Visual Reasoning in Geometry Theorem Proving

Michelle Y. Kim
IBM Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, New York 10598

Abstract

We study the role of visual reasoning as a computationally feasible heuristic tool in geometry problem solving. We use an algebraic notation to represent geometric objects and to manipulate them. We show that this representation captures powerful heuristics for proving geometry theorems, and that it allows a systematic manipulation of geometric features in a manner similar to what may occur in human visual reasoning

1 Introduction

The question of visual imagery in humans has been a controversial subject within cognitive psychology [Anderson, 1979, Kosslyn, 1980, Shepard and Cooper, 1982, Pylyshyn, 197.1]. No one doubts the conscious phenomena of imagery, or the act of visualization. What is problematic is the ultimate nature of images as mental representation [Johnson-Laird, 1983]. Is there a single underlying form of mental representation and are images only epiphcnominal; or are images a distinct sort of mental representation? Regardless of the outcome of this debate, that visual imagery as a natural means of dealing with spatial problems will remain irrefutable.

Visual imagery, or visual reasoning, as a useful tool for scientists and mathematicians has been demonstrated in the past [McKim, 1980, Kosslyn, 1983, Simon, 1987]. Our goal is to explore visual reasoning as a computationally feasible tool in problem solving. We investigate its role in the classical AI domain, discovering proofs for theorems in cuclidean plane geometry. To discover a proof requires ingenuity, imagination, and insights to a problem. Considering a model of the problem generally provides most valuable insights to a problem. In our domain, the model is a diagram, and through its manipulation the problem starts unfolding. Heuristic values of a model, or a diagram, are that it provides a counter-example of an unprovable theorem and more importantly that it serves as a vehicle for 'perceptual reasoning," perceptual in the sense that many facts are self-evident from the diagram and that they need not be established from fundamental axioms.

Building theorem-proving systems for geometry has been attempted frequently in the past [Gelernter, 1958, Gelernter, 1963, Goldstein, 1973, Nevins, 1974, Anderson, 1981, Anderson, 1983, Anderson, 1985, Coclho and Pcrcira, 1986, Lakin and Simon, 1987]. Sec [Coelho and Pcrcira, 1986] for a comparative study of previous work. Most notable system among them is the Geometry Theorem Prover [Gelernter, 1958, Gelernter, 1963], which was one of the earliest automated theorem provers and was distinguished by its reliance on a diagram to guide the proof The prover used the diagram as the pruning heuristic, e.g., it rejected as false any goal that was not true in the diagram. Its use of diagrams, however, was limited in that diagrams supplied only yes or no answers to questions of the form: Is segment AB equal to Segment CD in the figure?' Note that finite precision arithmetic, applied to the diagram, occasionally caused a provable sub-goal to be pruned erroneously. Furthermore, constructions such as adding lines to the diagram were done only as the last resource, and the help lines were drawn by randomly connecting any unconnected points, when all else failed.

Our aim is to further explore the heuristic values of the diagram, and show a method that allows the diagrams to be *perrcived,* or seen, and to be manipulated in a creative manner, similar to what may occur in human visual reasoning. To represent geometric features, we use an algebraic notation and capture what may be the key computational efficiencies that occur in human visual reasoning.

In the next section visual reasoning in plane geometry is discussed. In Section 3, a representation scheme by which geometric features are described is given. In Section 4, some useful patterns that are found in many geometry problems are identified and the methods by which they may be recognized are discussed. In Section 5, we describe Machine's I, an early version of machine implementation, which may be used as a front-end heuristic device to a more general geometry theorem prover. Finally conclusions appear in Section 6.

2 Visual Reasoning in Geometry

Visual reasoning in geometry may be considered as a two-step process: patterning, and analysis. Patterning, or pattern-seeking, as an active nature of visual perception has been formulated as a theory, known as Gestalt theory, by psychologists [McKirn, 1980]. The pattern that we perceive in a problem strongly influences the manner by which we approach the problem. So powerful is the perceptual tendency to perceive meaningful patterns, we will fill in missing parts. This effect is known as *closure.* Seeking meaningful patterns or generating them, or closing-in, is a particularly effective aid in geometry theorem proving. Once a meaningful pattern is found, its implications arc analyzed, or inferred.

Consider the problem in Figure 1: "Given a square *{ABCD),* take the midpoints of the four sides, and prove that the two triangles *(EEH)* and *(GFH)* are congruent to each other."

