Machine Recognition and Correction of Freehand Geometric Line Sketches

Shriram Revankar,
Computer Science, SUNY Buffalo, NY 14260
B. Yegnanarayana,
Computer Science, IIT Madras, INDIA, 600 036

Abstract –**We propose scale independent geometric models to recognize and correct freehand geometric line sketches. The models are defined as compositions of a set of primitives that account for freehand drawing approximations. The recognized sketches are corrected to their precise geometric shapes.**

1 INTRODUCTION

A significant part of human communication involves freehand line drawings. Information that is inherently geometric such as house plans, locations of an apartment or an item in a supermarket, electronic circuit layout, etc., are better conveyed by geometric line diagrams than by written text or speech. Even though freehand sketches are crude approximations of the intended objects and locations, we do successfully communicate through them. A successful communication involves recognizing the approximation errors such as rounded corners, imperfect straight lines, etc., that are inherent to freehand drawings, and mentally correcting them before interpreting.

The goal of our research is to facilitate similar communication between a user and a computer (or a robot). Several researchers have contributed towards this goal [1, 2, 3, 4]. They developed the models or templates that allow flexible matching to recognize the objects approximated by the input images. In some schemes the input error magnitudes were ignored by deciding that the input object corresponded to the best matched model [1, 2], whereas in the other schemes the allowable error magnitudes were controlled by empirically determined constants [3, 4]. This implies that the error magnitude in an input image is independent of the object in the image. Such models are inadequate to recognize objects in freehand drawings because in freehand drawings, the input error magnitude does depend on the geometry of the objects drawn [5]. To account for this data dependency, we develop new models to recognize various geometric shapes in a freehand drawing. These models allow flexible matching, and are defined as compositions of predefined primitives that account for the freehand drawing approximations. The magnitude of tolerated approximation error depends on the geometry of the object being matched. We also develop a facility to correct and print the recognized objects.

2 OVERVIEW

Input to our system is a set of line segments extracted from the greytone image of a hand drawn geometric line sketch. The image is initially binarized using an adaptive threshold [6], where regional morphology is used to vary a default global threshold [7]. Approximate straight line segments are extracted from the binarized image by tracing the centers of pixel runs, and the direction of trace is controlled by the average line width of the sketch.

The extracted line segments are represented by a graph structure [8]. The graph makes the intersegment relations explicit, facilitates search for geometric shapes, and preserves the internal relations during correction. The nodes of the graph represent the measurable quantities such as lengths, coordinates, angles subtended by the line segments, and the links represent the relations between the connected nodes. To avoid recorrection of the corrected subgraph and to avoid unnecessary search, each node is tagged by ***hard***, ***soft*** or ***open*** nodes, corresponding to the corrected, uncorrected and open-loop sections of the graph. Figure 1 is a schematic diagram of a graph.

We define the geometric models for approximate matching in terms of rules that describe graphs of the various geometric shapes. A geometric shape is recognized in the input if it has a subgraph that satisfies the corresponding rule. The recognized subgraph is corrected to its precise geometry while maintaining connectivity and and size. At every stage, the corrected changes are propagated over the uncorrected section of the graph to avoid global shape distortion. The sketch corresponding to the corrected graph is printed.

ISSN# 0-7803-0233-8/91 $1.00©1991 IEEE

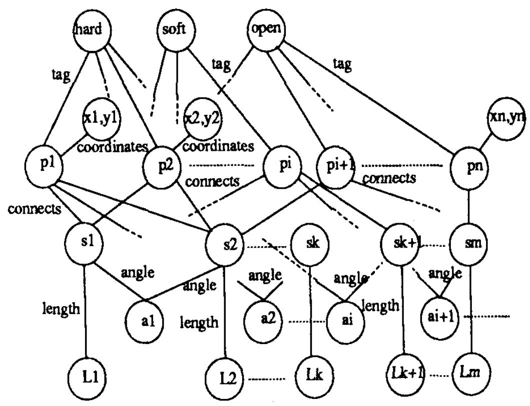


Figure 1: Graph representation of a line sketch. p1,p2,.. are end points of line segments s1, s2, etc. L1,L2,.. are the lengths and a1,a2,... are the relative orientations of the segments

