Pattern Recognition, Vol. 25, No. 2, pp. 165–172, 1992
Printed in Great Britain

0031–3203/92 $5.00 + .00
Pergamon Press plc
Pattern Recognition Society

ON SEQUENTIAL SHAPE DESCRIPTIONS

H. V. JAGADISH and A. M. BRUCKSTEIN*
AT&T Bell Laboratories, Murray Hill, NJ 07974, U.S.A.

(*Received 27 August* 1990; *in revised form 5 June* 1991; *received for publication* 20 *June* 1991)

Abstract—Given a shape, we wish to describe it as the union and/or difference of primitive, possibly parameterized, shapes that constitute an alphabet. We would like this description to be ordered such that "most" of the description is conveyed within the first few terms of the description. In other words, we want as small an error as possible for any possible truncation of a description. We present a new criterion for evaluating such sequential descriptions.

For the specific case of right-angled, or rectilinear, polygons in a plane, and using only a rectangle as the primitive shape, we present an algorithm for finding optimal sequential descriptions. Though the running time of this algorithm is exponential in the worst case, we show how running time can be traded off against optimality, and how "reasonable" solutions can be found quickly.

Planar shape   Rectilinear shape   Rectangular cover   Greedy algorithm
Branch-and-bound   Approximate match

1. INTRODUCTION

When we look at a planar shape and want to describe it, say over the phone, we usually rely on a knowledge base shared with the listener and come up with very succinct descriptions. We could say something like: this is a T-shaped object with a long and thin vertical line and a short and thick horizontal line. Such a description, although qualitative, is often enough for practical purposes. Computers are involved in many shape analysis and description tasks, however, they have not yet reached a level of intelligence that would enable them to provide such simple and short but sufficient qualitative descriptions.

To mimic the qualitative description process described above we should define a knowledge base that will be used as an "alphabet" of shapes and simple operations of combining shapes and describing their spatial layout and relationships. This idea has been used in computer-aided design, where shapes are often built as combinations of instances of a basic repertoire of primitives. The instancing provides their position and the combination operations can be either union or set-difference applied to instanced primitives. We could model the shape description discussed above as a similar process, however, it is clear that what we have to do first is solve an inverse problem: given the complete shape decompose it into simpler primitives. Then we may give the resulting description to a partner. It is this inverse problem that we study in this paper. To this end, we have first to come up with ways to evaluate the multitude of possible descriptions, and then to choose the one that best suits our purpose.

An obvious application of our work is the description of an image over a slow network. This sort of problem is faced, for example, if a remote user on a 1200 or 2400 baud line wishes to browse an image database. It is discussed in reference [1] that, at least in the context of documents, what the user wants is a quick idea of the image layout, with further refinement being produced on demand. Similarly, many of us have often found ourselves in a situation where we wanted a quick printout of a paper that we were writing, simply to check that the margins were correct, and that no text or display overflowed columns. If the page image were represented in the sequential description form we suggest here, after a fraction of the description has been conveyed to the printer, we could truncate the communication, and obtain the results that we were interested in.

In a different context, consider the updating of the screen after a change, or in response to a redraw command. Depending on the speed of the link to the monitor, this activity could take up to several seconds. In most cases this update is either in scan order, say from top to bottom, or on an "object" basis, one object at a time. In neither case does the user have a reasonable idea of what the updated screen will look like until most of the update is complete. Using our sequential description method the most significant aspects of the screen could be drawn first, permitting the user to make an estimate of the final form of the screen well in advance of the redraw completion.

In reference (2), it has been suggested that a gross description of a page in terms of rectangles can be

* Visiting from The Computer Science Department, Technion, Haifa, Israel.

used effectively to determine columns, headings, and such other page layout features. Once more, one can consider this a truncated sequential description.

