MMP/Geometer
- A Software Package for Automated Geometry Reasoning*

Xiao-Shan Gao and Qiang Lin
Institute of System Science, AMSS, Academia Sinica
Beijing 100080, China
email: (xgao,qlin)@mmrc.iss.ac.cn

Abstract

In this paper, we introduce a software package, MMP/Geometer, developed by us to automate some of the basic geometric activities including geometric theorem proving, geometric theorem discovering, and geometric diagram generation. As a theorem prover, MMP/Geometer implements Wu's method for Euclidean and differential geometries, the area method and the geometric deductive database method. With these methods, we can not only prove difficult geometric theorems but also discover new theorems and generate short and readable proofs. As a geometric diagram editor, MMP/Geometer is an intelligent dynamic geometry software tool which may be used to input and manipulate geometric diagrams conveniently and interactively by combining the idea of dynamic geometry and methods of automated diagram generation.

Keywords. Geometry software, automated reasoning, geometric theorem proving, geometric theorem discovering, geometric diagram generation, intelligent dynamic geometry.

1 Introduction

MMP (Mathematics-Mechanization Platform) is a stand-alone software platform under development [12], which is supported by a Chinese National Key Basic Research Project "Mathematics Mechanization and Platform for Automated Reasoning." The aim of MMP is to mechanize some of the basic mathematical activities including automated solution of algebraic and differential equations and automated geometry theorem proving and discovering, following Wu's idea of mathematics mechanization [31, 34]. MMP is based on algorithms for symbolic computation and automated reasoning, and in particular the Wu-Ritt characteristic set (CS) method [34, 28]. MMP also implements application packages in automated geometry reasoning, automated geometric diagram generation, differential equation solving, robotics, mechanism design, and CAGD.

MMP/Geometer is a package of MMP for automated geometry reasoning. The aim of MMP/Geometer is try to automate some of the basic geometric activities including geometric theorem proving, geometric theorem discovering, and geometric diagram generation. The current version is mainly for plane Euclidean geometries and the differential geometry of space curves.

The goal of MMP/Geometer is to provide a convenient and powerful tool to learn and use geometry by combining the methods of geometric theorem proving and geometric diagram generation. The introduction of computer into geometry may give new life into the learning and study of the classical field [10]. Geometry is at the heart for many engineering problems from robotics, CAD, and computer vision. We expect that MMP/Geometer may have applications in these fields. Actually, some of the methods implemented in MMP/Geometer are directly targeted at engineering problems [11, 14, 15, 16].

*This work was supported by a National Key Basic Research Project (NO. G19980306) and by a USA NSF grant CCR-0201253.

1.1 Automated Geometric Theorem Proving and Discovering

Study of automated geometric theorem proving (AGTP) may be traced back to the landmark work by Gelernter and his collaborators [17] in the late fifties. The extensive study of AGTP in the past twenty years is due to the introduction of Wu's method in late seventies [31], which is surprisingly efficient for proving difficult geometric theorems. Inspired by this work, many successful methods of AGTP were invented in the past twenty years. For a recent survey of these methods, please consult [7].

AGTP is one of the successful fields of automated reasoning. There are few areas for which one can claim that machine proofs are superior to human proofs. Geometry theorem proving is such an area. Our experiments show that MMP/Geometer is a quite efficient in proving geometry theorems. Within its domain, it invites comparison with the best of human geometry provers. Precisely speaking, we have implemented the following methods.

Wu's method might be the most powerful method in terms of proving difficult geometric theorems and applying to more geometries [31, 34]. Wu's method is a coordinate-based method. It first transfers geometric conditions into polynomial or differential equations in the coordinates of the points involved, then deals with the equations with the characteristic set method.

The area method uses high-level geometric lemmas about geometry invariants such as the area and the Pythagorean difference as the basic tool of proving geometry theorems [8]. The method can be used to produce human-readable proofs for geometry theorems.

The deductive database method is based on the theory of deductive database. We may use it to generate the fixpoint for a given geometric configuration under a fixed set of geometric rules or axioms[9]. With this method, we can not only find a large portion of the well-known facts about a given configuration, but also to produce proofs in traditional style.

For almost every method of AGTP, there is a prover. We will not give detailed introduction to existing geometry software packages. A survey may be found in [19]. Comparing with previous provers, MMP/Geometer has the following distinct features. First, it implements some of the representative methods for AGTP, while most previous provers are for one method. An exception is Geometry Expert (GEX), which also implements the methods mentioned above [13]. The difference is that GEX only implements a simple version of Wu's method due to the lack of an implementation of Wu-Ritt's zero decomposition theorem. Second, MMP/Geometer is stand-alone, while most of the previous provers are implemented in Lisp or Maple. Third, MMP/Geometer is capable of producing human-readable proofs and proofs in traditional style. Finally, MMP/Geometer has a powerful graphic interface, which will be introduced in the next subsection.

1.2 Automated Geometric Diagram Generation (AGDG)

By implementing general AGDG methods, MMP/Geometer may be used as a general diagram editor. It is often said that a picture is more than one thousand words. But in reality, it is still very difficult to generate large scale geometric diagrams like those in Figure 10 with a computer. With MMP/Geometer, we intend to provide an intelligent tool for generating and manipulating such diagrams. Also, AGDG methods have direct applications in parametric CAD [18, 16, 14], linkage design [15], etc.

For AGTP, the diagram editor may provide a nice graphic user interface (GUI). Also, AGDG methods may enhance the proving scope for AGTP methods with constructive statements as input by finding the construction sequence for problems whose ruler and compass construction is difficult to find, as shown by the example in Figure 9.

Dynamic geometry software systems, noticeably, Gabri [21], Geometer's Sketchpad [20], Cinderella [26], and Geometry Expert [13] may generate diagrams interactively based on ruler and compass construction. These systems are mainly used to education and simulation of linkages. As compared with models built with real materials, visual models built with dynamic geometry software are more flexible, powerful, and more open for manipulation.

It is well known that the drawing scope of ruler and compass construction has limitations. To draw more complicated diagrams, we need the method of automated geometry diagram generation (AGDG), which 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 [18].

In MMP/Geometer, by combining the idea of dynamic geometry and AGDG we obtain what we called the *intelligent dynamic geometry*, which can be used to input and manipulate diagrams more easily. It can be used to manipulate geometric diagrams interactively as dynamic geometry software and does not have the limitation of ruler and compass construction.

The rest of the paper is organized as follows. Section 2 describes the input formats of a geometric statement. Section 3 reports the implementation of methods of AGTP. Section 4 reports the implementation of the methods of AGDG.

2 Input of a Geometric Statement

The user may input a geometric statement to MMP/Geometer in two ways: either by typing the English description of the statement or by drawing the diagram of the statement on a computer screen with a mouse. After a statement is inputted, it may be described in four forms: the *natural language form*, the *constructive form*, the *predicate form*, and the *algebraic form*. The purpose of using different input forms is that each input form has its merit. For instance, the natural language input may be the favorable choice for high school students. The constructive form is the input form for several proving methods, like the area method[8]. The predicate form is the most general way of describing a statement. We use Simson's Theorem to illustrate these forms.

