Segmentation of Plane Curves

THEODOSIOS PAVLIDIS, MEMBER, IEEE, AND STEVEN L. HOROWITZ

Abstract—**Piecewise approximation is described as a way of feature extraction, data compaction, and noise filtering of boundaries of regions of pictures and waveforms. A new fast algorithm is proposed which allows for a variable number of segments. After an arbitrary initial choice, segments are split or merged in order to drive the error norm under a prespecified bound. Results of computer experiments with cell outlines and electrocardiograms are reported.**

Index Terms—**Boundary segmentation, data compaction, feature extracton, pattern recognition, piecewise functional approximation, polygonal contours, waveform segmentation.**

I. INTRODUCTION

THE mathematical approximation of plane curves is important in many aspects of pattern recognition, image processing, and graphics. Such curves occur either as boundaries of regions of a picture or a map or as waveforms. In both cases describing them in a compact mathematical way is useful towards at least three goals:

1) feature extraction (shape descriptors),
2) data compaction, and
3) noise filtering.

The pattern recognition literature contains many examples of applications of this problem [1]–[10]. It is fairly obvious that, unless the curve to be approximated is rather simple, polynomial approximations or series expansions are inadequate [11], [12]. A preferable approach is approximation by piecewise polynomial functions where the breakpoints are adjusted in order to fit the data. There is substantial mathematical literature on the subject in connection with the problem of spline approximation with variable knots [13]–[15]. However the strong continuity requirements of the latter are not always relevant in pattern recognition related problems where discontinuities are inherent in the nature of the data. Even where continuity of the approximation is desirable there are a number of reasons for locating the breakpoints while ignoring continuity.

1) In a number of cases, especially for piecewise linear approximations, continuity in many of the breakpoints is a property of the unconstrained optimal approximation [16]. A simple case where one can easily verify this property is the piecewise linear approximation of convex functions [17], [18].

Manuscript received May 28, 1973; revised January 7, 1974. This work was supported by NSF Grant GK-36180.

The authors are with the Department of Electrical Engineering and Computer Science Laboratory, Princeton University, Princeton, N.J.

2) The computational effort under the continuity constraint is greater than when it is absent [19].

3) Continuous spline approximations with variable knots, do not have a unique optimum [10], [13], [15].

Thus it seems more efficient to search for piecewise polynomial approximation of a given curve without the continuity constraints and then adjust locally the approximation in order to achieve continuity. Although such a solution will not be necessarily optimal it can be achieved with greater speed than the optimal solution.

This paper is a further extension of [20] and considers the following problem. Given a set of points $S = \{x_i,y_i \mid i = 1,2\cdots N\}$ determine the minimum number n such that S is divided in n subsets $S_1,S_2\cdots S_n$, where on each of them the data points are approximated by a polynomial of order at most $m - 1$ with an error norm less than a prespecified quantity e.

This seems to satisfy the intuitive notice of curve fitting in a more satisfactory way than other proposed measures, as in finding a minimum length polygonal curve [4], [5]. Furthermore the computational requirements of the method are comparable if not lesser than those of various suboptimal techniques [4], [5], [8].

The choice of the error norm depends to some extent on the application and we will review that question briefly in the next section. Then we will describe an algorithm for solving the problem followed by examples of implementation.

In most of the sequel we will restrict ourselves to piecewise linear approximations, i.e., $m = 2$. However the algorithm discussed in Section IV is valid for the general case and even for approximations of the form

$$\sum_{i=1}^{m} a_i\varphi_i(x) \tag{1}$$

where $\varphi_i(x)$ is a set of linearly independent functions.

II. THE CHOICE OF AN ERROR NORM

First we have to distinguish between waveforms and arbitrary plane curves. In the former case the pointwise error can be defined along a coordinate, namely,

$$e_i = \mid y_i - p(x_i) \mid \tag{2}$$

where $p(x)$ is the approximating polynomial. Such a measure is not adequate in the second case where it is more meaningful to define e_i as the Euclidean distance between $\{x_i,y_i\}$ and the approximating curve evaluated at that point. If this curve is a line with equation

$$\sin \varphi \cdot x + \cos \varphi \cdot y = d$$

then we have [21]

$$e_i = | \sin \varphi \cdot x_i + \cos\varphi \cdot y_i - d |. \quad (3)$$

Note that φ is the angle of the line with the x axis and d the distance of the line from the origin (Fig. 1).