On a more speculative note, scattered noise in imges, such as "salt and pepper" noise, is likely to be eliminated if we obtain a good sequential description of the image and then truncate it at some empirically determined threshold. Clearly, the possibility remains of excluding important image features and of retaining some noise, just as in every other automated noise removal technique. Further work is required to compare "sequential description truncation" techniques with other more standard techniques such as low-pass filtering.

Finally, shapes with the same prefix in their sequential description are expected to be similar, with the similarity increasing as the length of the prefix considered grows. However, "similarity" is not easily defined, so we defer further discussion of this matter to reference (3). For a lucidly written article describing the conceptual problems involved, see reference (4).

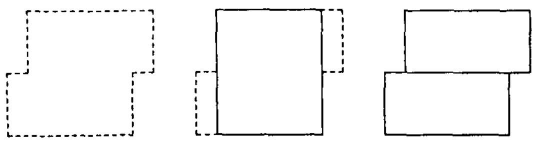
There is considerable previous work on describing a shape in terms of component primitives.[5-10] All the works mentioned above consider the problem of finding succinct descriptions of shapes in terms of primitives. However, none of them considers the important issue of sequentiality, i.e. of obtaining good descriptions very quickly.

This paper is organized as follows: the next section presents the formalization of the above discussion and defines the problem in precise terms. A criterion is proposed to evaluate the quality of a sequential description. In Section 3, we turn to a specific case of the problem, considering rectilinear polygons in a plane described in terms of a union and difference of primitive rectangles. We discuss methods to find good sequential shape descriptions in this case, and compare them with descriptions devised by a human being.

2. FINDING GOOD SEQUENTIAL DESCRIPTIONS

The problem we are addressing is the following. Given a set of shapes and an alphabet of primitive shapes, we want to find descriptions of the shape in terms of combinations of instances of primitive shapes.

The description will be a string of letters in the "alphabet" of primitive shapes (rectangles). Each *character* in this string represents a primitive shape (rectangle) parameterized by its location and dimensions. The characters in the string will also be assumed to carry information on how the parameterized rectangle has to combine with the partial description already available. For instance, it may be added to the description through a point-wise union, or it may be subtracted from it. More complex operations could also be invented: the basic shape instance might be reproduced n times where n is another parameter, with some specified spacing and angles between reproductions. Clearly any given shape in some class of shapes of interest will have multiple sequential descriptions, with the alphabet of our choice. Thus the Star of David could be described as the union of two triangles; it could also be described as a hexagon minus a set of six triangles, where a single parameterized unit can describe the six triangles.



(a) (b) (c)

Fig. 1. (a) A shape; (b) its best description in one unit; (c) its best description in two units.

A sequential description is defined as a nested sequence of approximations to a given shape. In such a sequence of approximations each description could be obtained, for example, by adding or subtracting an instance of a primitive shape to the previous description. In a sequential description our aim is to provide as much information as possible as early as possible. Therefore, we would like our shape description process to give a lot of information in the first stages of the process and successively refine the knowledge on the shape as more and more is said. We would also like the first stages of our description to capture the important features of the given shape.

Note that one parameterized instance of the basic shape as defined above is not one bit of information, but rather several bits. The exact number of bits per unit depends on the precise alphabet chosen, the number of parameters used, and so forth. Formally speaking, all the preceding arguments should apply to bits rather than characters of the alphabet. However, we shall permit truncation only at character boundaries. If truncation occurs elsewhere, we shall assume that the last several bits are discarded from the truncated word, and the last complete letter of the alphabet is retained. In our treatment we shall continue to deal with characters rather than bits, with the understanding that if each unit is a constant number of bits, all results obtained will be identical. (If the number of bits in a unit can vary in the alphabet selected, then some obvious modifications are required to the arguments here.)

Clearly, if a sequential description, for every truncation of it, has error no greater than any other description with the same length, then an absolute "best" sequential description has been found. Unfortunately, such an absolute best sequential description may not exist in general. For example, in Fig. 1, the best description with one unit is not a prefix of the best description with two units, evaluated against the symmetric area error criterion described below.

