, P_n\}$ the remaining points. We assume that the problem is not over-constraint, i.e., we have $|CT| \leq 2 * |PS| - 3$.

2. Build the GIB as described in [?]. Then repeat the following steps first for QS and then for PS until both QS and PS become empty.

3. Take a point P from QS or PS . For each constraint $T_i \in CT$ involving P , decide the locus Lc of the points satisfying T_i , assuming all points constructed in CS are known. We then obtain a set of triples

$$\{(P, T_1, Lc_1), \dots, (P, T_s, Lc_s)\}.$$

We consider three cases. (1) $s = 0$. Point P might be an arbitrarily chosen (free) point. We add a new construction $C = (\text{POINT}, P)$ to CS . (2) There exist $i \neq j$ such that $T_i \neq T_j$ and Lc_i and Lc_j are not parallel lines or concentric circles.²⁾ We add a new construction $C = (\text{INTERSECTION}, P, Lc_i, Lc_j)$ to CS and remove T_i and T_j from CT . (3) Otherwise, point P is a semi-free point: it can move freely on a line or a circle. We add a new construction $C = (\text{ON}, P, Lc_1)$ to CS and remove T_1 from CT .

4. Now we check whether the remaining problem is over constraint, that is, whether $|CT| > 2 * |PS|$ (note this inequality is different from the one in Step 1). (1) If it is over constraint then the construction sequence is invalid in this case. If P is from QS , the order given by the user can not be constructed and the method terminates. Otherwise, restore the removed constraints and repeat the preceding step for a new point from PS . (2) If it is not over-constraint, point P is constructed. We need to repeat the preceding step for a new point.

²⁾We can check these conditions using automated theorem proving methods such as Wu's method[?, ?] or the area method[?] which are quite fast.

2.2. Owen's Graph Decomposition Method

A powerful tool to solve geometry constraint systems is graph theory. In this approach, a constraint problem is first transformed into a graph whose vertices and edges represent the geometric objects and constraints respectively. A constraint problem without loops can be represented as trees [?]. Owen introduced the first graph analysis method that can handle constraint problems with loops [?]. Basically speaking, Owen's algorithm has two steps:

1. *To decompose the graph into sub-graphs.* A graph is called *bi-connected* if it is connected and we cannot find a vertex such that by removing this vertex the graph becomes two graphs. A graph is called *tri-connected* if it is bi-connected and we cannot find two vertices such that by removing these two vertices and the edge (if there exists one) connecting them, this graph becomes two graphs. The decomposition procedure is first decomposing the graph into bi-connected graphs and then decomposing the bi-connected graph into tri-connected graphs using algorithms from graph theory [?]. At the final stage, every sub-graph must have less than three vertices. Otherwise, the algorithm cannot handle the problem.
2. *Assemble the sub-graphs into larger ones.* This step is the reverse of the first step and has two sub-cases for a well-constrained problem.
 - (a) If two vertices are removed during the decomposition and both vertices connected with the same unknown vertex V , then we can construct the geometric object represented by V in the next step. For instance, if the two known vertices represent lines and V represents a point P , then we will take the intersection of the two lines to obtain point P in the next step.
 - (b) If a graph is decomposed into three sub-graphs in such a way that each pair of sub-graphs have a common vertex, then we can assemble the three sub-graphs together. Since the problem is well-constrained, each sub-graph represents a rigid part. Let the common vertex representing, say, three points. Since each pair of the points is in a rigid part, we can compute the distance between them. Then the assemble problem becomes the construction of a triangle with three known sides.

Let us consider the problem in Figure 1. Figure 4 (a) is the graph of this constraint problem, where $l_1, l_2, l_3, l_4, l_5, l_6$ are lines (not segments) AB, BC, CX, XD, DE, EA respectively. We will solve this problem in two steps.

Fig. 4. Graph and graph decomposition for the diagram in Fig. 1

1. *To decompose the graph into sub-graphs.* We first decompose the graph into three sub-graphs by breaking the graph at vertices X , l_1 , l_6 . The sub-graph $l_1 l_6 A$ needs no further decomposition, since it contains three points only. The other two sub-graphs are similar. Figure 4 (b) gives the decomposition for one of them.

Fig. 5. Three sub-diagrams of the diagram in Fig. 1

2. *Assemble the sub-graphs into larger ones.* The three sub-graphs obtained in the first step of decomposition represent the three diagrams in Figure ??, which are easy to construct with ruler and compass. The critical step is to assemble the three parts into one. Let us note that the common vertices of two of the three sub-graphs are X , l_1 , and l_6 . Since each part is a rigid body, we know the distances from X to l_1 and l_6 , and the angle formed by l_1 and l_6 . Then we can easily draw a diagram containing X , l_1 and l_6 and put the three diagrams in Figure 5 to this diagram.

According to the above algorithm, we cannot solve the problem in Figure 2. But Owen's software can solve this problem geometrically.

3. The Optimization Method for AGDC

Numerical methods are usually used as the last resort in the constraint solving process, because they can be used to solve any kinds of equation systems. The most commonly-used method in the numerical approach is the Newton-Raphson method. It is fast, but has the instability problem: the method is sensitive to the initial values. A small deviation in the initial value may lead to an unexpected or unwanted solution, or to the iteration divergence. To overcome this problem, recently the homotopy method has been proposed

and experimented with [?]. According to the report in [?], generally the homotopy method works much better in terms of stability. These two methods generally require the number of variables to be the same as the number of equations.

In [?], a method based on the optimization techniques for solving geometric constraint problems is proposed. Experiments with this method show that it is also quite stable. Furthermore, the method can *naturally* deal with under- and over-constrained problems.

Generally, a geometric constraint problem can be first translated into a system of equations:

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\dots \\ f_m(x_1, \dots, x_n) &= 0. \end{aligned}$$

Then the problem is how to solve this equation system. The optimization approach solves the equation system by converting it into finding X at which the sum of squares

$$\sigma(X) = \sum_{i=1}^m f_i(X)^2$$

is minimal, where $X = (x_1, x_2, \dots, x_n)$ is the set of variables. It is obvious that the equation system has a *real* solution X^* if and only if $\min \sigma(X)$ is 0. The problem of solving a system of equations is thus converted into the problem of finding the minimum of a real multi-variate function. The problem now can be solved by various well-developed numerical optimization methods [?, ?].

One obvious fact for this approach is that the number of equations m is not necessarily the same as the number of variables n . Thus for this approach, it is natural to deal with under- and over-constrained problems.

Fig. 6. The insensitivity to the inexact initial guesses of the solutions in the optimization method.