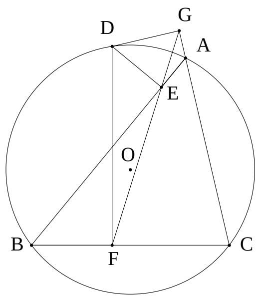

Figure 1: Simson's Theorem

Natural Language. MMP/Geometer accepts geometric statements described with a precisely defined pseudo-natural language. Detailed definition of this language may be found in [12]. MMP/Geometer may convert a statement in natural language into all other forms. Using this language, Simson's Theorem may be described as follows.

```
geom("Example Simson. Let D be a point on the circumcircle O of
triangle ABC. E is the foot from point D to line AB. F is the foot
from point D to line BC. G is the foot from point D to line AC.
Show that points E, F, and G are collinear.");
```

In the above example, `geom` is an MMP/Geometer command which accepts a geometric statement in any form and converts it to all other possible forms.

Constructive Form. In this form, the geometric objects in the statements can be described by a sequence of geometric constructions. A construction is to generate a new geometric object with lines and circles. Detailed description of the constructions used in MMP/Geometer may be found in [2, 5, 12]. A statement in constructive form may be converted to all other forms. The constructive form of Simson's Theorem is as follows.

```
geom([[[POINT,A,B,C],[CIRCUMCENTER,O,A,B,C],[ON,D,[CIR,O,A]], [FOOT,E,D,A,B],
[FOOT,F,D,B,C],[FOOT,G,D,A,C]], [[coll,E,F,G]]]);
```

Generally speaking, a statement in constructive form is a pair $[cs, c]$, where cs is a construction sequence used to generate the geometric objects in a statement and c is the set of conclusions.

Predicate Form. This is a natural way to describe a geometric statement. The hypotheses and conclusions are represented by geometric predicates. The following predicate form for Simson's Theorem is generated automatically from its constructive form mentioned above with MMP/Geometer.

```
geom([[y5,x5,y4,x4,y3,x3,y2,x2,y1,x1,v2,u1,v1],[],
[A,[0,0],B,[0,v1],C,[u2,v2],O,[x1,y1],D,[x2,y2],E,[x3,y3],F,[x4,y4],G,[x5,y5]],
[[cong,O,A,O,B],[cong,O,A,O,C],[cong,O,D,O,A],
 [coll,E,A,B],[perp,E,D,A,B],[coll,F,B,C],[perp,F,D,B,C],
 [coll,G,A,C],[perp,G,D,A,C]],
[[sqdis,A,B],[sqdis,B,C],[sqdis,A,C]],[[COLL,E,F,G]]]);
```

Predicate [sqdis,A,B] means $|OA|^2 = 0$. The three predicates $|AB| \neq 0, |BC| \neq 0, |AC| \neq 0$ are called the *non-degenerated conditions* (ndgs) for Simson's theorem.

A statement in predicate form can be represented by a 6-tuple: $[mv, pv, pset, ps, ds, c]$ where mv and pv are the main variables and parametric variables, $pset$ is the set of points and their coordinates, ps is a set of predicates representing the hypotheses, ds is a set of predicates representing the non-degenerate conditions, c is the set of conclusions.

Algebraic Form. In this form, coordinates are assigned to points in the statement and the hypotheses and conclusions are represented by algebraic equations. It is straight forward to convert a statement in predicate form to algebraic form. The following is the algebraic form of Simson's theorem in predicate form given above.

```
geom([[y5,x5,y4,x4,y3,x3,y2,x2,y1,x1,v2,u1,v1],[],
[2*v1*y1-v1^2,2*v2*y1+2*u2*x1-v2^2-u2^2,
y2^2-2*y1*y2+x2^2-2*x1*x2,-v1*x3,
-v1*y3+v1*y2,u2*y4-v2*x4+v1*x4-u2*v1,
-v2*y4+v1*y4-u2*x4+v2*y2-v1*y2+u2*x2,u2*y5-v2*x5,
-v2*y5-u2*x5+v2*y2+u2*x2],
[v1^2,v2^2-2*v1*v2+v1^2+u2^2,v2^2+u2^2],
[x4*y5-x3*y5-y4*x5+y3*x5+x3*y4-y3*x4]]);
```

A statement in algebraic form can be represented by a 5-tuple: $[mv, pv, ps, ds, c]$ where mv and pv are the main variables and parametric variables, ps is a set of equations representing the hypotheses, ds is a set of equations representing the degenerate conditions, c is the set of conclusions.

The relations between the four representation forms are illustrated in Figure 2. An arrow from a form A to a form B means that MMP/Geometer can convert A to B. We use the method in [5] to convert a constructive form to a predicate form. The arrow marked with AGDG means that we will use AGDG methods reported in Section 4 to convert a predicate form to a constructive form. All other conversions are either obvious or can be found in [12].

A statement $G = [mv, pv, ps, ds, c]$ in algebraic form is said to be *generally valid* if there exists a polynomial d in the variables pv such that

$$\forall x \in mv, \forall u \in pv[(ps = 0 \wedge ds \neq 0 \wedge d \neq 0) \Rightarrow c = 0]$$

If $ds = \emptyset$, then G is valid if

$$\forall c \in mv[(ps = 0 \wedge ds \neq 0) \Rightarrow c = 0]$$

Since statements in all other forms can be converted into statements in algebraic form, we say a statement is generally valid or valid if its algebraic form is generally valid or valid.

There are two ways to input a geometric statement graphically.

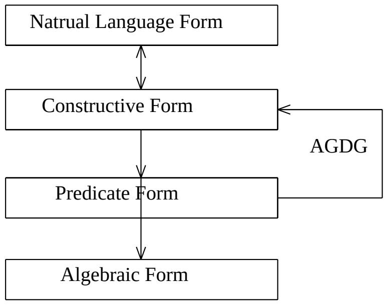

Figure 2: The relations between the four representation forms

Constructive. To input a statement constructively, we need to draw the diagram of the statement in a way similar to the ruler and compass construction. More precisely, we need to draw the objects in the statement sequentially with lines and circles. The drawing process can be naturally converted into a constructive description of the statement. For instance, to draw the diagram of Simson's Theorem, we may just follow its constructive description given above to draw points A, B, C, O, D, E, F, G sequentially with functions provided by the software.

Declarative. In this form, we will first draw a sketch of the diagram and then add the geometric conditions. The software will adjust the diagram automatically such that these conditions will be satisfied. This process is called *automated geometric diagram generation* (AGDG). The declarative drawing process can be naturally turned into a predicate description of the statement.

One way to draw a diagram in declarative form is to transform it into constructive form automatically. Of course, not all statements in declarative form can be transformed into constructive form. For this kind of problems, we need to develop more complicated methods to draw their diagrams. This is the main task of AGDG, which will be introduced in Section 4.

To draw the diagram of Simson's Theorem in this way, we may first draw four points A, B, C, D on a circle and draw lines DE, DF, and DG such that E, F, G are on line AB, BC, AC respectively. Now we have a sketch of the diagram. Finally, we add the conditions $DE \perp AB$, $DF \perp BC$, $DG \perp AC$ to the sketch and the software will adjust the sketch to satisfy these conditions.

3 Automated Geometric Theorem Proving and Discovering

3.1 Wu's Method

Wu's method is a coordinate-based method for *equational geometric statements*, that is, the premise and the conclusion of the statement can be represented by algebraic or differential equations. The method first transfers geometric conditions into polynomial or differential equations, then deals with the polynomial equations with the characteristic set method [31, 34]. Variants of this method may be found in [2, 25, 28, 35].

