Wiley 1977, pp. 267-311.

[25] H. Abelson and A. di Sessa, *Turtle Geometry*, MIT Press, 1980.

[26] L. Guibas, L. Ramshaw, and J. Stolfi, "A kinetic framework for computational geometry," tech. rept., Xerox Park and Stanford University, May, 1983.

[27] E. Mach, *Space and Geometry*, The Open Court Publishing Co., Chicago, 1906.

[29] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.

[30] G. T. Toussaint, Ed., *Computational Morphology*, North Holland, 1988.

[31] R.E. Mueller, *Iventivity*, E.M. Hale & Co., 1966.

[32] J.L. Adams, *Conceptual Blockbusting*, San Francisco Book Co. Inc., 1976.

[33] A. Koestler, *The Act of Creation*, Picador, 1975.

[34] R. Sommer, *The Mind's Eye*, Dell Publishing Co., New York, 1978.

[35] M. Samuels and N. Samuels, *Seeing with the Mind's Eye*, Random House, 1975.

[36] D.A. Dondis, *A Primer of Visual Literac*y, M.I.T. Press, Cambridge, 1973.

versity Press, 1945.

[8] I. Lakatos, *Proofs and Refutations: The Logic of Mathematical Discovery*, Cambridge University Press, 1976.

[9] J. Pascual-Leone, "The forms of knowing in the psychological organism: reflections on Royce and Roseboom (eds.) Psychology of Knowing," *Phil. Soc. Sci*., Vol. 6, 1976, pp. 175-181.

[10] H. Gardner, *Frames of Mind: The theory of Multiple Intelligences*, Basic Books Inc., 1983.

[11] P.N. Johnson-Laird and P.C. Wason, Eds., *Thinking*, Cambridge University Press, 1977.

[12] R. Arnheim, *Visual Thinkin*g, University of California Press, 1969.

[13] L. Pelland, "Kinesthetic stimulation as a method for improved drawing skill acquisition," Masters thesis, Concordia University, Montreal, 1980.

[14] L. Pelland, "La danse et le dessin," *Re-flex*, Vol. 3, No. 3, 1983, pp. 52-53.

[15] S. Papert, Mind-Storms, Basic Books Inc., 1980.

[16] B. Rosenberg and D. Landgridge, "A computational view of perception," *Perception*, Vol. 2, 1973, pp. 415-424.

[17] R.A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Infor. Proc. Letters*, Vol. 2, 1973, pp. 18-21.

18] W.F. Eddy, "A new convex hull algorithm for planar sets", *ACM Trans. Math. Software*, Vol. 3, No. 4, 1977, pp. 398-403 and 411-412.

[19] S.G. Akl and G.T. Toussaint, "Efficient convex hull algorithms for pattern recognition applications" *Proc. 4th Int. Joint Conf. on Pattern Recognition*, Kyoto, Japan, 1978, pp. 483-487.

[20] L. Devroye and G.T. Toussaint, "A note on linear expected time algorithms for finding convex hulls", *Computing*, Vol. 26, 1981, pp. 361-366.

[21] B.K. Bhattacharya and G.T. Toussaint, "A counter example to a diameter algorithm for convex polygons," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-4, May 1982? pp. 306-309.

[22] G.T. Toussaint, "Solving geometric problems with the rotating calipers," Proceedings MELECON'83, Athens, Greece, June 1983.

[23] A. Einstein, "Geometry and experience," Prussian Academy of Sciences, Berlin, January 27, 1921; in *Sidelights on Relativity*, Dover, 1983.

[24] W.B. Weimer, "A conceptual framework for cognitive psychology: motor theories of the mind," in *Perceiving, Acting and Knowing*, Eds., R. Shaw and J. Bransford, John

"As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality."