An advantage of this method is that it is not sensitive to the initial value. This feature is demonstrated in Figure ?? where the three circles are specified to be mutually tangent and to be tangent to two neighboring sides of a triangle whose three vertices are specified to be fixed. Figure ??(a) and ??(c) are the initial diagrams sketched by the user. It is easy to see the differences between the initial guesses from the exact solutions of this problem respectively in Figure ??(b) and ??(d) are rather large. Also this figure demonstrates how different initial values lead to different branches of the solutions.

Fig. 7. A difficult problem. (a) is the initial diagram drawn by the user. (b) is the diagram generated after all tangent constraints are added

Table 1. Running statistics for the tangent packing problems

# circles (# rows)	# equations	# variables	time
3 (2)	30	30	0.228
6 (3)	54	54	0.965
10 (4)	86	86	3.379
15 (5)	126	126	11.587
21 (6)	174	174	23.751

Figure ?? is the simplest case for the problem we call *the tangent packing problem*. The problem is to pack $n(n+1)/2$ circles (n rows of circles) tangent to adjacent circles and/or the adjacent neighboring sides of a given triangle. Figure ?? is the case of $n = 6$, i.e., we need to pack 21 circles in the triangle. This difficult problem contains 174 variables (since we introduce auxiliary tangent points) and 6 linear equations and 168 quadratic equations which could not be block triangularized. Table ?? shows the running results for different n of this problem.

4. Symbolic Computation Approaches to AGDC

Theoretically, both the synthetic and numerical approaches are not complete. Synthetic approaches are not decision procedures even for the case of ruler and compass construction, i.e., for many problems that can be drawn with ruler and compass, they cannot give a drawing procedure. Using numerical approaches, if a solution is not found, we cannot assert that the system does not have a solution. Also, if we find one or two solutions of a system, we do not know whether these are all the solutions of the system. To solve these completeness problems, we may use the symbolic computation techniques such as Wu-Ritt's characteristic method [?], the Groebner basis method [?], the Dixon Resultant method [?], and Collins' cylindrical algebraic decomposition method [?]. For instance, using Ritt-Wu's characteristic method [?] we can give decision procedures ([?])

1. to construct a constrained diagram with ruler and compass (rc-constructibility);
2. to detect whether a constrained diagram is well-(under-, over-)constrained;

3. to detect whether a set parameters are, and if they are conflict, find the relation among them; and
4. to detect whether a constraint is redundant .

Let us consider how to draw a regular pentagon $ABCDE$ with edge r . Let $A = (0, 0)$, $B = (r, 0)$, $C = (x_2, y_2)$, $D = (x_3, y_3)$, $E = (x_4, y_4)$. Since there are five points in the diagram, we need $2 * 5 - 3 = 7$ constraints. The constraint $|AB| = r$ is already satisfied automatically. We need six other constraints:

$$\begin{aligned}
|AB| = |AE|: \quad h_1 &= y_4^2 + x_4^2 - r^2 = 0 \\
|AB| = |BC|: \quad h_2 &= y_2^2 + x_2^2 - 2rx_2 = 0 \\
|AB| = |CD|: \quad h_3 &= y_3^2 - 2y_2y_3 + x_3^2 - 2x_2x_3 + y_2^2 + x_2^2 - r^2 = 0 \\
|AB| = |DE|: \quad h_4 &= y_4^2 - 2y_3y_4 + x_4^2 - 2x_3x_4 + y_3^2 + x_3^2 - r^2 = 0 \\
|AD| = |DB|: \quad h_5 &= 2rx_3 - r^2 = 0 \\
|AC| = |CE|: \quad h_6 &= y_4^2 - 2y_2y_4 + x_4^2 - 2x_2x_4 = 0.
\end{aligned}$$

Wu-Ritt's characteristic method [?] can be used to represent the zero set of any polynomial equation system as the union of the zero sets of polynomial sets in *triangular form*. There are methods to find all solutions of a set of polynomial equations in triangular form. Using Wu-Ritt's characteristic set method to the pentagon problem, we have

$$Zero(\{h_1, \dots, h_6\}) = \cup_{i=1}^4 Zero(TS_i/I_i)$$

where $Zero(PS)$ denotes the set of all zeros or solutions of the polynomial equation system PS and $Zero(PS/D) = Zero(PS) - Zero(D)$ for a polynomial D . The polynomial sets TS_i in triangular form are given below.

$TS_1 =$	$TS_2 =$	$TS_3 =$	$TS_4 =$
$y_4 - y_2$	$ry_4 + (2x_2 - 2r)y_2$	$y_4 - y_2$	y_4
$x_4 + x_2 - r$	$2x_4 - 2x_2 + r$	$x_4 + x_2 - r$	$x_4 + r$
$ry_3 + (-2x_2 + r)y_2$	$ry_3 + (2x_2 - 3r)y_2$	$ry_3 + (2x_2 - 3r)y_2$	$y_3 - y_2$
$2x_3 - r$	$2x_3 - r$	$2x_3 - r$	$2x_3 - r$
$4y_2^2 - 2rx_2 - r^2$	$4y_2^2 - 2rx_2 - r^2$	$4y_2^2 - 2rx_2 - r^2$	$4y_2^2 + 5r^2$
$4x_2^2 - 6rx_2 + r^2$	$4x_2^2 - 6rx_2 + r^2$	$4x_2^2 - 6rx_2 + r^2$	$2x_2 + r$

TS_4 does not have non-zero solutions, since it contains $4y_2^2 + 5r^2$. Each of TS_1 , TS_2 , and TS_3 contains six polynomials and has total degree four, i.e., for a non-zero value of r , each of TS_1 , TS_2 , and TS_3 generally gives four solutions to the problem (see Figures 8).

4.1. Well-, Under-, and Over-Constrained Systems

A constraint system is called

- *well-constrained* if the shape of the corresponding diagram has only a finite number of cases;
- *under-constrained* if the shape of the corresponding diagram has infinite solutions;
- *over-constrained* if the corresponding diagram has no solution.

These properties are quite difficult to detect, so people usually use another kind of definition. A constraint system is called

- *structurally well-constrained* if each of its sub-diagram with n points and lines has $2 * n - 3$ constraints;
- *structurally under-constrained* if some of its sub-diagrams with n points and lines have less than $2 * n - 3$ constraints;
- *structurally over-constrained* if some of its sub-diagrams with n points and lines have more than $2 * n - 3$ constraints.

It is an interesting problem to study what kinds of structurally well-constrained systems are well constrained. In [?], some results are given. But in my viewpoint, these kinds of results are more theoretical oriented than practical oriented. Actually we can find many examples to show that structurally well- and over- constrained diagram could be under-constrained and structurally well-, under-, and over- constrained diagram could be over-constrained.