The error norm over a point set S_k can be defined also in many ways. Two of the most common ones are

1) integral square error E_2, where

$$E_2 = \sum e_i^2 \qquad \text{for } (x_i,y_i) \epsilon S_k \quad (4)$$

and 2) maximum error

$$E_\infty = \max (e_i) \qquad \text{for } (x_i,y_i) \epsilon S_k. \quad (5)$$

The latter measure results in approximations which in some cases present a closer fit as judged visually [20] but it involves considerably more computational effort. Finding the approximation when e_i is given by (2) is a linear programming problem [20], [22]–[24] while when e_i is given by (3) it is a mathematical programming problem involving the minimization of a linear functional subject to both linear and quadratic constraints. On the other hand, there are explicit formulas for E_2 whether e_i is given by (2) or (3) [9], [25]. It can be shown easily that in the latter case E_2 becomes minimum if φ and d are chosen as follows:

$$\sin 2\varphi(V_{xx} - V_{yy}) + 2 \cos 2\varphi V_{xy} = 0 \quad (6)$$

$$d = \sin \varphi \cdot V_x + \cos \varphi \cdot V_y \quad (7)$$

where, if N_k is the number of points in S_k

$$V_x = \frac{1}{N_k} \sum_{S_k} x_i \quad v_y = \frac{1}{N_k} \sum_{S_k} y_i$$

$$V_{xx} = \sum_{S_k} (x_i - V_x)^2 \quad V_{yy} = \sum_{S_k} (y_i - V_y)^2$$

$$v_{xy} = \sum_{S_k} (x_i - V_x)(y_i - V_y). \quad (8)$$

The error norm E_2 is also given by

$$E_2 = \sin^2 \varphi \cdot V_{xx} + \cos^2 \varphi \cdot V_{yy} + \sin 2\varphi \cdot V_{xy}. \quad (9)$$

It is important to notice that during an iterative procedure where the sets S_k vary, one can update the quantities given by (8), without having to recalculate the sums over each S_k. This is done by simply adding or subtracting the contributions of the added or removed points.

Furthermore we shall see that only the quantity E_2 for each S_k is needed during that stage and φ and d need be calculated explicitly only at the end. Eliminating φ between (6) and (9) we obtain:

$$E_2 = \tfrac{1}{2}\{V_{xx} + V_{yy} - [(V_{xx} - V_{yy})^2 + 4V_{xy}^2]^{1/2}\}. \quad (10)$$

If a table of trigonometric functions is available during the computation then it is simpler to find φ from (6) and then evaluate E_2 from (9). If not then it may be preferable to evaluate the square root of (10). It is also easy to show that E_2 is the smallest root of the quadratic equation

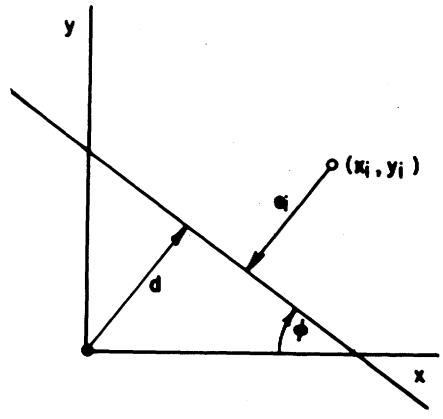


Fig. 1. Illustration of parameters used in (3).

$$Z^2 + (V_{xx} + V_{yy})Z + V_{xx}V_{yy} - V_{xy}^2 = 0. \quad (11)$$

Sensitivity to noise is an important difference between the E_2 and E_∞ norms. The curve $y = \cos(x)$ on the interval $[0,2\pi k]$ is fitted by the line $y = 0$ by both methods. However E_2 increases proportionately with increasing integral values of k whereas E_∞ remains constant.

III. MINIMIZING THE ERROR NORM FOR A GIVEN n

The problem of finding the "best" piecewise polynomial approximation can be expressed in either of two ways: 1) find the smallest n so that the error norm does not exceed a given bound [2], [8], [10], [17], [18], [26], [27] or 2) for a given n minimize the error norm. Methods to solve this problem have been reviewed elsewhere [20] and algorithms relevant to the present application can also be found in the literature [20], [28], [29]. Methods of the first type almost always involve a linear scan of the points to be approximated, possibly with repetitions. Except for suboptimal approximations or simple special cases [2], [17], [18], [27] the computational requirements of the methods are of order $N \log N$ or N^2 [8], [26]. Methods of the second type proceed with small changes in the locations of the breakpoints and in the worst case the number of iterations required is of order N (when all the breakpoints are clustered in the "wrong" end and have to be moved across the whole array of data points). Thus initialization is crucial. However that in itself is a nontrivial problem and one of the proposed schemes in the literature is completely satisfactory [20], [26]. In the next section we propose an algorithm which not only makes "good" initialization unnecessary but also allows for a variable number of segments. Thus it attacks the problem of the first type using algorithms of the second type with a significant overall increase in speed.

IV. A "SPLIT-AND-MERGE" ALGORITHM

The process of breakpoint adjustment can be accelerated substantially if one is allowed to merge adjacent segments with similar approximating coefficients and split segments with large error norms. In this way, it is easy to obtain segmentations where the error norm on each

segment (or over all of them) does not exceed a specified bound.