3 APPROXIMATE GEOMETRIC MODELS

A geometric model is a subgraph defined as a rule with a set of conditions on internal relations of a geometric shape such as connectivity, relative orientation, equality, and parallelism. To facilitate approximate matching each relation is considered to be matched if the error is less than the corresponding threshold of approximation. The thresholds are defined as follows:

1. **Threshold of connectivity(w_o):** Two line segments are connected if the minimum distance between a pair of endpoints is less than the average line width (w_o) of the input sketch.

2. **Threshold of relative orientation (A_o):** Relative orientation between two line segments is zero, if the angle subtended by them is less than a specified constant A_o. So a line is *approximately vertical* or *approximately horizontal* if it deviates from the horizontal or vertical axis by less than A_o.

3. **Thresholds of equality (E_o):** Two quantities L_1, L_2 are *approximately equal* if $(L_1 - L_2) \leq E_o(min(L_1, L_2))$. Where E_o is an empirically determined constant with a default value 0.1.

4. **Threshold of parallelism (P_o):** Two nonadjacent sides with lengths L_1, L_2 and separated by the distances H_1, H_2 at the end points are *approximately parallel* if they subtend an angle smaller than a threshold P_o, given by
$P_o = A_o.(1 + \frac{H}{k*L}),\ if\ (H \leq k * L)$
$P_o = 2A_o,\ if\ H > k * L,$
where $H = (H_1 + H_2)/2$, $L = min(L_1, L_2)$, and k is an empirically determined constant with default value 10.

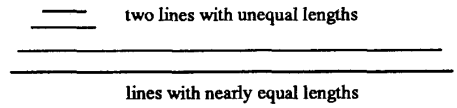


Figure 2: Both pairs of line have the same difference in length.

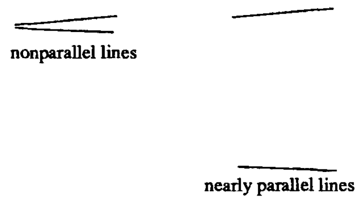


Figure 3: The same pair of line segments shown at two different distances.

These thresholds are selected because of the following observations:

- Lengths of valid line segments can not be smaller than the line width of the sketch.

- Hand-drawn lines deviate from the intended inclinations. Such deviations must be tolerated. The range of allowable deviation is specified by the threshold for relative orientation A_o.

- A difference in line lengths looks obvious if the lines are short. But the same difference becomes insignificant if the lines are longer. Figure 2 shows such an example. A similar argument can be made for difference in angles.

- The accuracy of representation of an angle subtended by two unconnected lines tends to be low, if they are separated by a large distance. This situation is shown in Figure 3. Secondly, if parallel lines to be drawn are long, one can draw them with better accuracy than if they were short. This is because a small angular deviation becomes visually obvious as the line lengths increase as shown in Figure 4. Hence the threshold

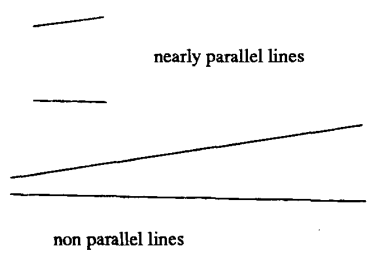


Figure 4: Sections of the same pair of lines with different lengths

selected should be proportional to the distance of separation and inversely proportional to length of the lines.

Using the above size and shape dependent thresholds, we define the models for various geometric shapes in Box 1, where *path* in a graph is a continuous traversal of connected nodes and arcs. It is called a *closed loop* if upon traversal, the starting node is reached without traversing any of the nodes on the path more than once. A *path length* is the number of segment nodes traversed in a path.