When a person is given such a well-defined sequential description task, he or she will attempt to minimize the error at each step, following the steepest descent, but will also take into consideration a penalty for being wrong at many places for a considerable amount of time (measured in iterations of the sequential description). We therefore propose the following criterion for selecting good sequential descriptions.

The cumulative error criterion

> The *cumulative error* of a sequential description is a (possibly weighted) sum over the total description length of the error in the sequence of partial descriptions.

For each exact sequential description of the finite length of a given shape, the cumulative error is well defined since the error increment becomes zero after a finite number of steps. We define as the "best" sequential description the one with least cumulative error.

The choice of weights for the weighted sum depends on the application. The simplest choice is a uniform weight of one. Such a weighting gives equal importance to error at all stages of the sequential description. On the other hand, in some applications one may know that there is a minimum three unit allowance for the shape description, so that the weights should be set to zero for errors after zero, one and two units of description. Similarly, if excessively long descriptions are to be penalized, the weights can be increased as the length of description increases. As an extreme case, one could have a weight of zero for all lengths of description except a single predetermined length. The criterion then reduces to finding the best sequential description of that specified length. In general, the criterion could utilize any arbitrary function of the error at each stage, not necessarily the weighted sum.

The measure of error at any stage depends on the relative importance of various types of differences between the shape and its description. We could measure the quality of a description by the area of the symmetric difference between the shape and its description. This implies equally weighting the error due to not covering a particular area that belongs to the given planar shape and the error of covering, with the approximate description, some portion of the background. We could add to such a quality criterion other factors that are known to be important in providing shape-related information, like penalty terms for not preserving topological properties: connectivity, Euler numbers, etc. We could also give different weights to different regions within the shape to be described, and in the background area, and measure weighted symmetric differences between the shape and its description. These factors are very important in shape description, and are easy to describe but very difficult to work with.

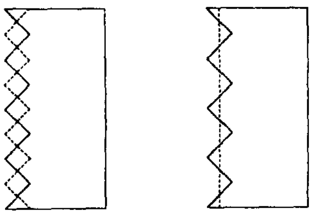


Fig. 2. A serrated shape boundary has greater error when (a) represented by a serration with phase error than when (b) represented by a straight line approximation.

Whatever error measure is chosen, there is an inherent problem with trying to produce a single quantified number representing the "distance" between two shapes in an attribute space of very high dimensionality. For example, consider serration at an edge. The area differential due to serration is not big, so it is not likely to be accounted for by one of the first few description units conveyed, given a symmetric area error measure. Even worse, serration specified with the wrong phase actually has a larger area error (see Fig. 2), even though to the human eye the description may appear closer to the shape than a description with no serration. This occurs even if serration at an edge is represented as a single attribute parameter and described in one primitive unit. Such considerations are a function of whether such low area features are important as object descriptors for human perception.

For the rest of this paper, we shall concentrate on a symmetric (Hamming distance) area difference as the error measure, and use the unweighted sum of errors as the cumulative error criterion. We shall permit only two operations with the primitives: the next unit of description may *add* a primitive to the existing approximate description, via a point-wise set union, or it may *subtract* a primitive from an existing description, via a point-wise set difference.

Let the primitive shapes be drawn from the set **R**. Then each unit of the sequential description comprises a pair ($R_i \in \mathbf{R}$, $\sigma_i \in \{+, -\}$), where the "+" represents set union and "−" represents set difference.* The description after n units is:

$$S_n = (((R_0\sigma_1R_1)\sigma_2R_2)\cdots\sigma_nR_n),$$

where R_0 is the initial empty description. Let S be the shape being approximated. The area error after n units is: $A_n = |S \text{ XOR } X_n|$. There exists some N such that $S_k = S$ (and $A_k = 0$) for all integer $k \geq N$. We