Wu's method is based on Wu-Ritt's zero decomposition theorem for polynomial and differential polynomial equations [31, 34]. It may be used to represent the zero set of a polynomial equation system as the union of zero sets of equations in *triangular form*, that is, equation systems like

$$f_1(u, x_1) = 0, f_2(u, x_1, x_2) = 0, \ldots, f_p(u, x_1, \ldots, x_p) = 0$$

where the u could be considered as a set of parameters and the x are the variables to be determined. Let PS and DS be two sets of polynomials or differential polynomials, $\mathrm{Zero}(PS/DS)$ the set of solutions of $PS = 0$ over the field of complex numbers, which do not vanish any of the equations in DS. Wu-Ritt's zero decomposition theorem [31, 28] may be stated as follows:

$$\mathrm{Zero}(PS/DS) = \cup_i \mathrm{Zero}(\mathbf{SAT}(AS_i)/DS) \tag{1}$$

where AS_i are polynomial sets in triangular form and $\mathbf{SAT}(AS_i)$ are the saturation ideals of AS_i. Variants of this algorithm may be found in [1, 2, 23, 24, 28, 35]. In MMP, the zero decomposition is implemented by Dingkang Wang [30].

Four versions of Wu's method are implemented in MMP/Geometer.

WU-C. This method can be used to decide whether a geometric statement in constructive form is *generically true*, or it is true except some special cases called non-degenerate conditions (ndgs). The implementation is based on a variant of Wu's method described in [5]. With this method, we may prove or dis-prove a geometric statement and give *sufficient* ndgs in geometric form. By sufficient ndgs, we mean

(1) if the statement under consideration is valid in geometry textbooks (generically true), then the statement is valid under these conditions;

(2) if the statement is not valid under these conditions, then it will not become valid by adding more conditions unless the newly added conditions make the statement trivially valid or change the meaning of the statement.

We use Simson's Theorem as an example.

```
wcprove("Example Simson. Let D be a point on the circumcircle O of
triangle ABC. E is the foot from point D to line AB. F is the foot
from point D to line BC. G is the foot from point D to line AC.
Show that points E, F, and G are collinear.");
```

MMP/Geometer first converts the statement into *constructive form*, then proves the constructive statement and gives the following output.

1. The statement is generically true, that is, it is true except some special cases.
2. The sufficient ndgs are: $\neg[\text{coll}, A, B, C], |AB| \neq 0, |BC| \neq 0, |AC| \neq 0$.

WU-G. This method decides whether a geometric statement in *algebraic form* is valid [31, 34].

Let PS and DS be the equation part and inequation part corresponding to the ndgs of a statement. MMP/Geometer will first use Wu-Ritt's zero decommission theorem to find triangular sets AS_i as in (1). Let $C = 0$ be the conclusion. If $\text{prem}(C, AS_i) = 0$, then $C = 0$ is valid on $\text{Zero}(\mathbf{SAT}(AS_i))$. If AS_i is irreducible, then this is also a necessary condition over the field of complex numbers.

For Simson's Theorem, we may prove that its following predicate form is valid.

```
wprove([[y5,x5,y4,x4,y3,x3,y2,x2,y1,x1,v2,u1,v1],[],
[A,[0,0],B,[0,v1],C,[u2,v2],O,[x1,y1],D,[x2,y2],E,[x3,y3],F,[x4,y4],G,[x5,y5]],
[[cong,O,A,O,B],[cong,O,A,O,C],[cong,O,D,O,A],[coll,E,A,B],
[perp,E,D,A,B],[coll,F,B,C],[perp,F,D,B,C],[coll,G,A,C],[perp,G,D,A,C]],
[[sqdis,A,B],[sqdis,B,C],[sqdis,A,C]],[[COLL,E,F,G]]]);
```

Note that the result obtained here is stronger than that obtained with method WU-C: one ndg condition $\neg[\text{coll}, A, B, C]$ is removed from the description.

WU-D. An advantage of Wu's method is that it can be used to prove differential geometry theorems and mechanics [33]. The following is an example.

Example Kepler-Newton. Prove Newton's gravitational laws using Kepler's laws. Kepler's first and second laws can be described as follows.

K1. Each planet describes an ellipse with the sun in one focus.

K2. The radius vector drawn from the sun to a planet sweeps out equal areas in equal times.

These laws can be expressed as the following differential equations.

$$h_1 = r^2 - x^2 - y^2 = 0$$
$$h_2 = a^2 - x''^2 - y''^2 = 0$$
$$k_1 = r - p - ex = 0 \wedge p' = 0 \wedge e' = 0$$
$$k_2 = x''y - y''x = 0.$$

Newton's law can be expressed as $n_1 = (ar^2)' = 0$. Then the problem is to prove

$$(h_1 = 0 \wedge h_2 = 0 \wedge k_1 = 0 \wedge k_2 = 0) \Rightarrow n_1 = 0.$$

With MMP/Geometer, it is proved that the above statement is false. When add a ndg condition $p \neq 0$ (the ellipse does not becomes a straightline), we may use MMP/Geometer to prove the statement.

```
depend([a,r,y,x],[t]);
wdprove([[a,r,y,x,p,e],[],[],
[r^2-x^2-y^2,  a^2-x[2]^2-y[2]^2, x*y[2]-x[2]*y, r-p-e*x],
[p],
[diff(a*r^2,t)]]);
```

Command *depend*$([a, r, y, x], [t])$ defines a, r, y, x as functions in t.

The following command proves that "A space curve C satisfies $t = k' = 0$ is a circle."

```
curve();
wprove_curve([[],[],[],[t,diff(k,s)],[],[[FIX_PLANE,C],[FIX_SPHERE,C]]]);
```

The curve C, its arc length s, its torsion t and curvature k are defined in command curve(). diff(k, s) is the differentiation of k with s. There are two conclusions: [FIX_PLANE,C] meaning that C is on a plane and [FIX_SPHERE,C] meaning that C is on a sphere.

WU-F. As pointed out by Wu [32], Wu-Ritt's zero decomposition theorem can be used to *discover* geometric relations automatically. We use the following example to illustrate this.

Example Heron-Qin Formula. Find the formula for the area of a triangle ABC in terms of its three sides. This problem may be solved with MMP/Geometer as follows

```
wderive(
[[x,y,k],[a,b,c],[B,[0,0], C,[a,0], A, [x,y]],
 [[dis, A, C, b],  [dis, A, B, c],[area, k, A, B, C]], [], []]);
```

The above command will find the following relation between the last main variable k and the parameters: a, b, c automatically:

$$16 * k^2 + a^4 - 2 * b^2 * a^2 - 2 * c^2 * a^2 + b^4 - 2 * c^2 * b^2 + c^4 = 0.$$

In general, we may formulate the problem as follows. Let $\mathcal{U} = \{u_1, \cdots, u_m\}$ be a set of parameters and $\mathcal{X} = \{x_1, \cdots, x_n\}$ a set of dependent variables. The relation between the u_i and the x_j is given by a set of algebraic or differential equations and inequations:

$$P_1(\mathcal{U},\mathcal{X}) = 0, \cdots, P_s(\mathcal{U},\mathcal{X}) = 0$$
$$D_1(\mathcal{U},\mathcal{X}) \neq 0, \cdots, D_r(\mathcal{U},\mathcal{X}) \neq 0.$$

The problem is to find the relation between x_1 and $\mathcal{U}$, which can be done with Wu-Ritt zero decomposition under the following order $u_1 < u_2 \cdots < u_m < x_1$.