3.1 Priority ordering of rules

The subgraphs corresponding to some geometric shapes such as square, equilateral triangle, etc. satisfy more than one rule thereby causing a conflict. This conflict can be resolved by giving priorities to the rules. A priority is a *reflexive, antisymmetric* and *transitive* relation, because for any *rule a*, *a* has priority over itself; if *a* has priority over *rule b*, then *b* cannot have priority over *a*; and if *b* has priority over *rule c*, then *a* also has priority over *c*. Such a relation imposes a partial ordering over the set of rules, and this ordering can be obtained by a *Hasse* diagram [9].

1. TRIANGLE: If there is a closed loop of path length three, then it is a triangle.
2. QUADRILATERAL: If there is a closed loop of path length four, then it is a quadrilateral.
3. POLYGON: If there is a closed loop of path length 'n', then it is an 'n' sided polygon.
4. ISOSCELES TRIANGLE: If there is a closed loop of path length 3 and two angles between pairs of connected line segments are approximately equal, then it is a isosceles triangle.
5. EQUILATERAL TRIANGLE: If there is a closed loop of path length three and the angles between three pairs of connected line segments are approximately equal, then it is an equilateral triangle.
6. RIGHTANGLED TRIANGLE: If there is a closed loop of path length three and the angle between a pair of line segments is approximately equal to 90 degrees, then it is a rightangled triangle.
7. RIGHT-ISOSCELES TRIANGLE: If there is a closed loop of path length three and two of its angles between pairs of its connected line segments are approximately equal and the angle between the third pair of connected line segments is approximately equal to 90 degrees, then it is a right-isosceles triangle.
8. TRAPEZIUM: If there is a closed loop of path length four and a pair of opposite sides are approximately parallel, then it is a trapezium.
9. PARALLELOGRAM: If there is a closed loop of path length four and both pairs of alternate line segments are approximately parallel, then it is a parallelogram.
10. RECTANGLE: If there is a closed loop of path length four and both pairs of alternate line segments are approximately parallel and an angle between a pair of its connected line segments is approximately equal to 90 degrees, then it is a rectangle.
11. SQUARE: If there is a closed loop of path length four and both pairs of alternate sides are approximately parallel and consecutive sides are approximately equal and subtend an angle approximately equal to 90 degrees, then it is a square.
12. RHOMBUS: If there is a closed loop of path length four and both pairs of alternate sides are approximately parallel and consecutive sides are approximately equal, then it is a rhombus.

Box 1. Rules describing the geometric models

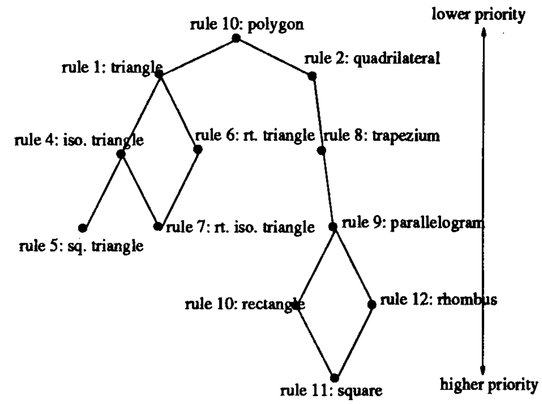


Figure 5: Hasse diagram showing the rule priorities

For simplicity of representation, let each rule be represented by its number. Then $\{1,2,3,4,5,6,7,8,9,10,11,12\}$ is the set of rules defining all the geometric models. A square pattern satisfies the set $\{2,3,8,9,10,11,12\}$. Thus a square can be viewed as partitioning the set of rules into $\{\overline{\{2,3,8,9,10,11,12\}},1,4,5,6,7\}$. All the partitions are shown in Box 2, and the corresponding Hasse diagram is shown in Figure 5. In a Hasse diagram arcs represent the priority relation and nodes represent the partitions. Larger the partition, lower is the node level.