* For convenience we have assumed a large set of coordinate-specific shapes. In practice, this set **R** may never be enumerated, but instead may be derived by applying transforms, such as rotation, translation, and scaling (with appropriate parameters), to a small enumerated set of base shapes. The specific base shape selected, and the set of transforms applied are one way of describing a shape of interest in the set **R**.

wish to minimize the cumulative error of the sequential description, which is:

$$\sum_{i=1}^{\infty} A_i = \sum_{i=1}^{N-1} |S \text{ XOR } S_i|.$$

3. OBTAINING OPTIMAL DESCRIPTIONS

In this section we restrict the shapes to be described to rectilinear polygons, that is, simple polygons, not necessarily convex, with edges being either horizontal or vertical line segments. We restrict the description to be in terms of rectangles only, with four coordinates used to identify each rectangle.

In the literature, rectangles are the only shapes with which reasonable covers have been found for given shapes. In addition, rectangles are intuitively a reasonable way to describe rectilinear polygons. Therefore, we believe it is pragmatic to restrict the alphabet to consist only of rectangles. Ultimately, in graphics, every shape is reduced to a set of pixels that are each on or off. These pixels are on a rectangular grid. Therefore, every shape ultimately is represented as a rectilinear polygon. The restriction to simple polygons is not really required for the rest of this section. However, we add this restriction since it is easier to study the correspondence with human intuition for simple polygons.

In view of the *NP*-completeness of the problem of deciding whether a rectilinear polygon is coverable with K or fewer rectangles,[11] and of similar complexity results for a wealth of covering problems (see reference (7) and the references therein), it is expected that our problem requires combinatorial techniques for its solution. We present one such technique below, and also show how "good" guesses at the optimum can often be obtained very rapidly.

We shall attempt to determine a description in terms of rectangles constrained to be composed of object-induced rectangles henceforth denoted *grid primitives*. Each edge can be extended infinitely in either direction, to obtain a grid of lines in a plane. Each rectangle in this grid, called a grid primitive, is either entirely in the object, or entirely out of it. A simple description would be to enumerate each grid primitive in the object, on a grid primitive per unit of description. Clearly, one can do better. Any pair of adjacent grid primitives must also form a rectangle, and so can be included in a single unit of description. The idea, loosely speaking, is to find large clusters of grid primitives that can be described in one unit.

Each grid primitive must entirely be in or entirely be out of the target object. A sequential description that at some stage includes a part of a grid primitive must at some later stage either include the rest of it, or subtract it out. In either case, one can always obtain a description that is at least as good, in which the grid primitive is either entirely included or entirely excluded at all times. Owing to the linearity of the area criterion, either completely including or completely excluding the grid primitive is going to produce an error no larger than including part of it. Since a partially included grid primitive must at some later stage either be included completely or excluded completely, the descriptions converge after this point. Therefore, given a sequential description of a rectilinear polygon using arbitrary rectangles, one is guaranteed to be able to find a sequential description that is at least as good using only grid primitives. As such, we shall confine our attention to grid primitives in what follows.

Let us call a grid primitive *black* if it is in the object, and *white* if it is not. Also, a grid primitive is *in* if it is included in the current description of the object, and *out* if it is not. Thus, to start with, all grid primitives are either out-black or out-white. At the end, when an exact description of the object is obtained, all grid primitives are either in-black or out-white. The total area of the grid primitives that are out-black and those that are in-white gives the error in the current description. If a description is built up purely additively, then there are no grid primitives that are in-white.

We can now define the notion of *domination*:

> rectangle X is said to dominate rectangle Y for addition iff X contains every out-black grid primitive in Y, Y contains every out-white grid primitive in X, and X–Y is either empty or has at least one out-black grid primitive;
> rectangle X is said to dominate rectangle Y for subtraction iff X contains every in-white grid primitive in Y, Y contains every in-black grid primitive in X, and X–Y is either empty or has at least one in-white grid primitive.