We use the pentagon example to show that sometimes we need more than $2n - 3$ constraints to define a constraint system with n points. The following table gives the result of checking three more geometric conditions on the three diagrams defined by TS_1 , TS_2 , and TS_3 using the automated theorem proving methods given in [?].

	TS_1	TS_2	TS_3
$ AD = AC $	true	false	false
$ BD = BE $	true	true	false
$ EB = EC $	true	false	true

Fig. 8. Three components for the regular pentagon problem

It is clear that the pentagon problem is well-defined. But the seven independent constraints do not properly define a regular pentagon. To define a regular pentagon, we need to add at least another constraint

$$|AD| = |AC|: \quad h_7 = y_3^2 + x_3^2 - y_2^2 - x_2^2.$$

Now there is only one useful triangular set TS_1 in the decomposition:

$$Zero(\{h_1, \dots, h_7\}) = Zero(TS_1/I_1) \cup Zero(TS_4/I_4).$$

But this constraint system is structurally over-constraint. Generally, we have [?]:

Theorem 4.1 *Using Wu-Ritt's characteristic method, we may decide whether a constrained diagram is well-, over-, or under- constrained over the field of complex numbers.*

4.2. Independent and Conflict Constraints

A set of constraints is said to be *independent* if no one of them can be removed and the new constraint system still defines the same diagram. A set of constraints is said to be *conflict* if no diagram in the Euclidean plane satisfies this constraint system.

Let us first comment that in the general case there is no connection between the concept of independence and conflict and the critical number $2n - 3$. Actually, there are conflict and/or non-independent systems with less than $2n - 3$ constraints, as well as non-conflict and/or independent systems with more than $2n - 3$ constraints.

For a constraint system CS , let PS be the corresponding algebraic equations of the constraints. The conflictness of CS means that PS does not have solutions. In the case of complex number field, this can be decided with Wu-Ritt's characteristic method [?, ?]. For the case of real numbers, a complete algorithm can be provided by the quantifier elimination algorithm [?]. For instance, if we want to draw a regular pentagon with edge 1 and diagonal 2, we will get a conflict constraint system. In other words, $Zero(PS) = \emptyset$ where PS consists of equations for $|AB| = |BC| = |CD| = |DE| = |EA| = 1$ and $|AD| = |DB| = |BE| = |EC| = 2$.

If two dimensional constraints are conflict, we can find the relation between the two dimensions. In [?], this is done with the Groebner basis. In [?], both the Groebner basis and Wu-Ritt's method are used to find such relations. In the regular pentagon problem, we just showed that the edge r and the diagonal d are two conflict dimensions. To find a relation between them, let PS be the set of equations for the constraints $|AB| = |BC| = |CD| = |DE| = |EA| = r$ and $|AD| = |DB| = |BE| = |EC| = d$. Using Wu-Ritt's characteristic method, we find two such relations:

$$d^2 + rd - r^2 = 0 \text{ and } d^2 - rd - r^2 = 0.$$

Since d and r are positive, the two reasonable solutions are $d = \frac{1+\sqrt{5}}{2}r$ and $d = \frac{-1+\sqrt{5}}{2}r$. The first one represents the case that r is the edge and d is the diagonal. The second one represents the case that r is the diagonal and d is the edge.

Let PS be the equation set of a constraint system. This system is not independent iff there is a polynomial P in PS such that

$$Zero(PS - \{P\}) \subset Zero(P) \text{ or } Zero(PS - \{P\}) = Zero(PS). \quad (4.1)$$

Note that to check (4.1) is actually to prove a geometry theorem. Let C_P be the geometric constraint of $P = 0$ and $C_1, \dots, C_n$ the geometric constraints for other equations in PS . Then (4.1) is equivalent to the following geometry theorem

$$\forall \text{ points}[(C_1 \wedge C_2 \cdots \wedge C_n) \Rightarrow C_P].$$

The problem of *automated geometry theorem proving* has been studied extensively [?, ?, ?]. We may use these efficient algorithms to prove whether (4.1) is true.

4.3. A Decision Procedure for Rc-constructibility

Let $\mathbf{Q}$ be the field of rational numbers. We allow us to abuse terms by saying that the numerical solutions of a diagram can be constructed with ruler and compass if the

corresponding diagram can be constructed with ruler and compass. It is well-known [?] that a number η can be constructed with ruler and compass iff there exist n real numbers $\eta_1, \eta_2, \dots, \eta_n = \eta$ and n quadratic equations

$$Q_1(x_1) = x_1^2 + b_1x_1 + c_1 = 0$$

$$Q_2(x_1x_2) = x_2^2 + b_2(x_1)x_2 + c_2(x_1) = 0$$

...

$$Q_n(x_1, x_2, \dots, x_n) = x_n^2 + b_n(x_1, \dots, x_{n-1})x_n + c_n(x_1, \dots, x_{n-1}) = 0$$

such that $Q_i(\eta_1, \eta_2, \dots, \eta_i) = 0, (i = 1, \dots, n)$. Since Q_i is of degree two, we can further assume that the b_i and c_i are linear in the variables $x_1, \dots, x_{i-1}$. From the above equations, we have the well known result: *an rc-constructible number is the root of an equation of degree 2^k for some $k > 0$. Thus the root of a cubic polynomial equation is rc-constructible iff the polynomial is reducible.*

Since we can efficiently factor polynomials with computer algebraic systems such as Maple [?], the only remaining problem is to decide whether the roots of polynomial equations of degree 2^k are rc-constructible.

Let us consider the simplest case: decide whether the roots of a quartic equation are rc-constructible. Let a quartic equation in $K[x]$ be

$$x^4 + h_3x^3 + h_2x^2 + h_1x + h_0 = 0 \quad (4.2)$$

where K is either the field of rational numbers $\mathbf{Q}$ or the field of rational functions $\mathbf{Q}(u_1, \dots, u_m)$. Let us assume that the quartic polynomial in (4.2) is irreducible. If a root y of (4.2) is rc-constructible, y can be written as the solution of the following two equations

$$x_1^2 + bx_1 + c = 0$$

$$y^2 + (-mx_1 + f)y - nx_1 + g = 0.$$

The first equation can be reduced to $x_1^2 + d = 0$. Since the roots are real numbers, we can further assume that the first equation is $x_1^2 - 1 = 0$. From the second equation, we have $x_1 = \frac{y^2 + fy + g}{my + n}$. Substitute this into $x_1^2 - 1 = 0$, the numerator is $(y^2 + fy + g)^2 - (my + n)^2 = 0$. Comparing the coefficients of y in this equation with that of (4.2), we obtain a set of equations

$$2f - h_3 = 0$$

$$m^2 - 2g - f^2 + h_2 = 0$$

$$2mn - 2fg + h_1 = 0$$

$$n^2 - g^2 + h_0 = 0$$

Eliminating f , m , and n , we have

$$8g^3 - 4h_2g^2 + (2h_1h_3 - 8h_0)g - h_0h_3^2 + 4h_0h_2 - h_1^2 = 0. \quad (4.3)$$

Let g_1 be a root of equation (4.3), f_1 , m_1 and n_1 solutions for f , m and n with g substituted by g_1 . Then all the roots of equation (4.2) can be obtained from the quadratic equations

$$y^2 + f_1y + g_1 \pm (m_1y + n_1) = 0.$$

Therefore, the roots of equation (4.2) can be obtained by successively solving several quadratic and a cubic equations. Roots of a cubic equation are rc-constructible iff the cubic polynomial

is reducible, i.e., it has a linear factor. Then we have a method of deciding whether the roots of a quartic equation can be constructed with ruler and compass. For the solution to the general case, please consult [?].