- square: $\{\overline{\{2,3,8,9,10,11,12\}},1,4,5,6,7\}$
- rectangle: $\{\overline{\{2,3,8,9,10\}},1,4,5,6,7,11,12\}$
- rhombus: $\{\overline{\{2,3,8,9,12\}},1,4,5,6,7,10,11\}$
- parallelogram: $\{\overline{\{2,3,8,9\}},1,4,5,6,7,10,11,12\}$
- trapezium: $\{\overline{\{2,3,8\}},1,4,5,6,7,9,10,11,12\}$
- quadrilateral: $\{\overline{\{2,3\}},1,4,5,6,7,8,9,10,11,12\}$
- equilateral triangle: $\{\overline{\{1,3,4,5\}},2,6,7,8,9,10,11,12\}$
- rt. isosceles triangle: $\{\overline{\{1,3,4,6,7\}},2,5,8,9,10,11,12\}$
- isosceles triangle: $\{\overline{\{1,3,4\}},2,5,6,7,8,9,10,11,12\}$
- rightangle triangle: $\{\overline{\{1,3,6\}},2,4,5,7,8,9,10,11,12\}$
- triangle: $\{\overline{\{1,3\}},2,4,5,6,7,8,9,10,11,12\}$
- polygon: $\{1,2,3,4,5,6,7,8,9,10,11,12\}$

Box 2. Rule partitions of geometric models

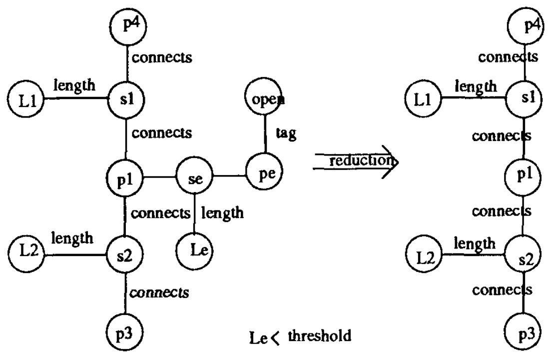


Figure 6: Deletion of a noisy line segment

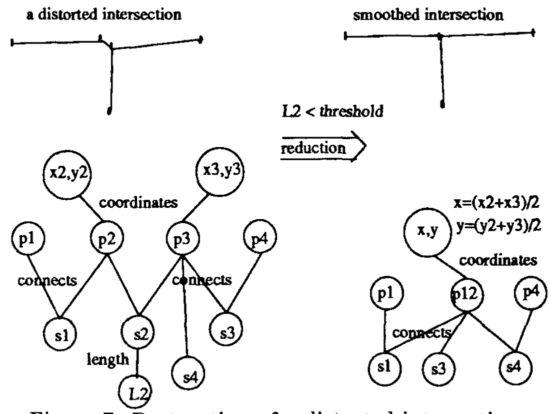


Figure 7: Restoration of a distorted intersection

The lower bound of the diagram, i.e. the rule with the largest partition is given the highest priority, and the upper bound in the diagram is given the lowest priority. This ordering is similar to the conventional *specificity ordering* [10]. The Hasse diagram has one upper bound and three lower bounds corresponding to the partitions of equilateral triangle, right isosceles triangle and square. Since these models are unrelated we can force an ordering on them without affecting the ordering of the Hasse diagram. One such forced priority relation that provides total ordering is the following:

> Square $\succ$ Rectangle $\succ$ Rhombus $\succ$ Parallelogram $\succ$ Trapezium $\succ$ Quadrilateral $\succ$ Equilateral Triangle $\succ$ Right Isosceles Triangle $\succ$ Isosceles Triangle $\succ$ right-angled triangle $\succ$ Triangle $\succ$ Polygon.

Here the symbol "$\succ$" is read as 'has the priority over'. This totally ordered relation is used for conflict resolution.

4 RECOGNITION AND CORRECTION

The recognition and correction process is a two stage process. In the first stage we use the threshold of approximations to reduce noisy and minor distortions. Figure 6 shows an example of deletion of a noisy line segment, where an open line segment is deleted if its length is smaller than the threshold of connectivity. In Figure 7 a distorted intersection and its graph representation are shown. Using the threshold w_o the graph is reduced to restore the intersection, and the correction is propagated to the connected nodes. This reduction is also done using other thresholds of approximations. After graph reduction, the nodes corresponding to the open line segments are tagged 'open' and the rest are tagged 'soft'.