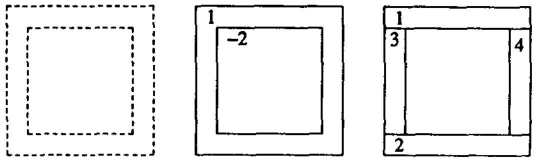
If rectangle X dominates rectangle Y for addition, we are guaranteed that the error at the current step is less if rectangle X is added rather than Y, and that the error will continue to be no greater for all future steps. To see that this is the case, recall that addition of a rectangle can cause out-black grid primitives to become in-black and out-white grid primitives to become in-white. Thus X causes every out-black grid primitive to become in-black that Y does, and possibly some more, while not creating any more in-white grid primitives than Y. Since the error is the total number of out-black and in-white primitives, the error after adding X is less than or equal to the error after adding Y to the description. If the errors are equal, then the descriptions after the additon are identical, and will continue to remain identical over any future sequence of additions and subtractions. If the errors are unequal, consider a grid primitive that is out-black after adding Y but in-black after adding X. (There must be at least one such, if the error with X is less than the error with Y.) A future addition of a rectangle may render this grid primitive in-black irrespective of whether X or Y was added. Similarly, a future subtraction of a rectangle may render this grid primitive out-black irrespective of whether X or Y was added. These future additions and subtractions are in no way constrained by the current status of this (or any other) grid primitive. As such, the error after adding Y can at best equal the error after adding X. A similar argument can be constructed when X dominates for subtraction.

At each step, one need not consider every possible rectangle for the next unit of description, but rather only those that are not dominated for addition or for subtraction. In practical terms, determining whether a rectangle is dominated is simple: first try extending it by one grid unit in each direction, one side at a time. If in at least one extension none of the grid primitives included are out-white, and at least one is out-black, then the rectangle in question is dominated for addition. Similarly if in an extension none of the grid primitives included are in-black, and at least one is in-white, then the rectangle is dominated for subtraction. Next try shrinking the rectangle by one unit in each direction, one side at a time. If in at least one such shrinkage, none of the grid primitives excluded are out-black, then the rectangle in question is dominated for addition. If in at least one shrinkage none of the grid primitives excluded are in-white, then the rectangle is dominated for subtraction. A rectangle cannot be dominated for addition (respectively, subtraction) unless dominance can be shown in one of the two steps above. The proof of correctness of this constructive procedure is simple and not presented here. The important point is that it is possible, within time proportional to the perimeter of the rectangle, to determine whether a rectangle can be dominated by another rectangle. There is no need to consider every possible other rectangle as a candidate for this purpose.

If there are a total of $2n$ edges in the object to be described, there are at most n horizontal grid lines, and at most n vertical grid lines.* (There are exactly n provided that no two edges are collinear.) The grid primitives outside the outermost grid lines are not of interest. Therefore, there are a total of at most $(n-1)^2$ grid primitives of interest.

Each rectangle is completely described by its lower-left corner grid primitive and its upper-right corner grid primitive. (The two can be equal, in the case when the rectangle comprises a single grid primitive.) Evaluating a discrete integral, we find the total number of rectangles possible is $n^2(n-1)^2/4$. Thus there are $O(n^4)$ rectangles that can be used for the first approximation to the given shape, $O(n^4)$ that can be used for the second approximation, and so forth. With the use of the concept of domination, the number of rectangle choices to be considered can be reduced, sometimes dramatically so. However, the worst case situation still forces us to consider $O(n^4)$ rectangles (see Appendix A).

* Since alternate edges must be horizontal and vertical, a right-angled polygon in a plane must have an even number of edges.



(a) (b) (c)

Fig. 3. (a) An annular shape; (b) its best description; (c) its best monotonically error reducing description.