Let us consider Pappus' Problem: given a fixed point P , a fixed angle AOB , and a fixed length d , draw a line passing through point P and cutting lines OA and OB with a segment of length d . Let the segment be XY . Use the following coordinates $O = (0, 0)$, $A = (w, 0)$, $P = (u, v)$, $B = (w, s)$, $X = (x_1, y_1)$, $Y = (x_2, 0)$. If considering w, u, v , and s as non-zero constants, with Wu-Ritt's method, we can find a triangular set which contains the following quartic equation $f = s^2(s^2 + w^2)x_1^4 - 2ws^2(vs + wu)x_1^3 + (v^2 + u^2 - d^2)w^2s^2x_1^2 + 2d^2w^3vsx_1 - d^2w^4v^2 = 0$. With the method mentioned above, it is easy to check that this diagram generally cannot be drawn with ruler and compass. It is interesting to note that if we assume that point P is on the bisector of angle AOB then the diagram is rc-constructible.

5. Applications: Engineering Geometry

We mentioned before that AGDC is first considered in the community of mechanical CAD. Here we want to show that by considering AGDC in a more general sense, it can be used to more engineering problems such as linkage design, computer vision, and geometric modeling. Other engineering applications of AGDC include robotics [?, ?, ?] and geometric modeling [?, ?]. AGDC methods can also be used to problems from mathematics, physics, and education. We call the methods of AGDC and these applications *engineering geometry*.

5.1. Mechanical Formula Derivation

In geometry, the techniques for AGDC have close connection with the so-called *mechanical formula derivation*. There are two kinds of formula derivation problems. One is deriving geometry formulas [?, ?, ?]; the other is finding locus equations [?, ?]. The problem of finding the relation between the edge and the diagonal of a regular pentagon in section ?? is an example of finding geometry formulas. In general, all these problem can be considered as find the manifold solutions of equation systems of positive dimension [?].

For a geometric configuration given by a set of polynomial equations

$$h_1(u_1, \dots, u_q, x_1, \dots, x_p) = 0, \dots, h_r(u_1, \dots, u_q, x_1, \dots, x_p) = 0$$

and a set of inequations

$$\{D = d_1(u, x) \neq 0, \dots, d_s(u, x) \neq 0\},$$

we want to find a relation (formula) between arbitrarily chosen variables $u_1, \dots, u_q$ (parameters) and a dependent variable, say, x_1 . In [?], one method based on the Groebner basis and one method based on the characteristic set are given.

An interesting example is Peaucellier's Linkage (Figure 9). Links AD , AB , DC and BC have equal length, as do links EA and EC . The length of FD equals the distance from E to F . The locations of joints E and F are fixed points on the plane, but the linkage is allowed to rotate about these points. As it does, what is the locus of the joint B ?

Fig. 9. A linkage generating a straight line

Let $F = (0, 0)$, $E = (r, 0)$, $C = (x_2, y_2)$, $D = (x_1, y_1)$, and $B = (x, y)$, n and m be the lengths of the projections of AD on the direction of line EDB (note that points E , D , and B are always collinear) and on the direction perpendicular to it. Then the geometry conditions can be expressed by the following set of polynomial equations H

$$\begin{aligned} h_1 &= y_1^2 + x_1^2 - r^2 = 0 & r &= FD \\ h_2 &= y_2^2 - 2y_1y_2 + x_2^2 - 2x_1x_2 + y_1^2 + x_1^2 - n^2 - m^2 = 0 & CD &= n^2 + m^2 \\ h_3 &= y_2^2 - 2yy_2 + x_2^2 - 2xx_2 + x^2 + y^2 - n^2 - m^2 = 0 & CB &= n^2 + m^2 \\ h_4 &= y_2^2 + x_2^2 - 2rx_2 - n^2 - 4rn - m^2 - 3r^2 = 0 & EC &= (n + 2r)^2 + m^2 \\ h_5 &= (x - r)y_1 - yx_1 + ry = 0 & E &\text{ is on } DB, \end{aligned}$$

together with the following set of polynomial inequations D :

$$d_1 = x_1 - x \neq 0 \quad B \neq D.$$

Selecting m , n , r , and y to be the parameters of the problem, we want to find the relation among m , n , r , y and x . Using characteristic set method in [?], we have found that $\text{Zero}(H/D)$ has only one non-degenerate component with the corresponding ascending chain $\text{ASC}_1^* =$

$$\begin{aligned} &x + 2n + r \\ &(x^2 - 2rx + y^2 + r^2)x_1 + rx^2 - 2r^2x - ry^2 + r^3 \\ &(x - r)y_1 - yx_1 + ry \\ &(4x^2 - 8rx + 4y^2 + 4r^2)x_2^2 + (-4x^3 + 4rx^2 + (-4y^2 - 16rn - 12r^2)x - 4ry^2 + 16r^2n + 12r^3)x_2 \\ &\quad + x^4 + (2y^2 + 8rn + 6r^2)x^2 + y^4 + (-4n^2 - 8rn - 4m^2 - 6r^2)y^2 + 16r^2n^2 + 24r^3n + 9r^4 \\ &2yy_2 + (2x - 2r)x_2 - x^2 - y^2 - 4rn - 3r^2. \end{aligned}$$

The first polynomial is the relation $x + 2n + r = 0$, which tells us that the locus is a line parallel to the y -axis.

5.2. Dynamic Geometry

By *dynamic geometry*, we mean models built by computer softwares that can be changed dynamically [?, ?]. Dynamic models have many advantages over models built with real materials. Their basic properties are: *dynamic transformation*, *free dragging*, *animation*, and *locus generation*. By doing dynamic transformation and free dragging, we can obtain various forms of diagrams easily and see the changing process vividly. Through animation, the user may observe the generation process of curves or the figures of functions and get an intuitive idea of the properties of the functions and curves. Furthermore, we can combine free dragging with locus generation to give a powerful tool of showing the properties of curves or functions.