3.2 The Area Method

This method uses high-level geometric invariants such as the area and the Pythagorean difference as the basic tool of proving geometric theorems [8]. Instead of eliminating variables as in Wu's method, the area method eliminates points from geometric invariants directly. The advantage is that short and human-readable proofs for geometric statements could be produced. This method works for constructive geometric statements.

The *signed area* S_{ABC} of triangle ABC is the usual area with a sign depending on the order of the three vertices of the triangle. The following properties of the signed area is clearly true.

A1 Let M be the intersection of two non-parallel lines AB and PQ and $Q \neq M$. Then $\frac{\overline{PM}}{\overline{QM}} = \frac{S_{PAB}}{S_{QAB}}$.

A2 If $A \neq B$, $PQ \parallel AB$ iff $S_{PAB} = S_{QAB}$.

Let us consider the following example.

```
aprove("Example Parallelogram. Let ABCD be a parallelogram.
O is the intersection of diagonals AC and BD.
Show that O is the midpoint of AC.");
```

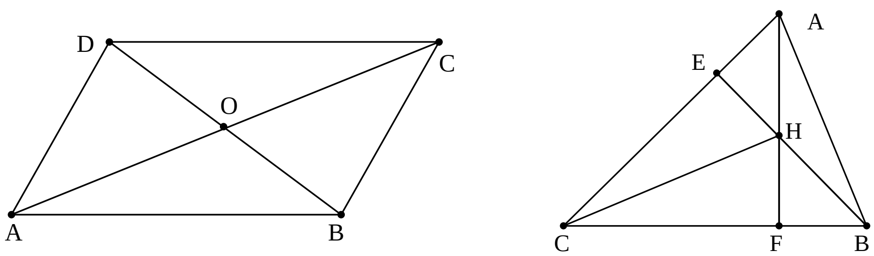

Figure 3: Parallelogram Theorem and Orthocenter Theorem

MMP/Geometer produces the following proof for the parallelogram theorem, which is with clear geometric meaning. The comments on the right hand side is added by the authors.

$\frac{AO}{OC} = \frac{S_{ABD}}{S_{BCD}}$ By A1.

$= \frac{S_{ABC}}{S_{ABC}} = 1.$ By A2, $S_{ABD} = S_{ABC}$, $S_{BCD} = S_{ABC}$.

To deal with perpendicularity, we need to define the *Pythagorean difference*. For points A, B, and C, the *Pythagorean difference*, P_{ABC}, is defined to be $P_{ABC} = \overline{AB}^2 + \overline{CB}^2 - \overline{AC}^2$. The following Pythagorean theorem is taken as a basic (unproven) property of the Pythagorean difference.

P1 $AB \bot PQ$ iff $P_{PAB} = P_{QAB}$.

Let us consider the following example.

```
aprove("Example Orthocenter. ABC is a triangle. F is the foot from A to BC.
E is the foot from B to AC. H is the intersection of lines AF and BE.
Show that CH is perpendicular to AB.");
```

By P1, we need only to show $P_{ACH} = P_{BCH}$. MMP/Geometer produces the following proof for the Orthocenter theorem.

$$\frac{P_{ACH}}{P_{BCH}} = \frac{P_{ACB}}{P_{BCA}} = \frac{P_{ACB}}{P_{ACB}} = 1.$$

By P1, we can eliminate point H as follows: $P_{ACH} = P_{ACB}$; $P_{BCH} = P_{BCA}$.

The following is the machine proof for Simson's Theorem. Point $G_1 = EF \cap AC$ is added by MMP/Geometer automatically. We need only to prove $\frac{\overline{AG}}{\overline{CG}} = \frac{\overline{AG_1}}{\overline{CG_1}}$. Details on this proof may be found in [8]

The machine proof

$$\left(\frac{\overline{AG}}{\overline{BG}}\right)/\left(\frac{\overline{AG_1}}{\overline{BG_1}}\right)$$

$$\stackrel{G_1}{=} \frac{S_{BEF}}{S_{AEF}} \cdot \frac{\overline{AG}}{\overline{BG}}$$

$$\stackrel{G}{=} \frac{P_{BAD} \cdot S_{BEF}}{S_{AEF} \cdot (-P_{ABD})}$$

$$\stackrel{F}{=} \frac{-P_{BAD} \cdot P_{ACD} \cdot S_{ABE} \cdot P_{ACA}}{(-P_{CAD} \cdot S_{ACE}) \cdot P_{ABD} \cdot P_{ACA}}$$

$$\stackrel{simplify}{=} \frac{P_{BAD} \cdot P_{ACD} \cdot S_{ABE}}{P_{CAD} \cdot S_{ACE} \cdot P_{ABD}}$$

$$\stackrel{E}{=} \frac{P_{BAD} \cdot P_{ACD} \cdot P_{CBD} \cdot S_{ABC} \cdot P_{BCB}}{P_{CAD} \cdot (-P_{BCD} \cdot S_{ABC}) \cdot P_{ABD} \cdot P_{BCB}}$$

$$\stackrel{simplify}{=} \frac{P_{BAD} \cdot P_{ACD} \cdot P_{CBD}}{-P_{CAD} \cdot P_{BCD} \cdot P_{ABD}}$$

$$\stackrel{co-cir}{=} \frac{(2\widetilde{AD} \cdot \widetilde{AB} \cdot \cos(BD)) \cdot (-2\widetilde{CD} \cdot \widetilde{AC} \cdot \cos(AD)) \cdot (2\widetilde{BD} \cdot \widetilde{BC} \cdot \cos(CD))}{-(2\widetilde{AD} \cdot \widetilde{AC} \cdot \cos(CD)) \cdot (-2\widetilde{CD} \cdot \widetilde{BC} \cdot \cos(BD)) \cdot (-2\widetilde{BD} \cdot \widetilde{AB} \cdot \cos(AD))}$$

$$\stackrel{simplify}{=} 1$$

The eliminants

$$\frac{\overline{AG_1}}{\overline{BG_1}} \stackrel{G_1}{=} \frac{S_{AEF}}{S_{BEF}}$$

$$\frac{\overline{AG}}{\overline{BG}} \stackrel{G}{=} \frac{P_{BAD}}{-P_{ABD}}$$

$$S_{AEF} \stackrel{F}{=} \frac{-P_{CAD} \cdot S_{ACE}}{P_{ACA}}$$

$$S_{BEF} \stackrel{F}{=} \frac{P_{ACD} \cdot S_{ABE}}{P_{ACA}}$$

$$S_{ACE} \stackrel{E}{=} \frac{-P_{BCD} \cdot S_{ABC}}{P_{BCB}}$$

$$S_{ABE} \stackrel{E}{=} \frac{P_{CBD} \cdot S_{ABC}}{P_{BCB}}$$

$$P_{ABD} = -2(\widetilde{BD} \cdot \widetilde{AB} \cdot \cos(AD))$$

$$P_{BCD} = -2(\widetilde{CD} \cdot \widetilde{BC} \cdot \cos(BD))$$

$$P_{CAD} = 2(\widetilde{AD} \cdot \widetilde{AC} \cdot \cos(CD))$$

$$P_{CBD} = 2(\widetilde{BD} \cdot \widetilde{BC} \cdot \cos(CD))$$

$$P_{ACD} = -2(\widetilde{CD} \cdot \widetilde{AC} \cdot \cos(AD))$$

$$P_{BAD} = 2(\widetilde{AD} \cdot \widetilde{AB} \cdot \cos(BD))$$

3.3 Deductive Database Method

For a given geometric diagram, MMP/Geometer can generate a geometric deductive database (GDD) which contains all the properties of this diagram that can be deduced from a fixed set of geometric axioms, and for each geometric property in the database, MMP/Geometer can generate a proof in traditional style [9].