Since we can enumerate the different rectangles that can be used at each step, we can use the branch-and-bound technique to solve the combinatorial problem. Pseudo-code for the branch-and-bound algorithm is given in Appendix B. The idea is to consider at each stage every (non-dominated) rectangle that could be used as the next element of a sequential description. The "tree" of alternatives is traversed depth-first fashion, and all choices made are remembered on a stack. Once the error has been driven down to zero following one particular branch, we backtrack and consider what would have happened if a different choice had been made. (This is the branching step.) When the cumulative error becomes greater than a bound that has already been achieved, stop exploring that path further. (This is the bounding step.)

It is tempting to believe that we need consider using only such rectangles as will decrease the error at each step: in other words that the area error is monotonically decreasing in the best sequential description. While this is certainly true in most cases, Fig. 3 presents a counter-example, in which the best sequential description actually has the error increase first and then decrease. Intuitively as well, human beings when tested preferred this best description.

In evaluating the different choices of rectangles at each step, our algorithm orders the choices such that the rectangles considered first are the ones that immediately decrease the error by the greatest amount. By thus considering the most likely candidates first, we make it likely that the true optimum will be found earlier than if the alternatives were evaluated in a purely random order. We also increase the efficacy of the bounding step, since many of the worst alternatives may not have to be considered at all.

Moreover, since the branch-and-bound is being performed in a "best-fit" fashion, it can be halted at any point after one candidate description has been found to obtain a reasonable, though not necessarily the best, answer. One can thus impose an upper bound on the amount of computer effort one is willing to spend, and walk away with a reasonable answer at any point. In fact, it is even possible to devise functions that estimate the likelihood of

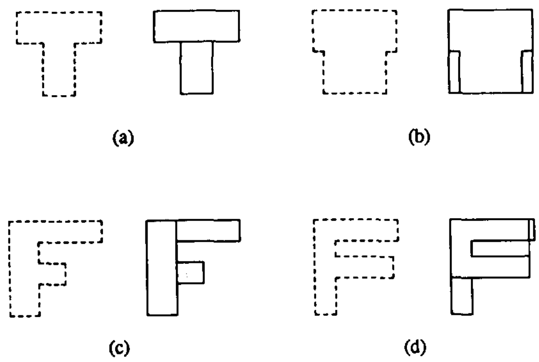


Fig. 4. Some simple shapes and their optimal descriptions.

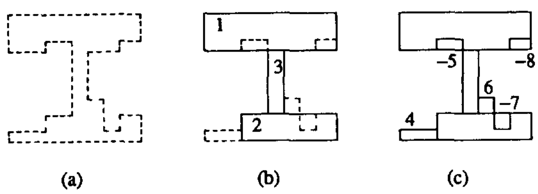


Fig. 5. (a) An example rectilinear polygon shape; (b) an intermediate stage in its description; (c) final (optimal) description.

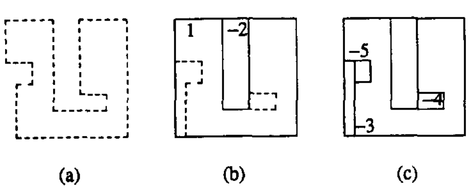


Fig. 6. (a) Another example rectilinear polygon shape; (b) an intermediate stage in its description; (c) final (optimal) description.

further improvements in the objective function, based upon improvements made in the recent past, and to use such functions to decide upon a stopping point. Note, though, that any such technique would still remain a heuristic, with no guarantees provided on the goodness of the solution obtained.

In particular, one could explore only the least error rectangle at each stage, ignoring all the others, to obtain a greedy algorithm. Even though this algorithm could be implemented as just a special case of the algorithm above, one can obtain a simpler implementation by throwing away all the book-keeping required to maintain the stacks. Pseudo-code is given in Appendix B.

Clearly, no more than n^2 rectangles can be required for a sequential description of a shape that can be placed on a grid with n horizontal and n vertical grid lines. As discussed above $O(n^4)$ choices may have to be considered for each rectangle in the description. Thus the total time required for the greedy algorithm is $O(n^6)$, where there are $O(n)$ rectilinear sides to the shape being described. In comparison, a full branch-and-bound could require time that is exponential in n.