Dynamic geometry is originated in the field of CAI (computer aided instruction) [?],

although it can be used in other areas such as linkage design [?] and computer vision [?].

Geometry Expert (GEX) [?] is a software for dynamic diagram drawing and automated geometry theorem proving and discovering. As a dynamic geometry software, GEX can be used to build *dynamic visual models* to assist teaching and learning of various mathematical concepts. As an automated reasoning software, we can build *dynamic logic models* which can do reasoning themselves. Logic models can be used for more intelligent educational tasks, such as automated generation of test problems, automated evaluation of students' answers, intelligent tutoring, etc.

With Geometry Expert, we can build the following classes of dynamic visual models.

1. Loci generated by diagrams constructed using ruler and compass. This class includes conics, trigonometric functions, various transcendental curves, and the following curves built with *Horner's construction* [?].
 - (a) Rational functions $y = \frac{f(x)}{g(x)}$.
 - (b) Inverse function and particularly function $y = \sqrt[n]{f(x)}$.
 - (c) Diagrams for any *rational plane curves* defined by their parametric equations.

$$x = \frac{P(t)}{R(t)}; y = \frac{Q(t)}{R(t)}$$

where P, Q, R are polynomials in t .

- (d) **Interpolation polynomials.** For n distinct points $(x_i, y_i), i = 1, \dots, n$, draw the diagram of a polynomial $y = f(x)$ such that $y_i = f(x_i)$.
2. Diagrams of functions using the numerical computation. This class includes functions of the form: $y = f(x)$ where $f(x)$ could be any "elementary functions" – $a^x, x^a, \log(x)$, trigonometric functions – and their arithmetic expressions and compositions. This part is quite similar to most "Graphic Calculator", but is more flexible and powerful.
3. Loci generated by linkages with rotating joints alone. As proved by Kempe [?], this class includes any algebraic curve $f(x, y) = 0$ where $f(x, y)$ is a polynomial of x and y .

To demonstrate how to use free dragging and locus generation, we consider the following problem. Let H be the orthocenter of triangle ABC . We fix points A and B and let point C moves on a circle c . We want to know the shape of the locus of point H . Let (x_0, y_0) and r be the center and radius of circle c , and $A = (0, 0)$, and $B = (d, 0)$. With the method in Section ??, we can derive the equation of this locus

$$((x - x_0)^2 + y_0^2 - r^2)y^2 + 2y_0(x - d)xy + (x - d)^2x^2 = 0.$$

But from this equation, it is still difficult to know the shape of this curve. With dynamic geometry software like *Geometry Expert*, we can continuously changing the radius of circle c by free dragging and observe the shape of the curve changing continuously (Figure 10).

Fig. 10. Shapes of the locus of the orthocenter when the radius of circle c changes.

5.3. Parametric CAD

A direct application of AGDC is mechanical CAD systems. Here the advantage over the traditional CAD systems is not limited to a more user friendly interface. In the constraint based system, the user may specify a design figure without knowing its shape previously which cannot be done in traditional CAD system. The constraint based system will find all the possible shapes of the specified system automatically. This is especially important in the phase of conceptual design.

For instance, for diagram in Figure 11, if the radius of circle B is known then this diagram is easy to draw. If we do not know the radius of this circle but instead know that this circle passing through a known point A , then this diagram is not that easy to draw with ruler and compass. This problem is actually one of the ten Appolonius' construction problems [?]. A solution to this problem generated by the global propagation method is given below.

Fig. 11. Appolonius' construction problem and CAD

Suppose that O_1 and O_2 are two known circles and A is a known point. We need to construct a circle B which tangents to circles O_1 and O_2 and passes through point A . Figure 11 (b) shows the solution given by our method. By a construction rule [?], line UV passes through the similarity center of circles O_1 and O_2 , i.e., the intersection W of line O_1O_2 and a common tangent line T_1T_2 of the two circles (which is easy to construct). Let S be the intersection of the circle passing through T_1, T_2, A and the circle with O_2 as center and passing through T_2 . H the intersection of lines AW and T_2S . Then O_2V is perpendicular to HV . We hence can construct point V . Now U can be constructed easily.

5.4. Linkage Design: A Realization of Kempe's Linkage

In Section ??, we gave a linkage to generate a straight line. It is also able to design linkages to generate conics [?]. So it is natural to ask: what kinds of curves can be drawn with linkages. This problem has been answered theoretically by Kempe [?]. Kempe proved that any algebraic plane curve can be generated with a linkage. With the AGDC techniques reported in Sections ??, ??, and ??, we may design a linkage automatically for a plane algebraic curve and to draw the curve with this the linkage dynamically.

First we need three kinds of tools.

1. The *Multiplicator* shown in Figure 12 (a) consists of similar crossed parallelograms. Using a multiplicator, we may obtain integral multiples of any angle, e.g. $a\varphi$ or $b\theta$.

Fig. 12. Three tools for linkage construction

2. The *Additor*. Joining one multiplicator to another will produce the combination $a\varphi \pm b\theta$. This is the mechanism shown in Figure 12 (b) where the plate BOK with angle B is connected rigidly to the bar. Thus we build up a linkage to produce $\angle BOK = a\varphi \pm b\theta \pm \beta$. If OB is taken equal to A , then the x-coordinate of the point B is $A \times \cos(a\varphi \pm b\theta \pm \beta)$.

3. The *Translator* shown in Figure 12 (c) consists of parallelograms with OB pivoted at O . In this linkage, the bar $O'B'$ is always parallel to OB and can be moved freely within its limits.

Let the algebraic plane curve be defined by the following equation

$$f(x, y) = 0. \quad (5.1)$$

We construct a parallelogram (Figure 13 (a)) such that OA and OB have lengths m and n and form angles θ and φ with the X-axis respectively. P is a point on the curve. Its coordinates are then given below.

$$x = m \times \cos \theta + n \times \cos \varphi; \quad (5.2)$$

$$y = m \times \sin \theta + n \times \sin \varphi = m \times \cos(\pi/2 - \theta) + n \times \cos(\pi/2 - \varphi) \quad (5.3)$$

Note that the products and powers of cosines can be expressed as the sum of cosines. If substituting equations (5.2) and (5.3) into (5.1), we shall have a sum of terms of the following form

$$f(x, y) = \sum_{i=1}^k A_i \times \cos(a_i \varphi \pm b_i \theta \pm \beta_i) + C = 0 \quad (5.4)$$

where A_i and C are constants; a_i and b_i are positive integers; and β_i equals 90, or 0.