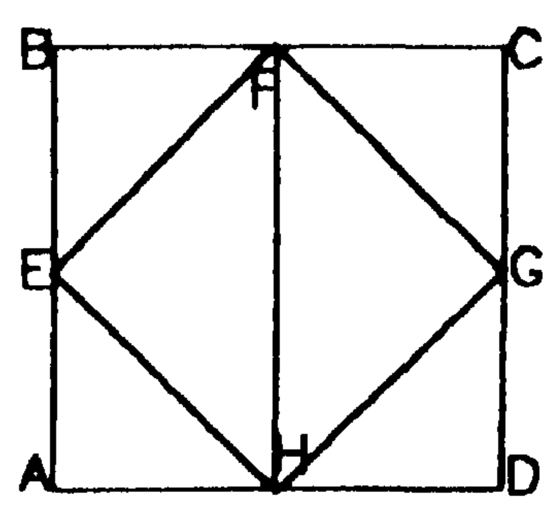

Figure 1. Reflection

To solve this problem, backward-chaining methods used by most of previous geometry-theorem proving systems [Gelernter, 1963, Goldstein, 1973, Coclho and Pereira, 1986] would first set up a goal to prove that the two triangles arc congruent, then set up sub-goals to prove that their corresponding sides are congruent, and then set up more sub-goals, repeatedly, to show that each pair of corresponding sides arc congruent, and so on. A human mathematician, given the problem, may perceive an apparent symmetry in the diagram by observing a reflection across *(FH)* or across *(EG).* As a symmetry is observed, it can be shown with little effort that the two triangles are congruent, and thus repeated proofs can be avoided.

Consider another problem as shown in Figure 2: 'Prove that the midpoint of the hypotenuse of a right triangle is equidistant from the three vertices." To solve this problem, suppose the right triangle *(ABC*)* is half-turned, or turned by 180 degrees, about the midpoint. This half-turn generates (DCB), which is a copy of *(ABC)* 180 degree turned about *(M).* The two triangles form a rectangle *(ABCD).* Once a rectangle is seen and its diagonals are computed, it can be trivially inferred that the two diagonals are congruent and that they bisect one another, and thus, the four line segments are congruent, and so on.

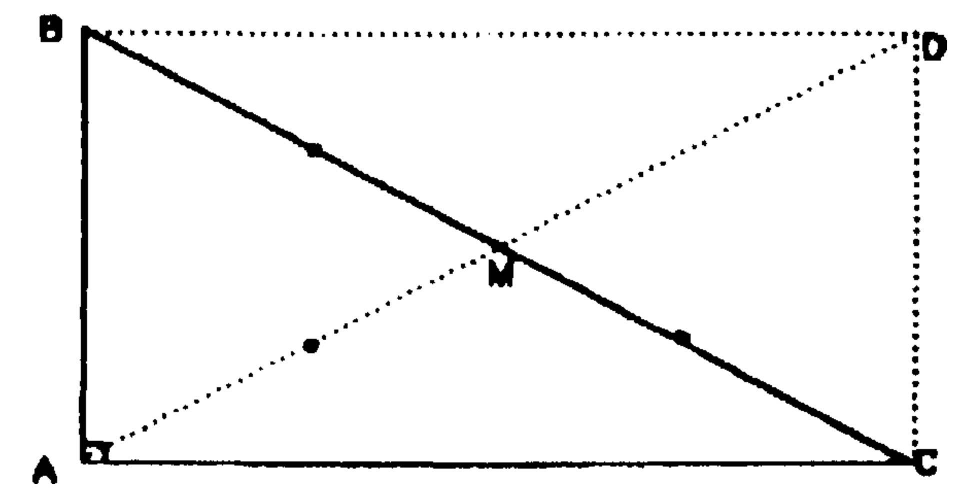

Figure 2. Half-turned right triangle

Observe that we have obtained an outline, or a plan, for a proof by finding a line symmetry in Figure I. We have obtained useful information for a proof in Figure 2 by turning the right triangle and transforming it to a rectangle. We show in the next section a notation that can capture important heuristics for proving geometry theorems. The notation can suggest that there is a reflection in one problem, and that a half-turn maybe useful in another.