In the second stage all the 'soft' tagged sections of the graphs are searched for match. The subgraphs that satisfy the rules are corrected individually to the shape of the highest priority rule they satisfy. After each step of correction, the corrected nodes are tagged 'hard', and the change in displacement and rotation are propagated over the graph to avoid cumulation of error and global shape distortion. The nodes with 'hard' tags are not recorrected.

We have implemented the graph in PROLOG, so inherently the search for a matching subgraph is done by depth first search. To correct a matched subgraph, we pick a reference line segment so that after the correction the corrected part of the sketch maintains approximately the same geometric relations with the rest of the sketch. The selection of reference line segment is governed by the following set of rules.

□

1. Select a line segment which is already corrected as the reference.
2. Select a line segment that is nearly horizontal or vertical to its corrected neighboring segments, as the reference.
3. Select a line segment that is nearly horizontal or vertical to the coordinate axis, as the reference.
4. Select a line segment as the reference.

□

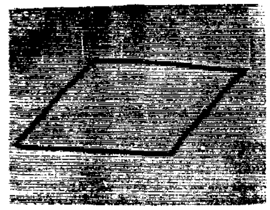
The symbol "□ . . .□" indicates that the rules within the symbol are *totally ordered* and the decision made by the earliest rule satisfied in the rule list is considered. If a rule with the highest priority is satisfied by more than one line segment of the matched subgraph, any one of them is selected as the reference.

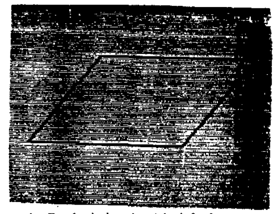
Correction is made with respect to the selected reference segment and in accordance with the relations governing the matched model. The connectivity is never destroyed because arcs defining connectivity are not updated. Even if the coordinate value of a *point* node is updated during correction, all the line segments connecting that point also refer to the updated point. Relations such as relative orientation, line lengths, etc. are also preserved, because corrective shifts given to any point are propagated over the uncorrected sections of the graph using the following geometric formulae.

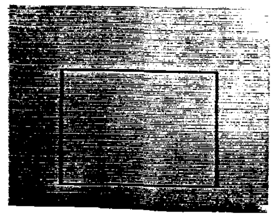
1. If a line segment $(x_1, y_1), (x_2, y_2)$ is rotated by an angle θ, then the rotated end point (x_3, y_3) is given by
   $x_3 = x_2 cos(\theta) - y_2 sin(\theta) + x_1$
   $y_3 = x_2 sin(\theta) + y_2 cos(\theta) + y_1$

2. If a line segment $(x_2, y_2), (x_3, y_3)$ of length L_2, is connected to a reference line segment $(x_1, y_1), (x_2, y_2)$ of length L_1, and if they subtend an angle β, the point (x_3, y_3) is given by
   $x_3 = x_2 - \frac{L_2}{L_1}((x_2 - x_1)cos(\beta) + (y_2 - y_1)sin(\beta))$
   $y_3 = y_2 - \frac{L_2}{L_1}((y_2 - y_1)cos(\beta) - (x_2 - x_1).in(\beta))$

The cumulative effect of various corrections might cause increase or decrease of the overall sketch size, and sometimes it may cause distortion overall shape. To suppress this cumulative effect, approximately equal sides and angles are averaged out before correction, thereby maintaining the approximate size of the sketch along with the shape.


a. Input greytone image


b. Drafted sketch with default thresholds


c. Drafted sketch with the threshold $A_o = 50^o$.

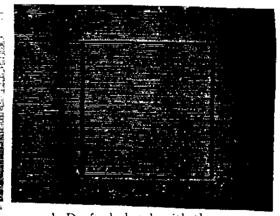
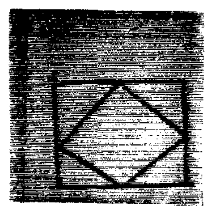

d. Drafted sketch with the thresholds $E_o = 0.25$ and $A_o = 50^o$.