Fig. 13. Realization of Kempe's linkage

For each term $A_i \times \cos(a_i\varphi \pm b_i\theta \pm \beta_i)$, we can use the Multiplier and Adder to construct a link OC_i such that OC_i is of length A_i and form an angle $a_i\varphi \pm b_i\theta \pm \beta_i$ with the X-axis. Using the Translators, we can construct a chain of links $OB_1, B_1B_2, B_2B_3, \dots$, as shown in Figure 12 (b), such that $B_1 = C_1$ and $OC_iB_iB_{i-1}$ is a parallelogram. Therefore, point B_n has x-coordinate :

$$X = \sum_{i=1}^n A_i \times \cos(a_i\varphi \pm b_i\theta \pm \beta_i) = f(x, y) - C \text{ (by equation (5.4))}$$

If P is moved along the given curve, its coordinates x, y satisfy: $f(x, y) = 0$. Accordingly, the locus of the end point B_n of the chain is

$$X + C = 0$$

a straight line parallel to the Y-axis. Conversely, if B_n is moved along this line (with the help of a Peaucellier cell, for instance) point P will generate the curve $f(x, y) = 0$. Thus we finish the construction.

After the construction of the chain of points $B_1, \dots, B_n$, we need to let point B_n move on line $X + C = 0$ and for a given position of B_n we find the positions of other points using a numerical method, such as the optimization method introduced in Section ?? . For examples of Kempe linkages, see [?].

5.5. Pose Determination: the PnP Problem

The Perspective- n -Point Problem (PnP), also known as the Location Determination Problem (LDP) [?] or the Exterior Camera Calibration Problem [?], is originated from camera calibration [?]. It is to determine the exterior parameters: the position and orientation of the camera with respect to a scene object from n correspondent points. It concerns many important fields, such as computer animation, computer vision, image analysis and automated cartography, robotics, etc. Fischler and Bolles [?] summary the problem as follows:

“ Given the relative spatial locations of n control points P_i , and given the angle to every pair of control points from an additional point called the Center of Perspective (C_P), find the lengths of the line segments joining C_P to each of the control points.”

Let $x_i = |C_P P_i|$, $a_{ij} = |P_i P_j|$, and $p_{ij} = \cos(\angle P_i C_P P_j)$. Applying the cosine theorem to triangles $P_i C_P P_j$, we obtain the PnP equation system:

$$x_i^2 + x_j^2 - 2x_i x_j p_{ij} - a_{ij}^2 = 0, (i = 1, \dots, n; j = i + 1, \dots, n).$$

The study of the PnP problem mainly consists of two aspects:

Diagram Construction. Design fast and stable algorithms that can be used to find all or some of the solutions of the PnP problem.

Solution Classification. Give a classification for the solutions of the PnP equation system, i.e., give explicit conditions under which the system has none, one, two, ... number of real physical solutions.

There are many results for the first problem and the second problem is not solved completely.

All three kinds of AGDC techniques studied in this paper can be used to attack this problem. In [?], the synthetic approach and the symbolic computational approach are used to study the $P3P$ problem. In [?], symbolic computation methods are used to the solution classification problem.

In the symbolic computation approach, we apply Wu-Ritt's zero decomposition algorithm [?, ?] to find a complete solution decomposition for the $P3P$ equation system. The decomposition has the following implications. First, it provides the first complete analytical solution to the $P3P$ problem. Previous work usually consider the main solutions and omit many special cases. This might cause problems when the given data is from the special case. Second, by expressing all solutions in *triangular form* it provides a fast and stable way for numerical solution. Third, it provides a clear solution space analysis of the $P3P$ problem and thus provides a good starting point for multiple solution analysis.

In the geometric approach, the three perspective angles are considered separately. Then the locus of the control point in each case is a torus and the control point is the intersection of the three tori. In this way, we give some pure geometric criteria for the number of solutions of the $P3P$ problem. One interesting result is

Theorem. *The $P3P$ problem can have only one solution if all the three angles formed by the three control points are obtuse.*

For the special case $a_{12} = a_{13} = a_{23} = 1$ and $p_{12} = p_{13}$, the following classification for the $P3P$ problem is given in [?]. For the reason of simplicity, we use two new parameters $p_{12} = r$ and $p_{23} = p$.

1. Point P has four solutions, if and only if

$$1 > q > \frac{1}{2}, \quad \frac{1+q^2}{2} > p > \frac{1}{2}, \quad \text{and } p > \frac{3}{2} - \frac{1}{2q^2}.$$

2. Point P has three solutions, if and only if

$$\left\{ \begin{array}{l} \frac{1}{2} < q < \frac{\sqrt{2}}{2} \\ \frac{3}{2} - \frac{1}{2q^2} < p \leq \frac{1}{2} \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} \frac{1}{2} < q < 1 \\ p = \frac{1+q^2}{2} \end{array} \right.$$

3. Point P has two solutions, if and only if

$$\left\{ \begin{array}{l} 0 < q \leq \frac{1}{2} \\ \frac{1}{2} < p < \frac{1+q^2}{2} \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} \frac{1}{2} < q < 1 \\ \frac{1+q^2}{2} < p < 1 \end{array} \right.$$

$$\text{or} \quad \left\{ \begin{array}{l} \frac{\sqrt{2}}{2} < q < 1 \\ \frac{1}{2} < p \leq \frac{3}{2} - \frac{1}{2q^2} \text{ and } p > 2q^2 - 1. \end{array} \right.$$

Fig. 14. Solution classification for $a_{12} = a_{13} = a_{23} = 1$ and $p_{12} = p_{13}$. L_1 is $p = \frac{1+q^2}{2}$, L_2 is $p = 2q^2 - 1$, and L_3 is $p = \frac{3}{2} - \frac{1}{2q^2}$. The number of solutions on the curves are omitted in this diagram.