We see then that computational geometry with kinesthetic heuristics offers a rich and more relevant manner of teaching a “living” mathematics. It offers the student a “truer” picture of reality. It should also be a good antidote for mathophobia. Readers familiar with Turtle Geometry [25] may have noticed its similarity to the computational geometry with kinesthetic heuristics discussed in this paper. In fact Turtle Geometry is a *computational* geometry and the moving turtle is clearly a kinesthetic operator. Some of the algorithms in computational geometry such as those using the rotating calipers can be easily placed in the turtle framework [25]. The main difference seems to be that the computational geometry presented here is more advanced than turtle geometry. While at present this material forms the content of graduate courses at many of the world’s leading universities, this is not because the material is that difficult, but rather because it lies at the frontiers of knowledge. I believe much of the material in [1] - [3] is eminently suited to high school students with personal computers at their disposal. Thus this material would appear to be a natural continuation of the turtle geometry program envisioned by Papert [15] for children.

We have seen also in this paper that there are "good" and "bad" kinesthetic heuristics in the sense that some yield correct algorithms whereas others yield incorrect ones. It would appear then that a good pedagogical strategy for the successful algorithm designer in computational geometry (particularly that aspect of it dealing with dynamic problems such as collision avoidance and spatial planning in robotics) is to train the kinesthetic sense. Turtle geometry certainly attempts to do this but mainly in the visual sense since students are "stuck" to the screen. We need to develop teaching tools where the student develops kinesthetic thinking through interaction with the "real" three dimensional world of motion. Perhaps dance [14] is but one of these techniques.

5. REFERENCES

[1} M.I. Shamos, "Computational Geometry", Ph.D. thesis, Yale University, 1978.

[2] F.P. Preparata, Ed., *Computational Geometry*, JAI Press, Inc., 1983.

[3] G.T. Toussaint, Ed., *Computational Geometry*, North Holland Publishing Co., 1984.

[4] G.T. Toussaint, "Pattern recognition and geometrical complexity," *Proc. 5th Int. Conf. on Pattern Recognition*, Miami Beach, 1980, pp. 1324-1347.

[5] G.T. Toussaint, “Computational geometric problems in pattern recognition”, *Pattern Recognition Theory and Applications*, J. Kittler, ed., NATO Advanced Study Institute, Oxford University, April 1981.

[6] D.T. Lee, "Computational geometry: a survey," *IEEE Transactions on Computers*, to appear.

[7] J. Hadamard, *The Psychology of Invention in the Mathematical Field*, Princeton Uni-

[22].

4. Conclusions

4.1 Design and analysis of geometric algorithms

In section I it was mentioned that a score of recently published algorithms in computational geometry have been discovered to fail recently [5]. An example of one such case was illustrated in section III with the diameter problem. One thing that all these algorithms have in common is that they are based on kinesthetic heuristics rather than logico-mathematical heuristics. A second thing that they all have in common is that they are much faster than their logico-mathematical counterparts, even when they do work correctly. This suggests that kinesthetic heuristics are better than logico-mathematical ones from the efficiency point of view and therefore suggests that the kinesthetic approach is a good pedagogical strategy for the design of geometric algorithms. On the other hand, the fact that many of these algorithms based on kinesthetic heuristics fail is somewhat disconcerting and suggests that we have here a "double edged sword".

I suggest that there are at least two reasons why many of these algorithms have failed. The first is that since a proof is a logico-mathematical exercise it "gets along well" with logico-mathematical heuristics. On the other hand there is an extra-logical gap between a logico-mathematical proof and a kinesthetic type algorithm. This gap must be bridged successfully for the algorithm to work correctly. This is often the first trap the algorithm designer must avoid. Of course with clear and rigorous logical thinking one should be able to construct this bridge without great difficulty. The other side of this sword however is the fact that often the kinesthetic heuristics are so convincing that either a proof seems unnecessary and is not given or attempted, or as is often the case an incorrect proof is outlined. I conjecture that when an incorrect proof is given it is because the convincing nature of the heuristic causes the proof to be merely a rationalization of something we are already convinced is correct. This is the second trap facing the algorithm designer. A good strategy is therefore to harness and use but also to mistrust the kinesthetic heuristics, no matter how convincing they may appear to be, and subject them to rigorous logico-mathematical analysis.

4.2 Implications for education

Traditionally, in the educational system, mathematics results in general, and proofs in particular, are presented as dry, purely logical, step by step procedures that lead in a "vertical" way to the "goal". This is of course only the finished product of the "living" mathematics which is a more passionate and semi-blind road to discovery [8]. In the words of Seymour Papert [15].