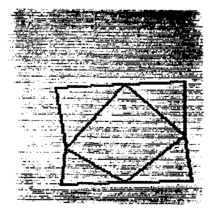
Figure 8: Drafted outputs of a quadrilateral

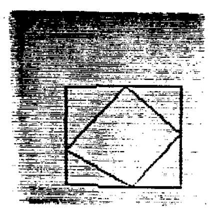
5 RESULTS

Figure 8 shows drafted sketches of a hand-drawn quadrilateral for three different thresholds of approximations. Figure 8b is the drafted version with default thresholds. There, the sketch is recognized as a parallelogram and corrected accordingly. In Figure 8c, the drafted sketch is a rectangle. That is because of a large threshold of relative orientation ($A_o = 50^o$). With the same A_o, when the threshold for equality E_o is raised to 0.25, the input is recognized and corrected as a square(Figure 8d).

The sketch in Figure 9a can be viewed either as a overlay of two quadrilaterals or as a combination of four triangles. This context can be forced on our system. Figure 9c is a drafted version with the default context, where every recognition is contextually consistent. There, the sketch is recognized as a combination of a quadrilateral and four triangles. Figure 9d is the drafted version when the context specified is "presence of only quadrilaterals" with threshold for equality 0.15. There, the inner quadrilateral is corrected as a square and rest of the line segments are corrected with reference to this square. In this context none of the triangles are recognized.


a. Input greytone image


b. Extracted line segments


c. Drafted sketch with the default context.

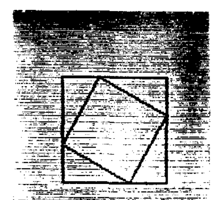

d. Drafted sketch with the context of only quadrilaterals.

Figure 9: Context dependent drafts of an input sketch

6 CONCLUSIONS

This work is a contribution towards facilitating natural man-machine communication. We have proposed new models for automatic recognition and correction of free hand geometric line sketches. While the techniques are suggested to solve some of the basic problems, the issues involved in the development of a full fledged automatic drafting system are not examined. Initially the line segments are extracted from the input greytone images of a line sketch. The extracted features are represented as a graph. The geometric shapes are recognized by flexible matching of graphs of geometric shapes with subgraphs of the input sketch. All matched subgraphs are corrected while maintaining the global shape and size of the sketch.

References

[1] H. Murase and T. Wakahara. Online hand sketched figure recognition. *Pattern Recognition*, 19(2):147–160, 1986.

[2] W.C. Liu and J.H. Pun. Machine recognition and plotting of hand sketched line figures. *IEEE Transactions on System, Man and Cybernetics*, SMC-8(1):52–57, January 1978.

[3] T.L. Huntsberger, C. Rangarajan, and S.N. Jayaramamurthy. Representation of uncertainity in computer vision using fuzzy sets. *IEEE Transactions on Computers*, C-35(2):145–156, February 1986.

[4] Linda C. Shapiro. Inexact pattern matching in esp. In *International Conference on Pattern Recognition*, pages 1128–1130, 1984.

[5] Shriram Revankar. Computer drafting of hand drawn line sketches. Master's thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, August 1987.

[6] Shriram Revankar and David B. Sher. Morphologically influenced adaptive threshold to extract myocardium from a sequence of two-dimensional echocardiograms. In *Proceedings: Fifth Annual Graduate Conference in Computer Science, State University of New York at Buffalo*, pages 31–41, March 23 1990.

[7] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on System Man and Cybernetics*, SMC-9(1):62–66, 1979.

[8] Z.M. Wojcik. A natural approach in image processing and pattern recognition: Rotation neighborhood technique, self adapting threshold, segmentation and shape recognition. *Pattern Recognition*, 18(5):299–326, 1985.

[9] Zvi Kohavi. *Switching and Automata Theory*. McGraw Hill Inc. New York, second edition, 1978.

[10] P.H. Winston. *Artificial Intelligence*. Adison Wesley Publishing Co., 1984.