4. Point P has one solution, if and only if

$$\left\{ \begin{array}{l} -\frac{\sqrt{3}}{2} < q < \frac{1}{2} \text{ or } \frac{\sqrt{2}}{2} \leq q < \frac{\sqrt{3}}{2} \\ 2q^2 - 1 < p < \frac{1}{2} \end{array} \right\} \text{ or } \left\{ \begin{array}{l} \frac{1}{2} < q < \frac{\sqrt{2}}{2} \\ 2q^2 - 1 < p \leq \frac{3}{2} - \frac{1}{2q^2} \end{array} \right.$$

$$\text{or } \left\{ \begin{array}{l} 0 < q \leq \frac{1}{2} \text{ or } \frac{\sqrt{2}}{2} \leq q < \frac{\sqrt{3}}{2} \\ p = \frac{1}{2} \end{array} \right\} \text{ or } \left\{ \begin{array}{l} 0 < q \leq \frac{1}{2} \\ p = \frac{1+q^2}{2} \end{array} \right.$$

References

- [1] B. Aldefeld, Variation of Geometries Based on a Geometric-Reasoning Method, *Computer Aided Design*, 20(3):117-126, 1988.
- [2] F. Arbab & B. Wang, Reasoning About Geometric Constraints, *Intelligent CAD II*, H. Yoshikawa and T. Holden (eds), North-Holland, 93-107, 1990.
- [3] A. Borning, The Programming Language Aspect of ThingLab, *ACN Tras. on Programming Language and Systems*, Vol. 3, No.4, 353-387, 1981.
- [4] W. Bouma, C.M. Hoffmann, I. Fudos, J. Cai and R. Paige, A Geometric Constraint Solver, *Computer Aided Design*, 27(6):487-501, 1995.
- [5] B. Brudelin, Constructing Three-Dimensional Geometric Objects Defined by Constraints, in *Workshop on Interactive 3D Graphics*, p.111-129, ACM, 1986.
- [6] S. A. Buchanan and A. de Pennington, Constraint Definition System: A Computer Algebra Based Approach to Solving Geometric Problems, *Computer Aided Design*, 25(12):740-750, 1993.
- [7] B. Buchberger, Gröbner Bases: an Algorithmic Method in Polynomial Ideal Theory, *Recent Trends in Multidimensional Systems theory*, D.Reidel Publ. Comp., 1985.
- [8] B. Char, K. Geddes, *Maple V*, Springer-Verlag, 1992.

- [9] S. C. Chou, Proving elementary geometry theorems using wu's algorithm, in W. Bledsoe, D. Loveland, eds, *Automated Theorem Proving: After 25 years*, AMS Contemporary Mathematics Series, chapter 29, pp. 243–286.
- [10] S. C. Chou, *Mechanical Geometry Theorem Proving*, D.Reidel Publishing Company, Dordrecht, Netherlands, 1988.
- [11] S. C. Chou, A Method for Mechanical Deriving of Formulas in Elementary Geometry, *J. of Automated Reasoning*, **3**(1987), 291—299.
- [12] S. C. Chou & X. S. Gao, Mechanical Formula Derivation in Elementary Geometries, *Proc. ISSAC-90*, ACM, New York, 1990, p. 265–270.
- [13] S. C. Chou & X. S. Gao, Ritt-Wu's Decomposition Algorithm and Geometry Theorem Proving, *CADE'10*, M.E. Stickel (Ed.) pp 207–220, Lect. Notes in Comp. Sci., No. 449, Springer-Verlag, 1990.
- [14] S. C. Chou, X. S. Gao, & J. Z. Zhang, *Machine Proofs in Geometry*, World Scientific, 1994.
- [15] S. C. Chou, X. S. Gao, & J.Z. Zhang, A Fixpoint Approach To Automated Geometry Theorem Proving, WSUCS-95-2, CS Dept, Wichita State University, 1995.
- [16] G. E. Collins, Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition, *Lect. Notes Comp. Sci.* 33, 134–183, Springer-Verlag, 1975.
- [17] J. Chuan, Geometric Constructions with the Computer, *Proc. of ATCM95*, pp.329-338.(1995)
- [18] K.H. Elster (ed), *Modern Mathematical Methods of Optimization*, Akademie Verlag, 1993.
- [19] M. A. Fishler, and R. C. Bolles, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartomated Cartography*, Communications of the ACM, Vol.24, No.6, pp.381-395, June 1981.
- [20] I. Fudos and C.M. Hoffmann, A Graph-Constructive Approach to Solving Systems of Geometric Constraints, *ACM Transactions on Graphics*, 16(2):179-216, 1997.
- [21] X. S. Gao & S. C. Chou, Solving Geometric Constraint Systems, I. A Global Propagation Approach, *Computer Aided Design*, Vol. 30, No. 1, 47-54, Elsevier Sci Ltd., 1998.
- [22] X. S. Gao & S. C. Chou, Solving Geometric Constraint Systems, II. A Symbolic Computational Approach, *Computer Aided Design*, Vol. 30, No.2, 115-122, Elsevier Sci Ltd., 1998.
- [23] X.S. Gao & S.C. Chou, Implicitization of Rational Parametric Equations, *Journal of symbolic computation*, (1992) 14, 459–470.
- [24] X. S. Gao, J. Z. Zhang, and S. C. Chou, *Geometry Expert*, Nine Chapters Pub., 1998, Taiwan.
- [25] X. S. Gao, C. C. Zhu, and Y. Huang, Building Dynamic Mathematical Models with Geometry Expert, I. Geometric Transformations, Functions and Plane Curves, *Proceedings of the Third Asian Technology Conference in Mathematics*, eds W.C. Yang, pp. 216-224, Springer, 1998.
- [26] X. S. Gao, C. C. Zhu, and Y. Huang, Building Dynamic Mathematical Models with Geometry Expert, II. Linkages, *Proceedings of the Third Asian Symposium on Computer Mathematics*, eds Z. B. Li, pp. 15-22, Springer, 1998.
- [27] X. S. Gao, H. F. Cheng, On the Solution Classification of the “P3P” Problem, *Proceedings of the Third Asian Symposium on Computer Mathematics*, eds Z. B. Li, pp. 185-200, Lanzhou University Press, 1998.
- [28] J. X. Ge, S. C. Chou, & X. S. Gao, Geometric Constraint Satisfaction Using Optimization Methods, WSUCS-98-1, CS Dept, Wichita State University, 1998, *submitted to CAD*.
- [29] H. Gelernter, Realization of a Geometry-Theorem Proving Machine, *Computers and Thought*, eds E.A. Feigenbaum, J. Feldman, p. 134-152, McGraw Hill.