3 Representing Geometric Features

A good choice of representation can greatly facilitate the recognition task. Suppose we represent the square in Figure I, using the primitives *a* — } and *b* = —>, by the string *(a@h)* • *(h(±)a),* where © is to join a pair of primitives, and • to describe a closed object (Definitions follow below). This string is a palindrome. As we shall see later, a palindrome strongly suggests that there is some type of symmetry. It is this capacity that we are alter. The representation is simple. It is also syntactic, and thus geometric features can be manipulated easily and systematically. More importantly, this method to pattern seeking provides useful semantic information despite its syntactic appearance.

3.1 Shape Primitives and their Operations

We define shape primitives as directional pairs *(p.l, p.a),* where *p* is the name of the primitive, *p.l is* the length, *p.a* is the angle. The angle increases in the counter-clock wise direction, 0 < *p.a* < 180, thus allowing a unique representation of a primitive.

Shape primitives are connected to form and characterize a structural pattern. We first define simple *joining* operations as shown in Figure 3. We show three ways of joining a pair of primitives, such that each of the primitives has two distinct connection points, a head and a tail. These three binary operations, denoted by ©. 0 , ®, allow a primitive to be attached to the other primitive only at its head or tail. 0 and (g) operations are commutative, while © is not. Similar techniques have been used in the past in the pattern recognition research in computer vision [Shaw, 1972].

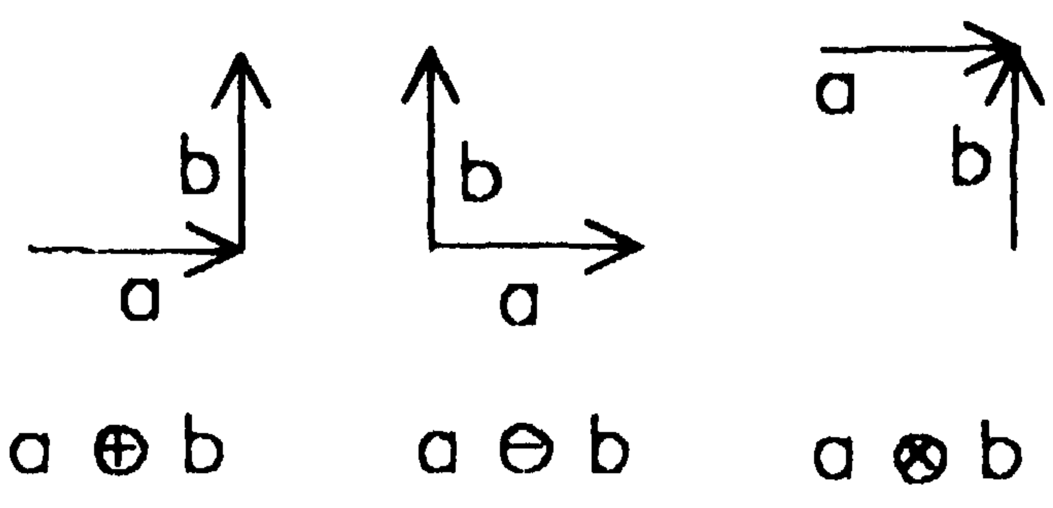

Figure 3. Rules for joining primitives

Operation * creates a primitive out of two joined primitives. In Figure 4, $*(a \oplus b)$ constructs primitive c, $*(a \ominus b)$ constructs d, and $*(a \otimes b)$ constructs d'. As a short hand notation, where there is no confusion, we will write $a*b$ to mean any of these three constructions.