Let D_0 be the premise of a geometric statement and R the set of geometric axioms or rules. We may use the *breadth-first forward chaining method* to find new properties of the corresponding diagram. Basically speaking, the method works as follows

$$\boxed{D_0} \stackrel{R}{\subset} \boxed{D_1} \stackrel{R}{\subset} \ldots \stackrel{R}{\subset} \boxed{D_k} \quad \text{(Fixpoint)}$$

where D_{i+1} is the union of D_i and the set of new properties obtained by applying rules in R to properties in D_i. If at certain step $D_k = D_{k+1}$, i.e.,

$$R(D_k) = D_k,$$

then we say that a *fixpoint* (of reasoning) for D_0 and R is reached.

The naive form of breadth-first forward chaining is notorious for its inefficiency. But, in the case of geometry reasoning, by introducing new data structure and search techniques, we manage to build a very effective prover based on this idea [9].

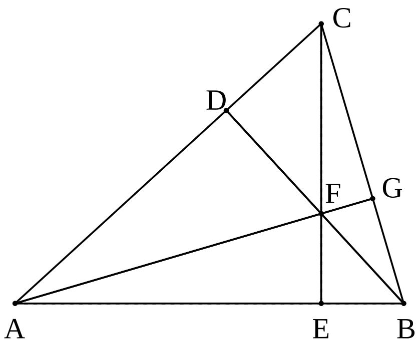

Figure 4: Orthocenter Theorem

The GDD method may take a statement in predicate, constructive, or natural language form as input. Let us consider the following form of the Orthocenter Theorem (Figure 4).

```
prove_gdd("Example Orthocenter. ABC is a triangle. D is the foot
from B to AC. E is the foot from C to AB. F is the intersection of
lines BD and CE. G is the intersection of lines BC and AF.");
```

The initial database is the hypotheses: $D, A, C; E, A, B; F, B, D; F, C, E; G, B, C; G, A, F$ are collinear sets, $BD \perp AC, CE \perp AB$. The fixpoint contains 151 geometry properties:

6 collinear point sets,
3 perpendicular pairs,
6 co-cyclic points sets,
24 equal angle pairs,
7 similar triangles sets,
105 and equal ratio pairs.

The forward chaining is a natural way of discovering properties for a given geometric configuration. Any thing obtained in the forward chaining may be looked as a "new" result. Take the simple configuration (Figure 4) related to the orthocenter theorem as an example. MMP/Geometer has discovered the most often mentioned properties about this configuration: $AG \perp BC$ (the orthocenter theorem) and $\angle EGA = \angle AGD$. The fixpoint also contains seven sets of similar triangles

$\triangle DBA \sim \triangle DCF \sim \triangle EBF \sim \triangle ECA$;
$\triangle DCB \sim \triangle DFA \sim \triangle GFB \sim \triangle GCA$;
$\triangle EFA \sim \triangle EBC \sim \triangle GBA \sim \triangle GFC$;
$\triangle FBC \sim \triangle FED \sim \triangle GBD \sim \triangle GEC$;
$\triangle ACB \sim \triangle AED \sim \triangle GCD \sim \triangle GEB$;
$\triangle CED \sim \triangle CAF \sim \triangle GAD \sim \triangle GEF$;
$\triangle FBA \sim \triangle EBD \sim \triangle FGD \sim \triangle EGA$.

For each geometric property in the database, MMP/Geometer can produce a proof of traditional style. The following is the proof for the Orthocenter Theorem ($AG \perp BC$) by MMP/Geometer. Notice that the proof is in an "analysis style", i.e., it starts from the conclusion and goes all the way to the hypotheses of the statement.

1. $AG \perp BC$,
   because $AC \perp BD$(hypothesis), (2)$\angle[AC, BD] = \angle[BC, AF]$.
2. $\angle[AC, BD] = \angle[BC, AF]$,
   because (3)$\angle[AC, BC] = \angle[BD, AF]$. (This is rule in MMP/Geometer).
3. $\angle[CA, CB] = \angle[BD, AF]$,
   because (4)$\angle[CA, CB] = \angle[DE, AB]$, (5)$\angle[BD, AF] = \angle[DE, AB]$.
4. $\angle[CA, CB] = \angle[DE, AB]$,
   because (6)co-cyclic$[B, D, C, E]$.

5. $\angle[BD, AF] = \angle[DE, AB]$,
   because (7)co-cyclic$[A, D, E, F]$.
6. co-cyclic[B,D,C,E],
   because $DC \perp DB$(hypothesis), $EC \perp EB$(hypothesis).
7. co-cyclic$[A, D, E, F]$,
   because $DF \perp DA$(hypothesis), $EF \perp EA$(hypothesis).

The first step of the proof can be understood as follows. $AG \perp BC$ is true because $AC \perp BD$ which is a hypothesis and $\angle[AC, BD] = \angle[BC, AF]$ which will be proved in the second step. The other steps can be understood similarly.

One of the largest databases obtained with MMP/Geometer is for the nine-point circle theorem (Figure 5(a)), which contains 6019 geometric relations. In predicate form, the database contains 6646428 predicates.

```
prove_gdd("Example Nine-point-circle. ABC is a triangle. K is the
midpoint of BC.  L is the midpoint of CA.  M is the midpoint of AB.
D is the foot from B to AC.  E is the foot from C to AB. F is the
intersection of lines BD and CE. G is the intersection of lines BC and
AF. H is the midpoint of AF.  I is the midpoint of BF.  J is the
midpoint of  CF.Show that K, G, J, D are cocircle.");
```

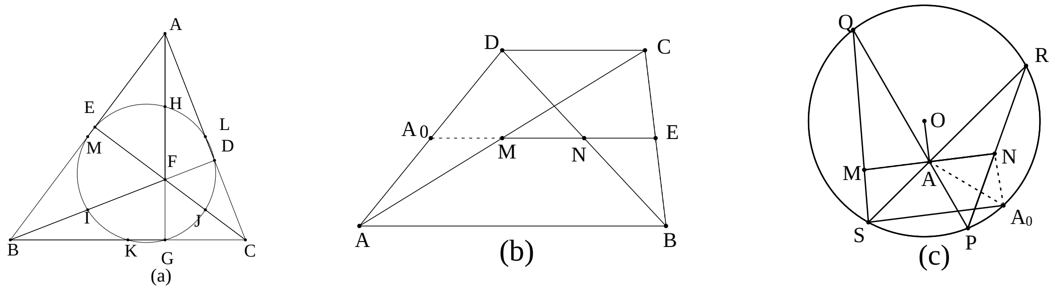

Figure 5: Three Geometric Theorems

Constructing new points or lines is one of the basic methods for solving geometric problems. One advantage of MMP/Geometer is that it can automatically add auxiliary points to prove a geometric statement if needed [9]. The following example is shown in Figure 5(b).

```
prove_gdd("Example Trapezoid. Let ABCD be a trapezoid. M is the midpoint
of AC. N is the midpoint of BD. E is the intersection of lines MN and BC.
Show that E is the midpoint of BC.");
```

The conclusion is not in the first fixpoint. The program then automatically adds an auxiliary point A_0 which is the midpoint of AD. With this auxiliary point, MMP/Geometer generates a fixpoint containing the conclusion.