> "Mathematical work does not proceed along the narrow logical path of truth to truth to truth, but bravely or gropingly follows deviations through the surrounding marshland of propositions which are neither simply and wholly true nor simply and wholly false."

We have seen in this paper that there is a strong gap between kinesthetic heuristics for solving a problem and the finished algorithm with its proof of correctness. This is of course only an exaggerated form of a situation that exists in scientific discovery in general [8] because of the gap that exists between "reality" and mathematics. Albert Einstein 123] put it this way:

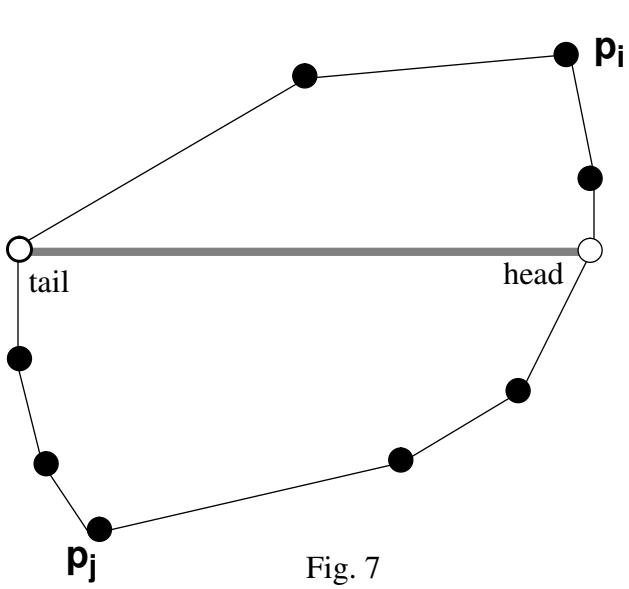

Fig. 7

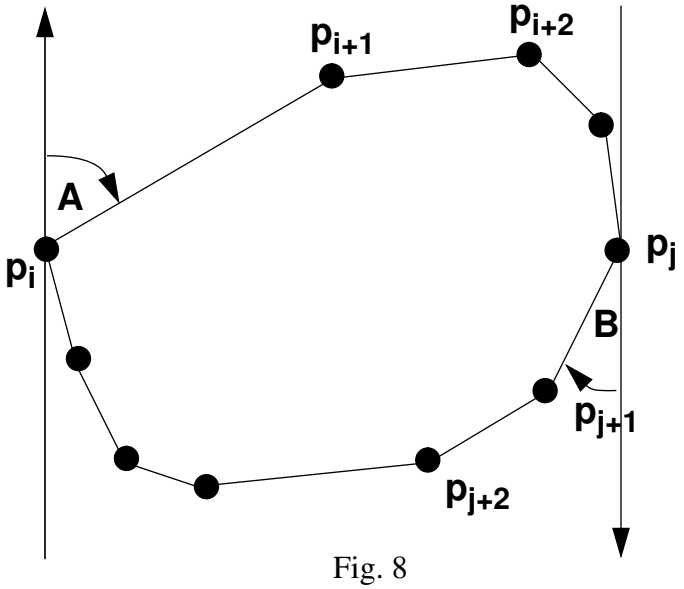

Fig. 8

gorithm [1].

Algorithm D-2

Begin

Step 1: Find the two points with minimum and maximum x coordinates. Call them p_i and p_j (see Fig. 8).

Step 2: Place the calipers vertically on p_i and p_j. (They will be rotated in a clockwise direction.)

Step 3: Measure the two angles A and B. Compute the distance between p_i and p_j as a candidate for the diameter D(P).

Step 4: Rotate the calipers by the smaller of the two angles (in Fig. 8 it is B). This gives a new pair of contact points (in Fig. 8 we now have p_i and p_{j+l}). Measure their distance.

Step 5: Repeat steps 3 and 4 until the calipers return to their original position. The largest distance encountered is the diameter D(P).

End

Algorithm D-2, like algorithm D-l, runs very fast and in time proportional to n. Unlike algorithm D-l, algorithm D-2 is always guaranteed to give the correct solution but the proof of correctness is longer and more difficult than that for the logico-mathematical algorithm.