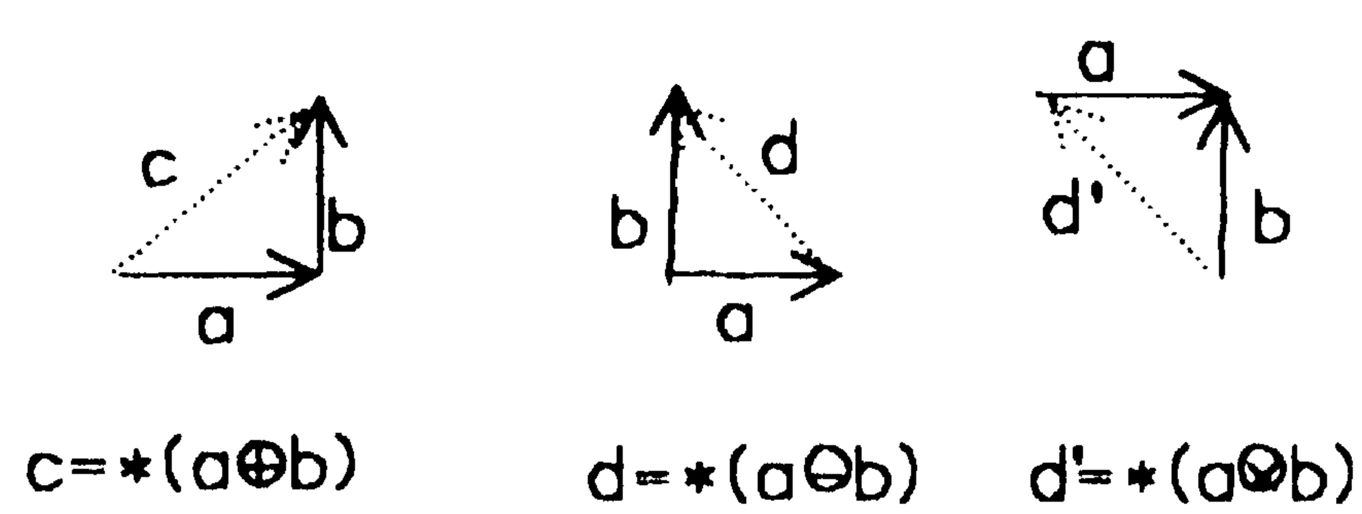

Figure 4. Primitive construction operator *

Operation • is to describe a closed structure. For example the three triangles in Figure 4 can be described by $c \bullet (a \oplus b)$, $b \bullet (a \oplus d)$, and $(d' \oplus a) \bullet b$, from left to right. We let operation • be commutative. Although we have defined three join operations above, we may only use operation ⊕ and operation • to describe a triangle. This is to guarantee a unique representation from which other equivalent representations can be computed.

The ***structural joining*** operation, denoted by $JN(a)$, joins a pair of closed structures over primitive a. $JN(a)$ is a useful operator in describing a non-convex structure. Thus, the non-convex quadrilateral in Figure 5 may be expressed as $((c \oplus e) \bullet d)\ JN(c)\ ((c \oplus f) \bullet g)$, or simply $(ced)\ JN(c)\ (cfg)$, where $(p_1 \ldots p_n)$ denotes an n-sided polygon and p_i s are its primitives. Operation JN is also commutative.

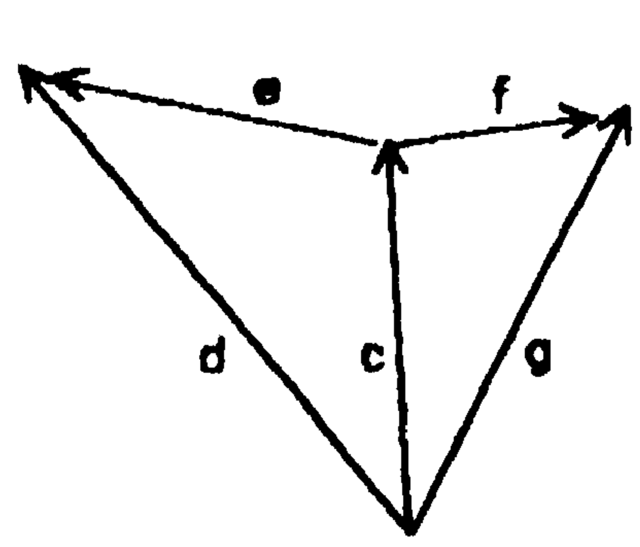

Figure 5. Structural joining

Most of geometric objects can now be represented using our shape primitives and their operations shown above. Note that the starting point may be added in the definition of a primitive, or a rule may be defined that connects two disjoint primitives, etc., thus making the method more general. This extension can be made in a straightforward manner, and in this paper we have adopted a simplifying notation for a simpler discussion. In the next section we discuss visual reasoning, or pattern seeking, using our representation.

4 Patterning and Reasoning

Geometric objects provide interesting abstractions of many patterns we find in nature, art, and industry. Symmetry and dilations, or scaling, are among them. Finding a line of symmetry or a point of symmetry provides important clues in the search of a proof. Where there is no apparent symmetry, it almost always pays to create one. As symmetry is to congruence, dilations is to similarity. A good many geometry problems like so many objects around us contain dilations. Identifying a dilation when it is present or by filling in missing parts when it is not apparent is as effective as finding symmetry.