The Butterfly theorem in Figure 5(c) also needs auxiliary points.

```
prove_gdd("Example Butterfly-Theorem. P, Q, R, and S are four points on
a circle O.  A is the intersection of lines PQ and SR. N is the
intersection of line PR and the line passing through A and
perpendicular to OA.  M is the intersection of line QS and the line
passing through A and perpendicular to OA.  Show that A is the
midpoint of NM.");
```

For this problem, MMP/Geometer automatically adds an auxiliary point A_0 which is the intersection of the line passing through S and parallel to AN and the circle O. With this point, MMP/Geometer generates a fixpoint which contains the conclusion.

Wu's Method		Area Method		GDD Method	
Example	Time	Example	Time	Example	Time
Simson (WU-C)	20	Parallelogram	10	Orthocenter	110
Simson (WU-G)	80	Orthocenter	10	Trapezoid	50
Kepler-Newton	3200	Simson	30	Butterfly	110
Curve	2100			Nine-point	560(s)
Herron-Qin	10				

Table 1: Statistics for the examples

Methods	No. of Theorems	Source
WU-C	512	[2]
WU-G	450	[3]
WU-D	100	[6]
WU-F	120	[4]
AREA	400	[8]
DBASE	170	[9]

Table 2: Theorems proved Wu's method, area method and GDD method

3.4 Experimental Results

MMP is developed with VC in a Windows environment. Current version may be found in the webpage listed in [12]. Table 1 contains the running times for the examples in this Section 3. The data is collected on a PC compatible with 1.5G CPU. The time is in microseconds.

Why do we implement more than one method in MMP/Geometer? First, each method has its advantages and shortcomings. Generally speaking, the proving power of the methods are as follows

$$\text{WU-C} > \text{WU-G} > \text{AREA} > \text{DBASE}.$$

Table 2 gives the number of theorems proved with these methods. When considering to produce elegant and human-understandable proofs, the order is reversed.

Second, with these methods, for the same theorem, the prover can produce a variety of proofs with different styles. This might be important in using MMP/Geometer to geometry education, since different methods allow students to explore different and better proofs.

Only a selective set of examples shown in Table 2 was tested in MMP/Geometer. We expect that almost all of the theorems mentioned in the above table can be proved with MMP/Geometer since the methods used in MMP/Geometer are improved version for the original methods used to prove them.

4 Automated Geometric Diagram Generation

4.1 Dynamic Geometry

By *dynamic geometry*, we mean computer generated geometric objects which could be changed dynamically. Generally, we may perform the following operations: dynamic transformation, dynamic measurement, free dragging, and animation. By doing dynamic transformations 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 for figures of functions.

Most dynamic geometry software systems [21, 20, 13] use construction sequences of lines and circles to generate diagrams. Since such sequences are easy to compute, dynamic geometry software systems are usually very fast.

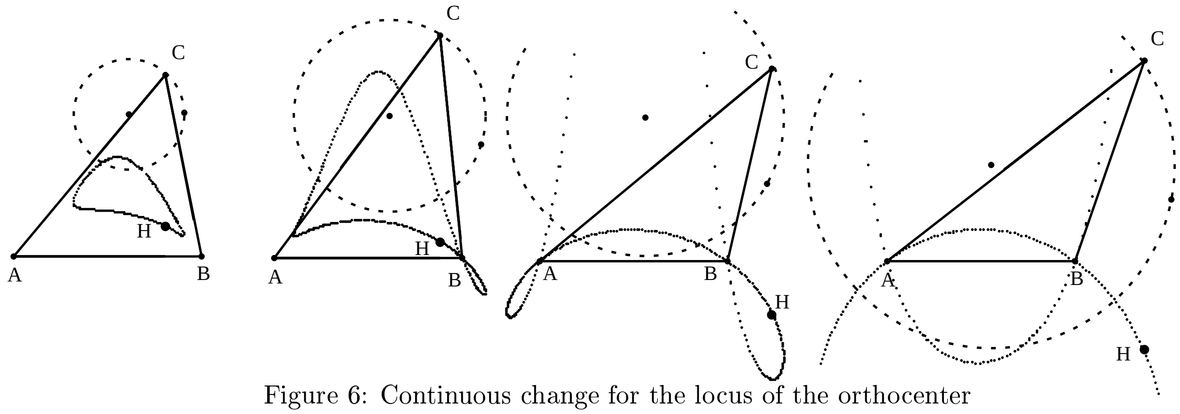

Figure 6: Continuous change for the locus of the orthocenter

As an example, let H be the orthocenter of triangle ABC. We fix points A and B and let point C move 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, $A = (0, 0)$, and $B = (d, 0)$. With Wu's method of mechanical formula derivation, we may obtain the equation of this locus

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

with MMP/Geometer as follows.

```
wderive([[y3,x3,y2,x2,y1,x1,y],[x,d,x0,y0,r],
[A,[0,0],B,[d,0],O,[x0,y0],C,[x1,y1],H,[x,y]],
[[dis,C,O,r],[perp,A,H,B,C],[perp,B,H,A,C]],[],[]]);
```

But from this equation, we still do not know the shape of the curve. With MMP/Geometer, we can draw the diagram of this curve as the locus of point H. By continuously changing the radius of circle c, we may observe the shape changes of the curve (Figure 6).

The diagram in Figure 7 is to generate the *locus of the moon* when the moon rotates around the earth on a circle and the earth rotates around the sun on an ellipse. For its description, please consult [12].

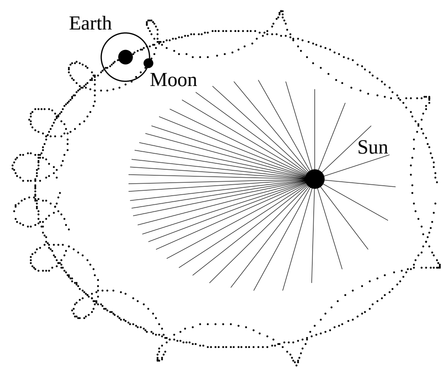

Figure 7: Locus of the moon

In dynamic geometry, the drawing process is based on the *constructive description* for a diagram, that is, each geometric object is constructed with ruler and compass. In a construction sequence, free and semi-free points could be dragged. The computation can be carried out in real-time because the equations raised from a construction sequence are almost in triangular form: variables are introduced at most two by two. Further, only equations with degree less than or equal to two are involved. MMP/Geometer has all the above functions of dynamic geometry.