We note here that the rotating caliper heuristic can be used to solve a variety of computational geometric problems besides the diameter problem and it is thus quite a powerful general tool

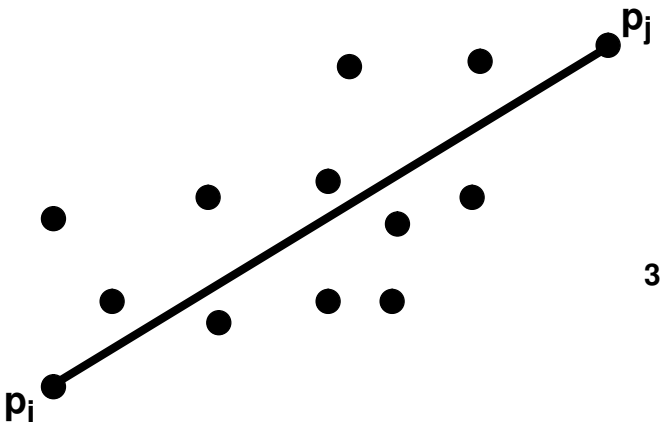

Fig. 5

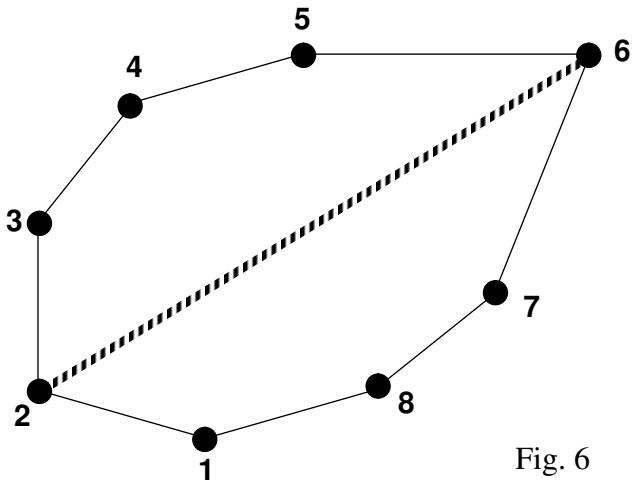

Fig. 6

er. (In Fig. 7 the head will stop at p_i)

Step 3: Now with the head fixed at p_i proceed to move the tail in the same way. (In Fig. 7 it will stop at p_j)

Step 4: Keep alternating the role of head and tail as in steps 2 and 3 until the elastic does not grow any longer. The final position of the elastic identifies the diameter D(P) and the diametrical pair of points.

End

This algorithm provides another example of a convincing kinesthetic heuristic which would appear not to require a proof. In fact no proof of correctness was published with this algorithm. While the algorithm is very fast indeed and runs in time proportional to n, it is in fact incorrect. A counter-example is given in [21] but the readers should discover one on their own. The reason for failure is similar to that for algorithm CH-3. As is often the case in hill-climbing procedures, algorithm D-l can get stuck in a local rather than global extremum.

The rotating caliper heuristic: This is a dynamic version of the sandwich heuristic. One way to measure the width of an object such as a hamburger patty is place it in between two slices of bread and measure the distance between the slices, i.e., the height of the sandwich not counting the thickness of the slices of bread. In a mechanics laboratory one does this with a set of calipers to ensure that "the two slices of bread remain parallel to each other." The rotating-calipers heuristic is based on the intuition that if we "rotate the two slices of bread" simultaneously keeping them parallel then the greatest distance encountered between points of contact should be the diameter. In our two-dimensional example this is illustrated in Figure 8. This idea leads to the following al-

of points for which the algorithm fails. This is left as an exercise for the reader.

The reason the algorithm can fail is that step 3 violates the notion of global exterior of the set by repeatedly searching for points with maximum x and y directions. Since these directions have already "been used up" in steps 1 and 2 they cannot be used again. Therefore this implementation is not the correct kinesthetic inverse of the shrinking procedure. We can remedy this by modifying step 3 to search for the points furthest in a direction perpendicular to the line through y_{max} and x_{max}. In this way we always search in new directions and we guarantee that points are global rather than local extrema.