Observe that it is necessary to recognize or generate meaningful patterns in a systematic manner. We may interpret each shape primitive as a symbol permissible in some grammar, then, the syntactic pattern recognition process is a straightforward task. In this report, we do not establish a formal grammar, but provide an informal description of the recognition process and show simple examples. For definitions of the patterns that we discuss below, and for more examples of using such patterns to guide the search of a proof, see [Kim, 1988].

4.1 Finding a Point Symmetry

Consider the parallelogram $(a \oplus b) \bullet (b \oplus a)$ in Figure 6. Syntactically, its string representation is a palindrome. A palindrome:

$$\{w \bullet w^r : w \in T^*\},$$

$$T = \{\oplus, \ominus, \otimes, *, \bullet\} \cup \{p : p \text{ a primitive}\},$$

suggests that there is a point symmetry in the structure. A palindrome uniquely defines a structure with a point symmetry and that a structure with a point symmetry may be uniquely defined as a palindrome.

As an example, we show below a series of inferencing processes that may occur given a palindrome $(a \oplus b) \bullet (b \oplus a)$, where a is not perpendicular to b: (1) the structure is a parallelogram, (2) its diagonals c and d are computed, c is constructed from $*(a \ominus b)$ and d from $*(a \oplus b)$, (3) the diagonals of a parallelogram bisect one another, (4) but $c.l \neq d.l$, hence the structure is not a rectangle, (5) and thus, our symmetry is only a point symmetry not a line symmetry, (6) the point of symmetry is the intersection of the two diagonals c and d, and so on.

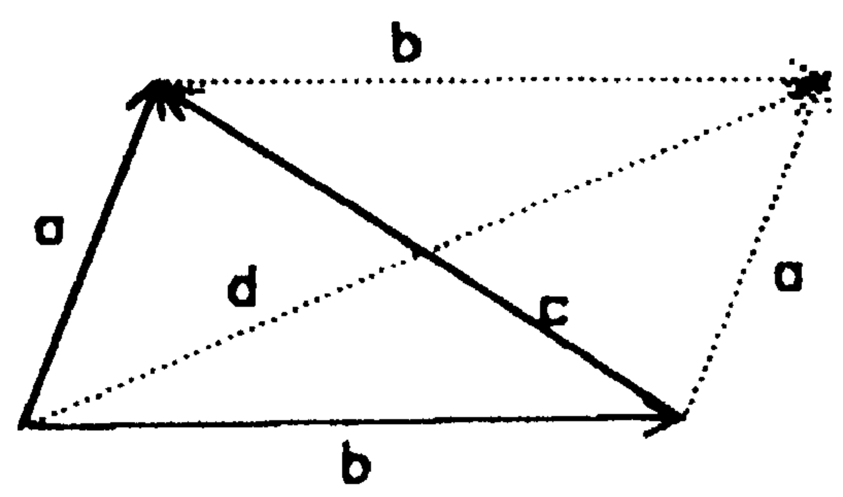

Figure 6. Half-turn

4.2 Finding a Line Symmetry (Reflection)

Informally, a reflection is a mirror image provided by an axis of symmetry. Without loosing generality, a reflection may be thought of as having two congruent structures joined over the axis.

Let w be a string representation that describes an isosceles triangle, $a \bullet (b \oplus c)$, where two of the three primitives are of equal length. Then w contains a reflection and its axis can be computed in a straightforward manner.

Consider the quadrilateral in Figure 7 $(a \oplus b) \bullet (a' \oplus b')$, $a'.l = a.l$, $b'.l = b.l$. Let c' be $a^{+}a'$. Triangle $a \oplus c' \bullet a'$ is an isosceles triangle and its axis is $d = a^{+}c$, where c is defined by: $c.a = c'.a$, $c.l = c'.l/2$, $c' = (c \oplus c)$. Then (acd) is a reflection of $(a'cd)$ over d. Similarly, $c' \oplus b' \bullet b$ is an isosceles triangle and its axis e can be computed in a straightforward manner. The quadrilateral contains a reflection in line $(d \oplus e)$, over which $(abde)$ and $(a'b'de)$ are joined.