4.2 Intelligent Dynamic Geometry

Most geometric diagrams in geometry textbooks are described declaratively, and the task of converting such a description to constructive form is usually done by human. For some diagrams, to find a constructive solution is quite difficult, and for more diagrams ruler and compass construction is impossible. In MMP/Geometer, by combining the idea of dynamic geometry and AGDG we implement an *intelligent dynamic geometry* software system, which can be used to input and manipulate diagrams more easily.

There are two major steps to draw a declaratively given diagrams. First, we try to find a ruler and compass construction by mechanizing some of the techniques of ruler and compass construction developed since the time of ancient Greek. If we fail to do so, general AGDG methods are used to draw the diagram.

To find a ruler and compass construction for a geometric diagram in predicate form, we first transform the geometric relations into a graph and then solve it in two steps.

1. We repeatedly remove those geometric objects that can be constructed explicitly until nothing can be done. This simple algorithm is linear [11] and solves about eighty percent of the 512 problems reported in [2].
2. If the above step fails, we use Owen and Hoffmann's triangle decomposition algorithms [18, 27] to reduce the problem into the solving of triangles.
3. If the above step fails, we use certain *geometric transformations* to solve the problem[11]. This is a quadratic algorithm and is complete for drawing problems of simple polygons.

To see the power of the methods, let us look at the example in Figure 8, where each line segment represents a given distance between its two end points. We have twenty two distances. In order to solve the problems algebraically, we need to solve an equation system consisting of twenty two quadratic equations. But with the triangle decomposition [18, 27], the problem can be reduced to solving of triangles, as shown in Figures 8(b) and (c). As a consequence, the problem is ruler and compass constructible.

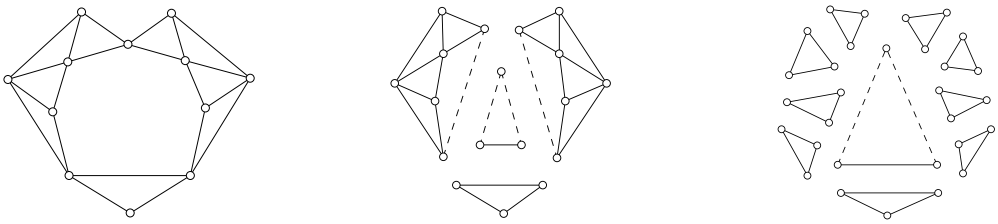

Figure 8: A diagram with 22 given distances

As another example, let us draw the quadrilateral in Figure 9, which cannot be solved with the triangle decomposition method. This diagram cannot be drawn with ruler and compass explicitly. To solve it, MMP/Geometer adds a parallelogram $P_1P_2P_3P_2'$ [11]. Since $\angle P_4P_1P_2'$, $|P_1P_2'|$, $|P_3P_2'|$ are known, it is easy to find a construction sequence for quadrilateral $P_3P_4P_1P_2'$. Point P_2 can be constructed easily.

Figure 9: Lengths of four edges and angle (L_2, L_4) are given

Using methods of AGDG allows us to have more power to manipulation the diagram. If a construction sequence for a diagram has been given, we may only drag the free and semi-free points in the diagram. This drawback may be overcomed as follows. Suppose that we want to drag a point, we may re-generate a new construction sequence in which the point is a free point. This kind of dragging is called *free dragging*.

Let us consider the diagram in Figure 4. A construction order for the points in this diagram is as follows

$$A, B, C, E, D, G, F$$

where A, B, C are free points. F is the intersection of lines BD and CE, hence a fixed point. In dynamic geometry software, we cannot drag this point. But in MMP/Geometer, when a user wants to drag this point, a new construction sequence

$$F, A, B, E, D, G, C$$

is automatically generated, in which F is a free point and can be dragged.

If we fail to find a ruler and compass construction for a problem, MMP/Geometer will try to solve the problem with the following algorithms.

1. Graph analyses methods are used to decompose the problem into *generalized construction sequence* [22]:

   $$C_1, C_2, \cdots, C_m$$

   where C_i are sets of geometric objects such that

   - C_i can be constructed from $\cup_{k=1}^{i-1} C_k$.
   - and C_i is the smallest set satisfying the above condition.

2. Compute the position of C_i from $C_1, \cdots, C_{i-1}$. MMP/Geometer uses two methods to do this.

   Optimization Method The problem is converted to solving a set of algebraic equations:

   $$f_1(\mathcal{X}) = 0, \cdots, f_m(\mathcal{X}) = 0$$

   where $\mathcal{X}$ is a set of variables. Let

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

   We use the BFGS method to find a position $\mathcal{X}_0$ such that $\sigma(\mathcal{X}_0)$ is a minimal value [16]. If $\sigma(\mathcal{X}_0) = 0$ then $\mathcal{X}_0$ is a set of solutions for original equation system.

   Locus Intersection Method (LIMd) The above method based on optimization can find one solution only. If we want to find all the solutions, we may use the LIMd method [14]. The LIMd method is a hybrid method to find all solutions to geometric problems.

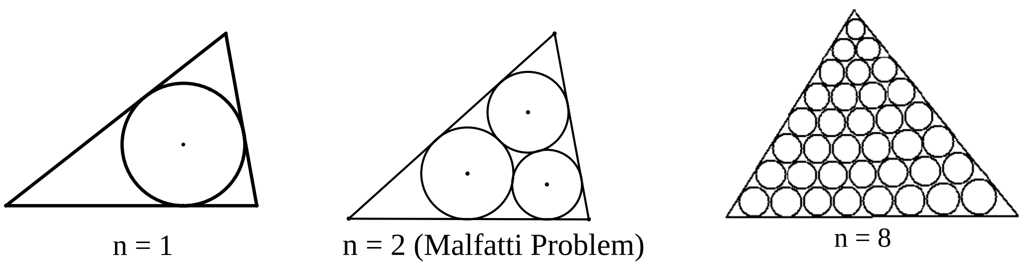

Figure 10: Packing circles into a triangle

Let us consider a problem of packing circles into a fixed triangle. The problem is to pack $n(n+1)/2$ circles (n rows of circles) tangent to adjacent circles and the adjacent neighboring sides of a given triangle. Figure 10 is the cases for $n = 1, 2, 8$. The most difficult case is $n = 8$, in which we need to solve 24 quartic equations and 84 quadratic equations simultaneously. This equation system cannot be simplified essentially. Table 3 shows the running times for different n of this problem on a PC with CPU 1.5G. It seems that problems of large size can be solved in real time.

# circles (# rows)	*# quartic equations*	*# quadratic equations*	*time*
3 (2)	6	3	0.18
6 (3)	9	9	0.50
10 (4)	12	18	1.72
15 (5)	15	30	1.93
21 (6)	18	45	4.15
28 (7)	21	63	10.92
36 (8)	24	84	15.45

Table 3: Running statistics for the tangent packing problems

5 Conclusions

MMP/Geometer is a geometry software. It can be used to prove and discover theorems in Euclidean and differential geometries. It can also be used to produce proofs with geometric meanings and proofs in traditional style. We are currently building a webpage which will include MMP/Geometer and a large set of geometry theorems including those mentioned in Table 2.

MMP/Geometer can also be used as a geometry diagram editor. For a given input, MMP/Geometer will draw the diagram first using ruler and compass construction and then using general AGDG methods. It seems that large diagrams such as those in Figure 10 can be handled satisfactorily.