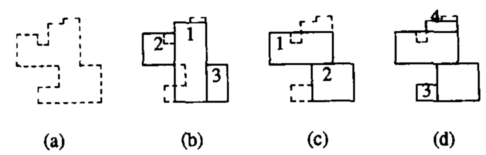


Fig. 7. (a) A final example rectilinear polygon shape, (b) the first few steps of its greedy description; (c) and (d) intermediate stages in its optimal description.

Examples

We now present several examples of the best description obtained by running on a computer the family of algorithms described above, to be compared with the descriptions preferred by people.

In Fig. 4, the optimal descriptions are presented for two familiar shapes (T and F). Observe how the optimal description changes as the relative sizes of the parts are changed. In both cases, there is some threshold where the switch-over occurs from description (a) to description (b). Where a mathematical criterion would place a sharp dividing line, humans may have a fuzzy transition. Just how "fat" must the limbs of a "T" get before you think of it as a rectangle with two little pieces knocked off rather than as a "T" shape?

Figures 5–7 present some examples of more complex shapes and their optimal descriptions. Each description rectangle is numbered according to its position in the sequential description, and has a minus sign attached if it is subtracted. In the examples depicted in Figs 5 and 6, the optimal description is also the greedy description. We found this to be the case in a large fraction of the examples that we tried. However, there are, as one should expect, many examples where the optimal description differs significantly from the greedy description. Such an example is presented in Fig. 7.

4. CONCLUSIONS

In this paper we have studied the problem of sequential description of shape, and proposed a criterion that we believe is appropriate for measuring the goodness of such sequential descriptions. Then we went on to consider a specific case of this problem, where we used addition and subtraction of rectangle primitives to describe ortholinear shapes. We presented an exact technique that could take exponential time, and a greedy technique that quickly discovers solutions close to the optimal. The descriptions obtained from the machine using these techniques were similar to descriptions generated by a human being with no knowledge of the computer results.

Acknowledgements—We would like to thank Mark Jones, Don Mitchell, Peter Selfridge, Guy Story, and especially, Bruce Ballard, for illuminating discussions on some of the ideas presented here.

REFERENCES

1. G. Nagy, Towards a structured-document-image utility, *Proc. IAPR Wkshop Syntactic Structural Pattern Recognition*, Murray Hill, NJ, pp. 293–309, June (1990).
2. H. S. Baird, S. E. Jones and S. J. Fortune, Image segmentation by shape-directed covers, AT&T Technical Memorandum, March (1990).
3. H. V. Jagadish, A retrieval technique for similar shapes, *Proc. ACM-SIGMOD Int. Conf. Management of Data*, May (1991).
4. D. Mumford, The problem of robust shape descriptors, Center for Intelligent Control Systems Report CICS-P-40, Harward University, Cambridge, MA, December (1987).
5. A. Rosenfeld (ed.), *Multiresolution Image Processing and Analysis*. Springer, Berlin (1984).
6. S.-K. Chang, Y. Cheng, S. S. Iyengar and R. L. Kashyap, A new method of image compression using irreducible covers of maximal rectangles, *IEEE Trans. Software Engng* **14**(5), 651–658 (1988).
7. D. S. Franzblau, Performance guarantees on a sweep-line heuristic for covering rectilinear polygons with rectangles, *SIAM J. Disc. Math.* **2**(3), 307–321 (1989).
8. T. Pavlidis, *Structural Pattern Recognition*. Springer, Berlin (1977).
9. I. Pitas and A. N. Venetsanopoulos, Shape decomposition by mathematical morphology, *Int. Conf. Comput. Vision*, London, U.K. (1987).
10. S. L. Tanimoto, Hierarchical picture indexing and-description, *Proc. IEEE Wkshop on Pictorial Data Description and Management*, Asilomar, CA, pp. 103–105, August (1980).
11. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco (1979).