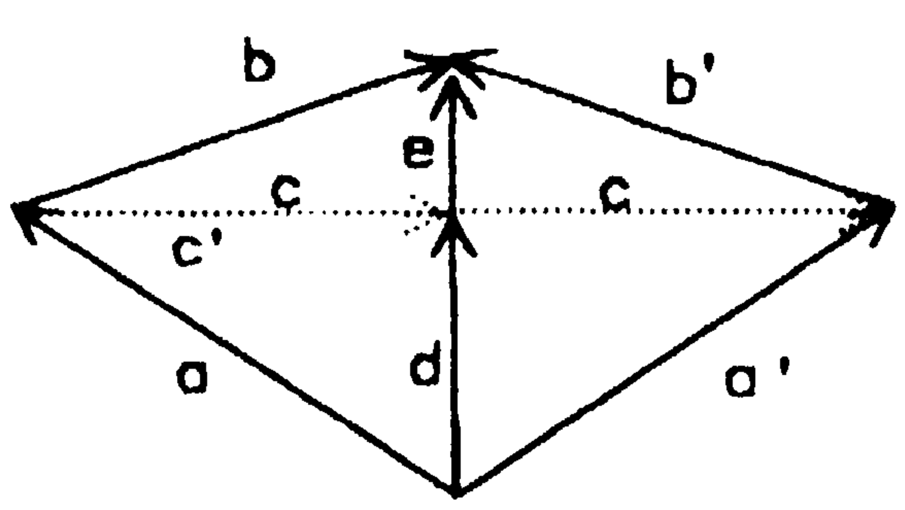

Figure 7. Reflection

4.3 Finding a Dilation (Scaling)

In Figure 8, the triangle $(a \oplus a) \bullet (c \oplus (b \oplus b))$ contains a scaled triangle:

$a \bullet (c' \oplus b)$, where $c'.l = c.l/2, c'.a = c.a$.

A string representation which is of the form $w \bullet w'$, where w or w' contains $(... \oplus a \oplus a \oplus ... \oplus b \oplus b)$, or w is of the form $(... \oplus a \oplus a)$ and w' of the form $(... \oplus b \oplus b)$, where a and b are two distinct primitives, suggest that there is a dilation in the structure and that the point of dilation is where the two primitives meet.

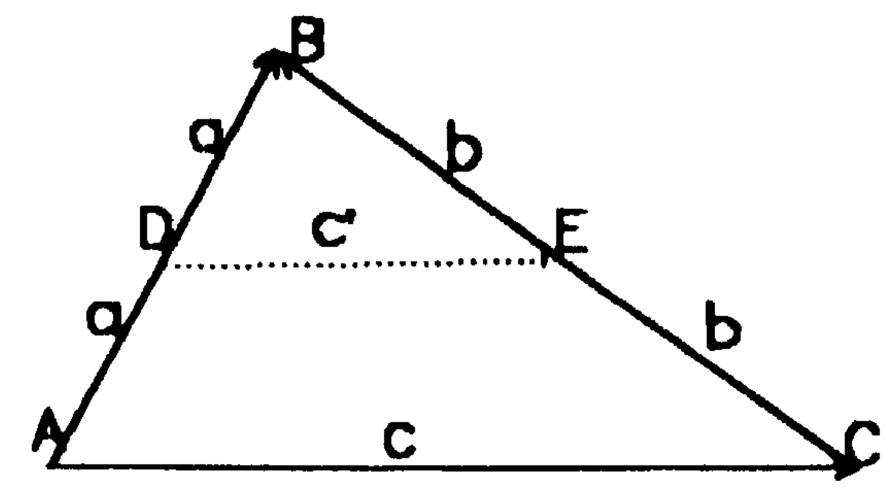

Figure 8. Dilation

Having discussed the principles of finding useful patterns using our representation, we show an example below, and provide a summary of what may occur in solving the problem. For more examples, sec [Kim, 1988].

4.4 An Example

The power of imagery becomes obvious with more involved problems. Consider the perpendicular bisector concurrence theorem depicted in Figure 9.