This algorithm has been independently discovered by several researchers (see for example [18], [19]). As with algorithm CH-2 it is more difficult than CH-l to prove correct but it is extremely fast. In fact, it was proven in [20] that in many situations it executes in time proportional to n, the number of input points.

3. The Diameter Problem

The diameter of a set of points $P = \{p_l, p_2,..., p_n\}$ on the plane, denoted by D(P) is the distance between the two furthest points of P. The diameter problem is to identify these points (points p_i and p_j for example in Fig. 5) given a set P. We will consider the special problem of finding the diameter of a convex polygon, i.e., the points form the vertices of a convex polygon P which we are given as an ordered list, say, in the clockwise direction (see Fig. 6).

3.1 The Logico-Mathematical Approach

One obvious way to solve this problem is to consider all pairs of points, compute their distance, and select the pair that yields the maximum distance. The proof of correctness of this procedure is trivial but the algorithm is slow. The time taken by this algorithm is proportional to the number of pairs of points, that is, proportional to n^2.

3.2 Kinesthetic Heuristics

One published algorithm which almost every student of computational geometry rediscovers is based on the growing elastic heuristic. This approach is similar to the convex hull approach in algorithm CH-3 (the first step is identical in fact) except that we always keep the elastic around two points of the set only. We think of the ends of the stretched elastic as the *head* and *tail*.

Algorithm D-l

Begin

Step 1: Find the vertices of P with the minimum and maximum x coordinate and place the elastic band around these two points. Let x_{max} be the head, x_{min} the tail. (see Fig. 7)

Step 2: With the tail fixed proceed to move the head to the adjacent clockwise or counterclockwise point if the elastic gets longer. Continue doing this until it does not grow any long-