APPENDIX A

In this appendix we show by counter-example that the idea of dominance does not significantly reduce the number of rectangles to be considered in the worst case.* Consider a shape similar to the one shown in Fig. A1, but on a large grid. The shape comprises roughly n diagonal bands in a grid of size n by n.

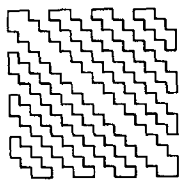


Fig. A1. A rectilinear shape with diagonal stripes.

Let us count the number of rectangles of different sizes we can create, that are not dominated at the first step of the sequential description. We need consider only rectangle additions. Consider a rectangle of height h, and width w. Provided that h and w are both at least 4, every position of such a rectangle will be valid, not dominated by any other rectangle, since an extension on any side would by necessity include at least one grid square not in the shape. But there are only $O(n)$ different rectangle sizes with at least h or w less than 4. For each rectangle size, there are no more than n^2 positions in which the rectangle can be placed on the grid. So there are at most $O(n^3)$ rectangles that are invalidated by the domination criterion, out of the $O(n^4)$ total possible number of rectangles. As such, the number of valid rectangles remains $O(n^4)$, and the use of domination does not prune too many rectangles in this example.

* We are grateful to Bob Holt for working with us to find this counter-example.

APPENDIX B

Branch-and-bound algorithm

```
Best Error = ∞;
Description Stack is Empty;
Cumulative Error Stack has the single entry 0;
  /*The top value in this stack is the current value of cumulative error*/
Best Sequence is null;
Current node = root of the search tree;
expand_node ( );
/*Best Sequence has the required best sequential description, with a cumulative error of Best Error*/

expand_node ( )
{
  Enumerate all non-dominated rectangle covers approximating the object, given the partial cover already described;
  Order these according to error, from least to greatest;
  If least error rectangle has an error of zero
    {
      Copy Stack to Best Sequence;
      Append this least error rectangle to Best Sequence;
```

```
      Best Error = Cumulative Error; /*read from top of stack*/
      return;
    }
  For each rectangle in order
    {
      if (Cumulative Error + Current Error >= Best Error) break;
      push on cumulative Error Stack, Cumulative Error + Current Error;
      Push on Description Stack the current rectangle;
      expand_node ( );
    }
  pop Description Stack;
  pop Cumulative Error Stack;
}
```

Greedy algorithm

```
Best Sequence is null
expand_node ( );

expand_node ( )
{
  Enumerate all non-dominated rectangle covers approximating the object,
  given the partial cover already described;
  Append least error rectangle to best sequence;
  If least error rectangle has an error of zero, return;
  Else expand_node ( );
}
```

About the Author—H. V. JAGADISH obtained his Ph.D. from Stanford University in 1985, and has since been with the Computing Systems Research Center in AT&T Bell Laboratories, Murray Hill, NJ. His focus is in the area of information management, particularly in object-oriented and multi-media databases. Specifically, he has recently been interested in the use of database for managing new forms of information, such as images, and the interaction between data management facilities and pattern recognition.

About the Author—ALFRED M. BRUCKSTEIN was born in Sighet, Transylvania, Romania, on 24 January 1954. He received B.Sc. and M.Sc. degrees in Electrical Engineering, from the Technion, Israel Institute of Technology, in 1977 and 1980, respectively, and a Ph.D. degree in Electrical Engineering from Stanford University, Stanford, CA, in 1984.

From October 1984 until June 1989 he was with the Faculty of Electrical Engineering at the Technion, Haifa. Presently he is with the Faculty of Computer Science there. His research interests are in computer vision, image processing, estimation theory, signal processing, algorithmic aspects of inverse scattering, point processes and mathematical models in neurophysiology.

Professor Bruckstein is a member of SIAM, MAA and AMS.