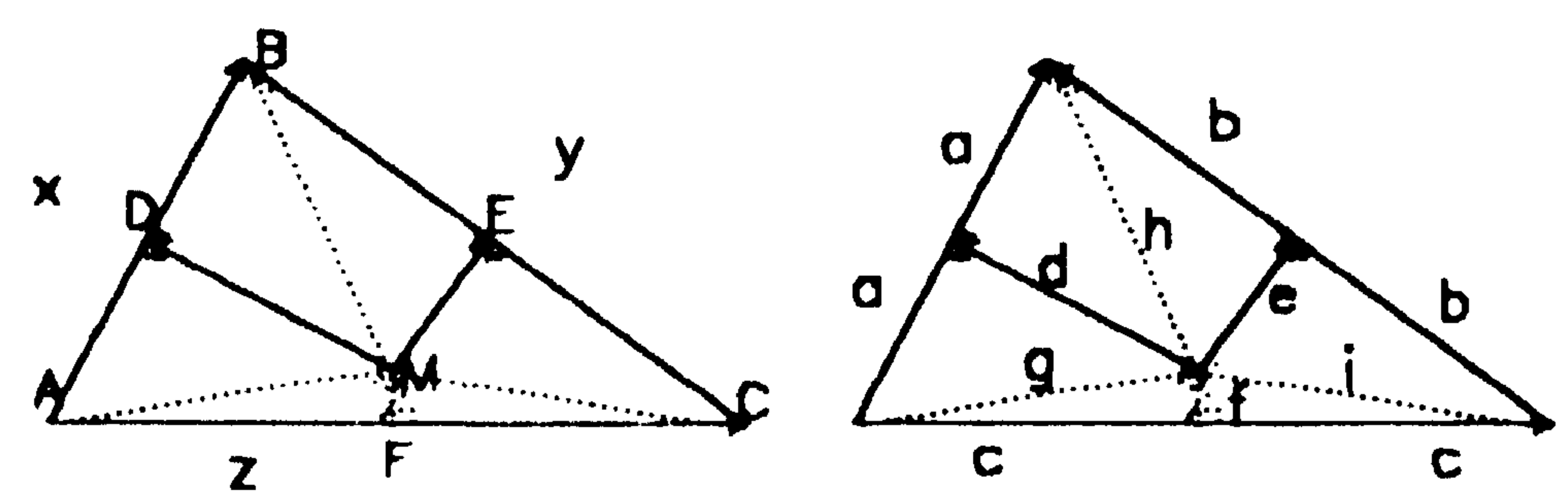

Figure 9. Proving the perpendicular bisector concurrence theorem

The theorem states that the perpendicular bisectors of the sides of a triangle are concurrent and that their point of concurrency is equidistant from the vertices of the triangle. We let (AB) denote a line segment connecting two points (A) and (B). We assume that the problem is given using this notation. In the figure (DM) and (EM) are two bisectors with the point of concurrency (M). Having shown that the three line segments (AM), (BM), and (CM) are congruent by showing two reflections, the proof is complete if (FM) is shown to be the third bisector. We let (ABC) be $x \bullet (z \oplus y)$, describing (AB), (BC), and (AC) by x , y, and z, respectively, and let $a.l = x.l/2$ and $b.l = y.l/2$. We, then construct d and e such that d is the perpendicular bisector of $(a \oplus a)$ and e is the perpendicular bisector of $(b \oplus b)$. As the two intersecting perpendicular bisectors have been defined it follows that:

(gda) $RL(d)$ (dah),

$g = {}^{+}(a \otimes d)$,

$h = {}^{+}(d \oplus a)$,

where $RL(d)$ denotes a reflection over d. Thus $g.l = h.l$. Similarly, $h.l = i.l$. Thus $g.l = h.l = i.l$. Following our definition of reflection, $(z \oplus i) \bullet g$, $i.l = g.l$, contains a reflection in line f such that $f = *(c \ominus g)$, and it follows that f is the perpendicular bisector of (AC).

5 An Implementation: Machine ' s I

"Machine's I (for eye)" is a rule-based program that has been implemented in ECEPS [IBM Enhanced Common Lisp Production System, IBM, 1988], as a front-end heuristic device to a geometry theorem prover. A user presents a problem to Machine's I by declaring the premises and the goal. The program then builds a model, or a diagram, of the problem, draws it, and describes it in terms of shape primitives and the operations. Shape primitives and structures have been represented by working memory elements, and their manipulations have been implemented as ECEPS rules.

The program first starts to pattern: Obvious patterns arc detected, or a meaningful pattern is created. Note that in patterning, not only the premises but the goal may provide a useful clue. As patterning progresses, new facts arc inferred. In fact, this patterning phase may be considered as a mixture of backward and forward chaining- Backward in the sense that the goal to prove may strongly influences the patterning, and forward by the way reasoning proceeds from the premises. Having patterned, the results may be passed to a geometry theorem prover, so that a proof can be completed. It has been observed that for simple theorems the proof was often immediate after the patterning phase.

Much of the proof procedures addressed in this paper can be efficiently implemented in ECEPS due to its power. Unlike most resolution-based mechanical theorem proving systems in Prolog that lack operational semantics [Coelho and Pereira, 1986], ECU'S provides powerful demand-driven pattern matching capabilities [Schor et al., 1986], which allow a dynamic pattern matching. More, it provides a flexible control strategy by prioritizing rule firings.