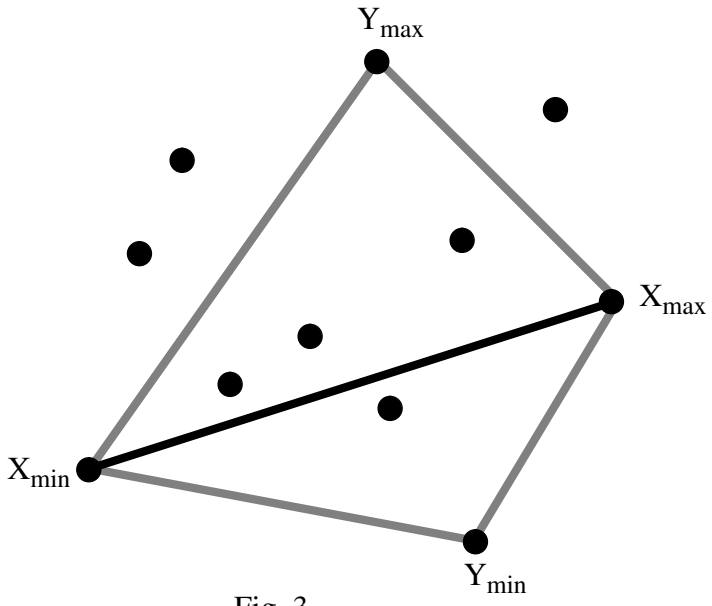

Fig. 3

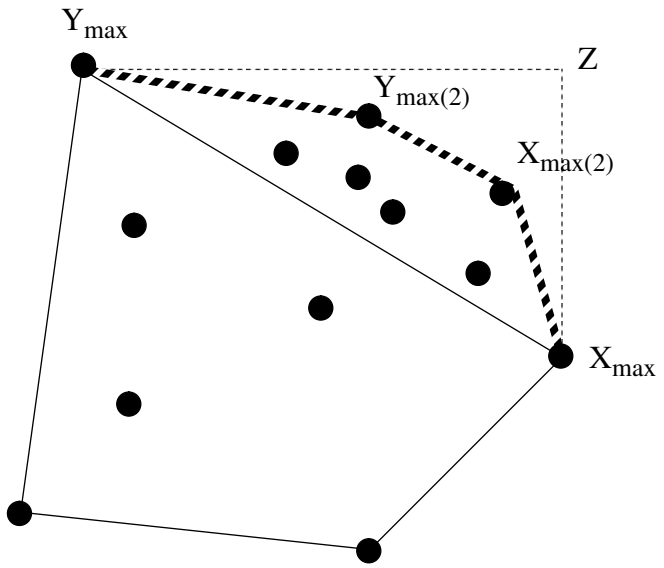

Fig. 4

elastic is a parallel operation and in algorithm design we usually look for a sequential procedure. We now describe an algorithm very frequently proposed by students upon first encountering this heuristic. The algorithm uses the kinesthetic heuristic of taking an un-stretched elastic band and step by step enlarging it until it covers all the points. Nobody has any difficulty doing this with a real elastic band on a board with pegs sticking out of it representing the points. However, their formal descriptions of what they do often contradict their actions. It seems here that "an action is worth a thousand pictures".

Algorithm CH-3

Begin

Step 1: Place the elastic band stretched between the two points with maximum and minimum x coordinates. (see Fig. 3)

Step 2: Take the upper section of the elastic and connect it onto the point with maximum y coordinate. Similarly, take the lower section to y_{min} (see Fig. 3). This leaves us with four corners to work with.

Step 3: Consider the north-east corner or triangle y_{max}, z, x_{max}. For all points in this triangle find the two points with maximum x and y coordinates ($y_{max(2)}$ and $x_{max(2)}$) again and connect the elastic band from y_{max} to x_{max} via these new points (see Fig. 4).

Step 4: Repeat step 3 for points left "outside" the elastic band until none remain.

Step 5: Repeat steps 3 and 4 for the three remaining corners.

End

While this algorithm often works correctly it is in fact incorrect and it is easy to find sets

with this type of heuristic that are even slower (see Shamos [1]).

2.2 Kinesthetic heuristics

Contrary to the "combinatorial" type of convex hull algorithms illustrated in section 2.1, several algorithms exist that clearly use kinesthetic heuristics. As an example consider the gift-wrapping heuristic: here we think of wrapping a very long string around the set of points (see Fig. 2). This idea gives rise to the following algorithm due to Jarvis [17].

Algorithm CH-2

Begin

Step 1: Find the point with minimum y coordinate. Put it on the list of convex hull vertices.

Step 2: To determine the next point “touched by the string” compute all angles that the x axis makes with the string through all other points and select the point corresponding to the minimum counter-clockwise angle. Add this point to the list of convex hull vertices.

Step 3: Repeat step 2 with respect to the newly found point until the starting point is encountered again.

End

The idea of "swinging" a string or stick around until it "hits" a point is of course, without doubt a kinesthetic heuristic and we "know" from our experience in the real world that this procedure will identify the first, and therefore "outermost", point in some direction. To provide a formal proof of correctness for this algorithm requires more effort than for algorithm CH-l although we are-so convinced of its correctness from the kinesthetic point of view that no proof seems required. In fact, in [17] no proof of correctness was given. A "good" kinesthetic heuristic will often have this self-evident property. However, some kinesthetic heuristics are either incorrect or not self-evident and beg us for formal clarification. Recognizing the "good" kinesthetic heuristics seems to be part of the art of good algorithm design.

On the other hand, CH-2 runs much faster than CH-l. If there are H points on the convex hull then for each such point the algorithm takes time proportional to n due to step 2. Therefore the total time taken is proportional to Hn. Since H is less than n the algorithm never takes more than n^2 units of time which is less than the n^3 units taken by algorithm CH-l.

The elastic band heuristic: Imagine that an elastic band is stretched out into the form of a large circle that contains all the points of Fig. 1. Now we let go of the elastic band. The band shrinks until it comes into contact with the convex hull vertices. Finally at rest the elastic band defines the convex hull.