Altogether, we hope that MMP/Geometer may provide a useful tool for people to study, to learn and to use geometry.

Acknowledgment. We want to thank Prof. Wen-Tsün Wu for long time encouragements. The first author also wants to thank Prof. Shang-Ching Chou for insightful discussions. Many of the methods implemented in MMP/Geometer are developed under collaboration with Prof. Chou.

References

[1] P. Aubry, D. Lazard, and M. Moreno Maza, On the Theories of Triangular Sets, *J. of Symbolic Computation*, **28**, 45-124, 1999.

[2] S.C. Chou, *Mechanical Geometry Theorem Proving*, D.Reidel Publishing Company, Dordrecht, Netherlands, 1988.

[3] S.C. Chou and X.S. Gao, Ritt-Wu's Decomposition Algorithm and Geometry Theorem Proving, *CADE'10*, M.E. Stickel (Ed.) pp 207-220, LNCS, No. 449, Springer-Verlag, 1990.

[4] S.C. Chou and X.S. Gao, Mechanical Formula Derivation in Elementary Geometries, *Proc. ISSAC-90*, ACM, New York, 1990, 265-270.

[5] S.C. Chou and X.S. Gao, Proving Constructive Geometry Statements, *CADE11*, D. Kapur (eds), 20-34, Lect. Notes on Com Sci., No. 607, Springer-Verlag, 1992.

[6] S.C. Chou and X.S. Gao, Automated Reasoning in Differential Geometry and Mechanics: Part II. Mechanical Theorem Proving, *Journal of Automated Reasoning*, 10:173-189, 1993, Kluwer Academic Publishers.

[7] S.C. Chou and X.S. Gao, Automated Reasoning in Geometry, *Handbook of Automated Reasoning*, (eds. A. Robinson and A. Voronkov), 709-749, Elsevier, Amsterdam, 2001.

[8] S.C. Chou, X.S. Gao and J.Z. Zhang, *Machine Proofs in Geometry*, World Scientific, Singapore, 1994.

[9] S.C. Chou, X.S. Gao and J.Z. Zhang, A Deductive Database Approach To Automated Geometry Theorem Proving and Discovering, *J. Automated Reasoning*, 25, 219-246, 2000.

[10] P. J. Davis, The Rise, Fall, and Possible Transfiguration of Triangle Geometry: A Mini-history, *The American Mathematical Monthly*, **102**, 204–214, 1993.

[11] X.S. Gao, L. Huang and K, Jiang, A Hybrid Method for Solving Geometric Constraint Problems in *Automated Deduction in Geometry*, J. Richter-Gebert and D. Wang (eds), 16-25, Springer-Verlag, Berlin, 2001.

[12] X.S. Gao, D.K. Wang, D. Lin, Z. Qiu and H. Yang, MMP: A Mathematics-Mechanization Platform - a progress report, Preprints, MMRC, Academ,ia Sinica, April, 2002. http://www.mmrc.iss.ac.cn/˜mmsoft/.

[13] X.S. Gao, J.Z. Zhang and S.C. Chou, *Geometry Expert* (in Chinese), Nine Chapter Pub., Taipai, Taiwan, 1998.

[14] X.-S. Gao, C.M. Hoffmann, W. Yang, Solving Basic Gometrci Constraint Configurations with Locus Intersection, *Proc. ACM SM02*, 95-104, Saarbruecken Germany, ACM Press, New York, 2002.

[15] X.-S. Gao, C. Zhu, S.-C. Chou, and J.-X. Ge, Automated Generation of Kempe Linkages for Algebraic Curves and Surfaces, *Mechanism and Machine Theory*, **36**(9), 1019-1033, 2002.

[16] J. Ge, S.C. Chou and X.S. Gao, Geometric Constraint Satisfaction Using Optimization Methods, *Computer Aided Design*, **31**(14), 867-879, 2000.

[17] H. Gelernter, Realization of a Geometry-theorem Proving Machine, *Comput. and Thought*, (E.A. Feigenbaum, J. Feldman, eds.), 134–152, Mcgraw Hill, New York, 1963.

[18] C. Hoffmann, Geometric Constraint Solving in R^2 and R^3, in "Computing in Euclidean Geometry", eds D. Z. Du and F. Huang, pp. 266–298, World Scientific, 1995.

[19] H. Hong, D. Wang and F. Winkler, Short Description of Existing Provers, *Ann. Math. Artif. Intell.*, 13: 195-202, 1995.

[20] N. Jakiw, *Geometer's Sketchpad, User Guide and Reference Manual*, Key Curriculum Press, Berkeley, 1994.

[21] J.M. Laborde, *GABRI Geometry II*, Texas Instruments, Dallas, 1994.

[22] R.S. Latheam and A.E. Middleditch, Connectivity Analysis: a Tool for Processing Geometric Constraints, *Computer Aided Design*, **28**(11), 917-928, Elsevier Science Ltd., 1994.

[23] D. Lazard, A New method for Solving Algebraic Systems of Positive Dimension, *Discrete Appl. Math.*, **33**, 147-160, 1991.

[24] M. Kalkbrener, A Generalized Euclidean Algorithm for Computing Triangular Representations of Algebraic Varieties, *J. Symb. Comput.*, **15**, 143–167, 1993.

[25] D. Kapur and H. K. Wan, Refutational Proofs of Geometry Theorems via Characteristic Set Computation, *Proc. of ISSAC'90*, 277-284, ACM Press, New York, 1990.

[26] J. Richter-Gebert and U.H. Kortenkamp, *The Interactive Geometry Software Cinderalla*, Springer, Berlin Heidelberg, 1999.

[27] J. Owen, Algebraic Solution for Geometry from Dimensional Constraints, in *ACM Symp., Found of Solid Modeling*, ACM Press, Austin TX, 1991, pp. 397-407.

[28] D. Wang, *Elimination Methods*, Springer, Berlin, 2000.

[29] D. Wang, Geother: A Geometry Theorem Prover, In: *Proc. CADE-13*, LNAI 1104, 166-170, Springer, Berlin, 1996.

[30] D.K. Wang, Polynomial Equations Solving and Mechanical Geometric Theorem Proving. Ph.D Thesis, Inst. of Sys. Sci., Academia Sinica, 1993.

[31] W.T. Wu, *Basic Principles of Mechanical Theorem Proving in Geometries*, Volume I: Part of Elementary Geometries, Science Press, Beijing (in Chinese), 1984. English Version, Springer-Verlag, Berlin, 1994.

[32] W.T. Wu, A Mechanization Method of Geometry and its Applications I. Distances, Areas, and Volumes. *J. Sys. Sci. and Math. Scis.*, **6**, 204–216, 1986.

[33] W.T. Wu, A Constructive Theory of Differential Algebraic Geometry, In: LNM **1255**, Springer, Berlin Heidelberg, 173–189, 1987.

[34] W.T. Wu, *Mathematics Mechanization*, Science Press/Kluwer, Beijing, 2000.

[35] L. Yang, J. Zhang and X.R. Hou, *Nonlinear Algebraic Equation System and Automated Theorem Proving*, Shanghai Sci. and Tech. Education Publ. House, Shanghai (1996) [in Chinese].