- [30] J. Hopcroft and R. Tarjan, Dividing A Graph into Triconnected Components, *SIAM J. Computing*, Vol. 2, No. 3, 135 – 157, 1973.
- [31] R. Horaud, B. Conio, and O. Leboulleux, *An Analytic Solution for the Perspective 4-Point Problem*, CVGIP, Vol.47, pp.33-44, 1989.
- [32] A. Heydon & G. Nelson, The Juno-2 Constraint-Based Drawing Editor, *SRC Research Report 131a*, 1994.
- [33] C. Hoffmann, Geometric Constraint Solving in R^2 and R^3 , in “Computing in Euclidean Geometry”, eds D. Z. Du and F. Huang, p. 266–298, World Scientific, 1995.
- [34] C. Hoffmann and I. Fudos, Constraint-based Parametric Conics for CAD, *Geometric Aided Design*, vol 28, No. 2, p.91–100, 1996.
- [35] Y. Huang and W. D. Wu, Kinematic solution of a steawrt platform, in *Proc. IWMM’92*, Inter. Academic Publishers, Beijing, pp. 181–188, 1992.
- [36] N. Jakiw, *Geometer’s Sketchpad*, User guide and Reference Manual, Key Curriculum Press, 1994.
- [37] N. Jacobson, *Basic Algebra*, Vol. 1, San Francisco, Freeman, 1985.
- [38] D. Kapur, Geometry Theorem Proving Using Hilbert’s Nullstellensatz, *Proc. of SYMSAC’86*, Waterloo, 1986, 202–208.
- [39] D. Kapur, T. Saxena, L. Yang L., Algebraic and Geometric Reasoning with Dixon resultants, in *Proc. of ISSAC’94*, Oxford, 1994.
- [40] A. B. Kempe, On a general method of describing plane curves of the nth degree by linkwork, *Proc. of L.M.S.*, p.213–216, 1876; voir aussi *Messenger of Math.*, T. VI., p.143-144.
- [41] K. Kondo, Algebraic method for manipulation of dimensional relationships in geometric models, *Geometric Aided Design*, 24(3), p. 141-147, 1992.
- [42] G. Kramer, *Solving Geometric Constraint Systems*, MIT Press, 1992.
- [43] G. Kramer, A Geometric Constraint Engine, *Artificial Intelligence*, vol. 58, p. 327–360, 1992.
- [44] H. Lamure and D. Michelucci, Solving Geometric Constraints By Homotopy, *IEEE Trans on Visualization and Computer Graphics*, 2(1):28-34, 1996.
- [45] R.S. Latheam and A.E. Middleditch, Connectivity Analysis: a Tool for Processing Geometric Constraints, *Computer Aided Design*, 28(11):917-928, 1994.
- [46] J.Y. Lee, and K. Kim, Geometric Reasoning for Knowledge-Based Parametric Design Using Graph representation *Computer Aided Design*, 28(10):831-841, 1996.
- [47] K. Lee, and G. Andrews, Inference of the Positions of Components in an Assembly: part2, *Computer Aided Design*, 17(1):20-24, 1985.
- [48] W. Leler, *Constraint Programming Languages*, Addison Wesley, 1988.
- [49] R. Light and D. Gossard, Modification of Geometric Models through Variational Geometry, *Geometric Aided Design*, 14:208-214, July, 1982.
- [50] V.C. Lin, D.C. Gossard, & R.A. Light, Variational Geometry In Computer-Aided Design, *Computer Graphics*, vol. 15, no. 3, 171–177, 1981.
- [51] G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd (eds), *Optimization*, Elsevier Science Publishers B.V., 1989.
- [52] J. Owen, Algebraic Solution for Geometry from Dimensional Constraints, in *ACM symp. Found. of Solid Modeling*, ACM press, p. 397-407, Austin TX, 1991.
- [53] J. Owen, Constraints of Simple Geometry in Two and Three Dimensions, *Inter. J. of Comp. Geometry and Its Applications*, 1996, 6, 421-434.

- [54] D.N. Rocheleau and K. Lee, System for Interactive Assembly Modeling, *Computer Aided Design*, 19(1):65-72, 1987.
- [55] D. Serrano and D. Gossard, Constraint management in MCAE, in *Artificial Intelligence in Engineering: Design*, J. Gero (ed.), 93-110, Elsevier, Amsterdam, 1988.
- [56] G. L. Steele & G. L. Sussman, CONSTRAINTS – A Language for Expressing Almost-Hierarchical Descriptions, *Artificial Intelligence*, p. 1-39, Vol. 14, 1980.
- [57] H. Suzuki, H. Ando and F. Kimura, Geometric Constraints and Reasoning for Geometrical CAD Systems, *Computer and Graphics*, 14(2):211-224, 1990
- [58] G. Sunde, Specification of Shape by Dimensions and Other Geometric Constraints, in *Geometric Modeling for CAD Applications*, M. J. Wozny, et al eds, 199-213, North Holland, 1988.
- [59] I. Sutherland, Sketchpad, A Man-Machine Graphical Communication System, in *Proc. of the Spring Joint Comp. Conference*, North-Holland, 329-345, 1963.
- [60] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*, Univ. of California Press, Berkeley, Calif., 1951.
- [61] P. Todd, A k-tree Generalization that Characterizes Consistency of Dimensioned Engineering Drawings, *SIAM J. of Disc. Math.*, Vol 2, 255-261, 1989.
- [62] R. C. Veltkamp, Geometric Constraint Management with Quanta, *Intelligent Computer Aided Design*, D.C. Brown et al eds, 409-426, North-Holland, 1992.
- [63] A. Verroust, F. Schonek, and D. Roller, Rule-oriented method for parameterized computer-aided design. *Geometric Aided Design*, 24(3), p.531-540, October, 1992.
- [64] D. M. Wang & X. S. Gao, Geometry Theorems Proved Mechanically Using Wu's Method, Part on Elementary Geometries, MM preprint No2, 1987.
- [65] W. T. Wu *Basic Principles of Mechanical Theorem Proving in Geometries*, Springer-Verlag, 1993.
- [66] W. T. Wu A Mechanizations Method of Geometry and its Applications I. Distances, areas and volumes, *J. Sys. Sci. & Math. Scis.*, 6(1986), 204-216.
- [67] W. T. Wu *Mathematics Mechanization*, Science Press, 1998.
- [68] W. T. Wu A Mechanization Method of Geometry and Its Applications VI. Solving Inverse Kinematics Equations of PUMA-Type Robotics MM Research Preprints, No. 4, 1989, Ins. of Systems Science, Academia Sinica.
- [69] W. T. Wu and D K. Wang, On the Surface Fitting Problems in CAGD, *Mathematics in Practice and Theory*, 3, 1994.
- [70] L. Yang, H. Fu, & Z. Zeng, A Practical Symbolic Algorithm for Inverse Kinematics of 6R Manipulators with Simple Geometry, *Proc. CADE14*, pp. 73-86, Springer-Verlag, Berlin, 1997.
- [71] L. Yang, J.Z. Zhang, & X.R. Hou, Non-Linear Algebraic Equations and Theorem Machine Proof, ShangHai Science and Education Press, 1997, (in Chinese).
- [72] L. Yang, A Simplified Algorithm for Solution Classification of the P3P Problem, this Proceedings.
- [73] J. Z. Zhang, L. Yang, & X. C. Yang, The Realization of Elementary Configurations in Euclidean Space, *Science in China (series A)*, vol. 37, No.1, p.15-26, 1994.