Again, this is a strong kinesthetic heuristic evoking in us such body-syntonic notions as encircling, shrinking and stretching, coming into contact with, blocking, etc. Of course, the heuristic is only the germ of the algorithm. There is a gap which must be bridged when we jump from the heuristic to the precise description of the algorithm in the language of constructive mathematics. For example, in algorithm CH-2 “swinging the string until it hits a point” was described mathematically as “computing angles and selecting the minimum angle.” Similarly, with the elastic band we must find methods for implementing the heuristic. Several possibilities exist. The shrinking of the

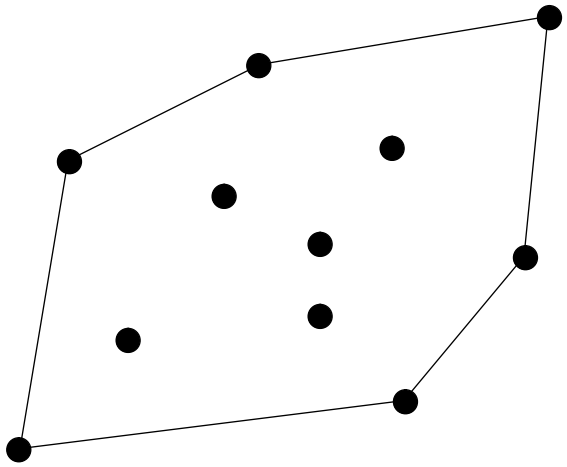

Fig. 1

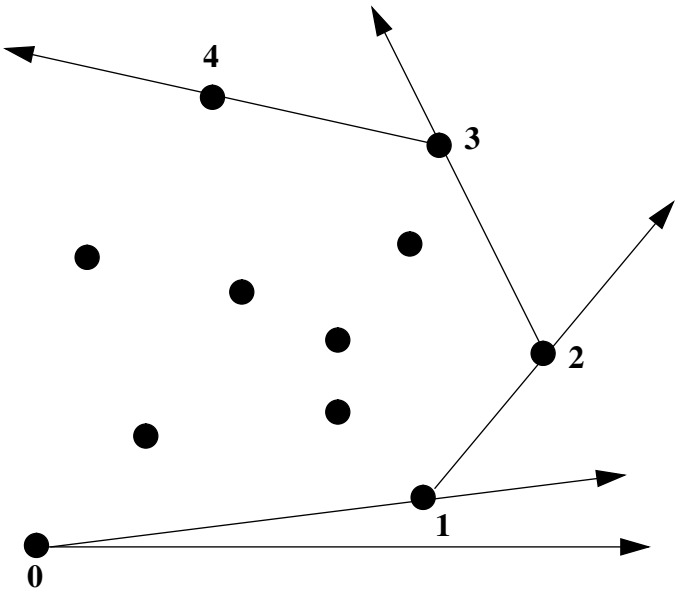

Fig. 2

clusions are based on many more which will be included in an expanded later version of this paper.

2. The Convex Hull Problem

2.1 Logico-mathematical heuristics

The convex hull of a set of n points on the plane P = $\{p_1, p_2,..., p_n\}$ is defined as the minimum area convex polygon CH(P) enclosing the set (see Fig. 1). We are interested in finding this convex polygon for a given set P.

Logico-mathematical heuristic: If two points p_i,p_j are such that all the n-2 remaining points in P lie on the same half-plane determined by the line through p_i and p_j then [p_i,p_j] forms an edge of the convex hull of P. This property leads to the following algorithm due to Rosenberg and Landgridge [16].

Algorithm CH-l

Begin

Step 1: Through each pair of points draw a line.

Step 2: For each line drawn in step 1 test to see if the remaining points all lie to one side If they do then add the pair to a list containing the edges of the convex hull.

End

This algorithm is typical of the logico-mathematical type. It is easy to prove its correctness and easy to program. The heuristic or property is clearly a logico-mathematical (in fact combinatoric) one. It could be, however, classified as a visual heuristic but it is clearly not kinesthetic. However the algorithm is very slow for large inputs. Because the number of lines drawn in step 1 is proportional to n and for each of these the number of points tested in step 2 is proportional to n, the resulting total time taken by the algorithm is proportional to n^3. Several other algorithms exist

metric algorithms offers a rich environment for psychologists and artificial intelligence researchers to study because of the strong visual and kinetic components present in geometry problems. This is even more so due to the coming of the robotics age where vision and movement are two of the crucial problems encountered. This paper presents some views on this topic and conclusions regarding both the design and analysis of geometric algorithms as well as implications for education. It is by no means a report of rigorous psychological testing on subjects, but rather an account of a personal study on the subject in the spirit of Hadamard [7], Mach [27] and Lakatos [8].