6 Conclusions

We have shown that a simple syntactic method provided powerful heuristic information for proving geometry theorems. The representation is simple, easy to manipulate, and yet it captures what may be the key computational efficiencies that occur in human visual reasoning. We do not have a good characterization of what is involved in human visual reasoning. Nonetheless, the implications of capturing visual heuristics in a simple notation are great and need to be pursued further.

A ckno wledgemen ts

The author is indebted to D. Sabbah and S. Addanki for their many valuable suggestions, H. Lee for his help in ECLPS, C. Albcrga and D. Ferguson for their helpful comments on the draft. Special thanks are also due to S. Kosslyn for a stimulating initial discussion, and the contents of the manuscript has been greatly improved by the comments of A. Newell.

References

[Anderson, 1979] Anderson, J. R, "Arguments concerning representations for mental imagery," Psychological Review, 85, 249-277.

[Anderson, 1981] Anderson, J. R, 'Tuning of Search of the Problem Space for Geometry Proofs/' Proc. 7th I.ICAI, 1981.

[Anderson, 1983] Anderson, .1. R, "Acquisition of Proof Skills in Geometry," Machine learning: An Artificial Intelligence Approach, Michalski, Carbonell, Mitchell (Eds), Morgan Kaufmann Publishers, 1983, 190-219.

[Anderson, 1985] Anderson, J. R, 'The Geometry Tutor," Proc. 9th IJCAI, 1985.

[Coelho and Pcreira, 1986] Coelho, II., Pereira, E. M., "Automated Reasoning in Geometry Theorem Proving with Prolog," Journal of Automated Reasoning, Vol. 2, No. 4, 1986, 329-389.

[IBM, 1988], IBM, "Enhanced Common Eisp Production System User's Guid and Reference," SC38-7016, 1988.

[Gclcmtcr, 1958] Gclerntcr, II., Rochester, N., "Intelligent Behavior in Problem-Solving Machines," IBM Journal of Research and Development, Vol. 2, No. 4, 1958.

[Gclerntcr, 1963] Gclerntcr, II., "Realization of a Geometry-Theorem Proving Machine," Computers and Thought, Ecigenbaum, Ecldman (Eds), Mc-Graw-IlilI, 1963, 134-152.

[Goldstein, 1973] Goldstein, I., "Elementary Geometry Theorem Proving," MIT, AI laboratory Memo No. 280, 1973.

[Johnson-Ixiird, 1983] Johnson-laird, P. N., Mental Models," Harvard University Press, 1983.

[Kim, 1988] Kim, M. Y, The Role of Visual Imagery in Geometry Theorem Proving," IBM Computer Science Research Report, RC13628, March 1988.

[Kosslyn, 1980] Kosslyn, S. M., "Image and Mind," Harvard University Press, 1980.

[Kosslyn, 1983] Kosslyn, S. M., "Ghosts in the Mind's Machine/ W. W. Norton & Company, 1983.

[McKim, 1980] McKim, R. II., 'Thinking Visually," Eifetime learning Publications, 1980.

[Ncvins, 1974] Ncvins, A. J., "Plane Geometry Theorem Proving Using Forward Chaining/' MIT, AI Laboratory Memo No. 303, 1974.

[I,akin and Simon, 1987] I akin, J. II., Simon, II. A., "Why a Diagram is (Sometimes) Worth Ten Thousand Words/' Cognitive Science, 1987.

[Schor et al., 1986] Schor, M., Daly, T., Fee, U.S., Iihbits, B. R., 'Advance in Retc Pattern Matching," Proc. AAAI, 1986.

[Shepard and Cooper, 1982] Shcpard, R. N., Cooper, I,. A., ''Mental Images and Their Transformations," Mir Press, 1982.

[Simon, 1987] Simon, II. A., "Images of Empty Space: Einstein's Word Pictures," Carnegie-Melon University, 1987.

[Shaw, 1972] Shaw, A. C, "Picture Graphs, Grammars, and Parsing/' Frontiers of Pattern Recognition, Watanabe (Hd.), Academic Press, 1972.

[Pylyshyn, 1973] Pylyshyn, Z. W, "What the mind's eye tells the mind's brain: A critique of mental imagery," Psychological bulletin, 1980, 1-24.