1.3 Logico-mathematical vs. kinesthetic or body-syntonic heuristics

It is useful to distinguish between various forms of knowledge [9], intelligence [10] and thinking [11]. While it is true that our knowledge, intelligence, and thinking, particularly that concerned with geometry, are originally acquired through our visual, tactual and other sensory interaction with our environment, it is also true that eventually, as we acquire idealized concepts, our logico-mathematical "machine" takes on a "life of its own" and we are therefore capable of doing logico-mathematical thinking with a minimum of "interference" from the "real world." This is a "double-edged sword": it has its advantages and disadvantages. In this paper we will distinguish between *logico-mathematical* thinking and *kinesthetic* thinking. Similarly we may speak of logico-mathematical versus kinesthetic knowledge or information. While the boundary between the two forms is necessarily fuzzy there are nevertheless instances where they are quite distinct as will become evident when we consider the design of geometric algorithms. It is also useful to distinguish between kinesthetic and visual thinking [12], [31] - [36] although the boundary here is also fuzzy and motor theories of vision have been proposed [24]. That tactual kinesthetic information exists quite apart from visual information was dramatically demonstrated recently by Louise Pelland [13], [14]. She carried out a study in which two groups of students had to draw an object (an artichoke cut in half). One group spent some time manipulating, touching, and feeling the object before drawing it. The other group spent the same amount of time looking at it from different points of view and imagining that they were touching and feeling the object. The drawings were then ranked by professional artists according to various criteria. With significant results of statistical tests she demonstrated that the "touchers" had a considerably improved drawing skill acquisition over the "non-touchers."

In this paper we review some algorithms from the computational geometry literature with the aim of classifying them according to whether they were designed using logico-mathematical heuristics or kinesthetic heuristics. The latter type of heuristic will also be referred to as body-syntonic, a term used by Papert [15] for a similar notion.

We will explore the idea that kinesthetic heuristics lead to faster geometric algorithms but are usually more difficult to prove correct. On the other hand some kinesthetic heuristics are intuitively extremely convincing. This has implications for what constitutes the notion of a "proof" of correctness. We also discuss the implications of kinesthetic heuristics for education.

Due to time constraints we illustrate these ideas with only a token of examples. The con-

On the Role of Logical, Visual and Kinesthetic Thinking in Geometry[1]

Godfried Toussaint

School of Computer Science
McGill University
Montreal

ABSTRACT

Computational geometry is a new branch of knowledge in computer science that deals with the analysis and design of algorithms for solving geometric problems. These problems typically arise in computer graphics, image processing, computer vision, and robotics. In the past few years many of the algorithms proposed for solving geometric problems have been found to be incorrect. These incorrect algorithms rather than being "purely mathematical" often have a strong kinesthetic component to them.This paper explores the relationship between computational geometric thinking and kinesthetic thinking and its implications for education.

Key Words and Phrases: convex hull problem, diameter, kinesthetic heuristics, kinesthetic thinking, mathematical discovery, algorithm design and analysis, computational geometry, artificial intelligence.

CR Categories: 3.36, 3.63, 5.25, 5.32, 5.5

1. Introduction

1.1 Computational geometry

Computational geometry is claimed by many computer scientists to be a new field of computer science which began with the thesis of Michael Shamos [l] in 1978. Since that time, spurred by developments in computer graphics, VLSI (Very Large Scale Integration), computer vision and robotics, it has blossomed into a dynamic discipline of its own and has given birth to several books in the area [2], [3], [28], [29]. For two existing surveys of the field see [4] and [5]. A third survey is to appear shortly [6]. Recently it has been discovered by several authors [5] that a score of published algorithms, that have been proposed for solving a variety of geometric problems, actually do not work correctly, i.e., they do not always compute what they are intended to compute. Thinking about these algorithms, how they fail, and trying to correct them has led to the ideas presented in this paper.

1.2 The psychology of discovery in computational geometry

Psychologists, mathematicians and physicists have been interested for a long time in the process of discovery in mathematics [7], [8], [27]. The process of discovery in the design of geo-

1. A preliminary version of this paper titled *Computational Geometric Thinking as Kinesthetic Thinking* was presented at the *Conference on Thinking*, Harvard University, August 1984.