Merging of segments as described above results in controlled increases of the error norm. This can be seen from the following two propositions.

Proposition 1: Let S_1 and S_2 be two adjacent segments such that the maximum error of approximation on each one of them does not exceed ϵ. The maximum error of approximation on $S_1 U S_2$ is bounded from above by

$$\epsilon + \delta L \tag{12}$$

where

$$\delta = [\sum_{i=1}^{m} (a_i^1 - a_i^2)^2]^{1/2} \tag{13}$$

and

$$L = \max [\sum_{i=1}^{m} \varphi_i(t_j)^2]^{1/2} \qquad \text{for } t_j \epsilon S_1 U S_2. \tag{14}$$

The coefficients a_i and functions φ_i are as in (1).

This is a special case of a general result proven elsewhere [7]. For piecewise linear approximation, L is bounded from above by the length of the arc $S_1 U S_2$ [7].

A similar result can be obtained for the integral square error.

Proposition 2: Let E_2^0 be the maximum error norm over S_1 and S_2 and let δ and L be defined as in (13) and (14). Then if E_2^v denotes the error norm over $S_1 U S_2$ the following holds:

$$E_2^v \leq 2E_2^0 + \delta^2 L^2 M \tag{15}$$

where M is the number of points in $S_1 U S_2$. For piecewise linear approximations

$$E_2^v \leq 2E_2^0 + 2\delta^2 L^3. \tag{16}$$

The proof is given in the Appendix. The above bounds are quite conservative in practice but they serve as a guideline for the algorithm described next.

Let $S_1, S_2, \cdots, S_{n_r}$ be the segments at the rth iteration and $E^1, E^2, \cdots, E^{n_r}$ the respective error norms. We distinguish two cases. In one, it is required to have $E^i \leq E^{\max}$ for all segments. In the other, it is necessary to have the error norm E over all the segments less than or equal to $E^{\max}$. In the case of uniform approximation

$$E_\infty = \max_i E_\infty^i$$

and, therefore, the two cases coincide. For the E_2 norm, they are distinct since

$$E_2 = \sum_{i=1}^{n_r} E_2^i.$$

Let R by an algorithm for endpoint adjustment such that on each iteration it cannot reverse any inequality of the form $E^i \leq E^{\max}$ or $E \leq E^{\max}$ [20], [29].

The following is the "split-and-merge" procedure for the first case.

Step 1: For $i = 1,2,\cdots n_r$ check if E^i exceeds $E^{\max}$. If it does split the ith interval into two and increment n_r. The dividing point is determined by Rule A below. Calculate the error norms on each new interval.

Step 2: For $i = 1,2,\cdots n_r - 1$ merge segments S_i and S_{i+1} provided that this will result in a new segment with $E^i \leq E^{\max}$. Then decrease n_r by one and calculate the error norm on the new interval.

Step 3: Do one iteration of algorithm R.

Step 4: If no changes in the segments have occurred in any of Steps 1–3 above, then *terminate*. Or else go to Step 1.

Rule A: If two or more points are known where the pointwise error is maximum then use as a dividing point the midpoint between a pair of them. Otherwise divide S_i in half.

Remarks: In Step 1, i may not be incremented after a split and in this way all such splits can be made in one pass.

In Step 2, one can either try to merge all pairs of adjacent segments and cancel the merge if $E^i > E^{\max}$ or he can use Propositions 1 or 2 above to select pairs for which a merge will be attempted.

In Step 4, it is usually necessary to check only if endpoint adjustments have been made in Step 3 since this is the last "active" step (see proof of theorem below).

For the second case, the "split-and-merge" procedure takes the following form.

Step 1a: If $E > E^{\max}$ then go to Step 1b, or else go to Step 2.

Step 1b: Find the segment with the largest E^i and split it into two as in Step 1 of the first case. Then go to Step 1a.

Step 2: Find the pair of adjacent segments whose merge will cause the smallest increase in E and merge them if this will not cause E to exceed $E^{\max}$. Repeat till no further mergers are possible.

Step 3: Do one iteration of algorithm R.

Step 4: If no changes in the segments have occurred in Steps 2 or 3, then *terminate*. Or else go to Step 1.

It should be emphasized that the above procedures have a dual goal. First they find a local minimum for the number of segments so that the constraints on the error norm(s) are satisfied. However, for that number of segments there is a range of values for the error norm and the procedure, for the types of R mentioned above, proceeds next to minimize the latter. The minima are again local.

Theorem: The above procedure is an algorithm (in both cases) and it terminates after a local minimum for the number of segments is reached and the criterion for the termination of the algorithm R is satisfied.

Proof: Obviously the error norm is zero on each segment which has at most as many points as degrees of freedom in the approximating polynomial. Therefore, after a sufficient number of split operations all the E^i's will become less or equal to $E^{\max}$ or sufficiently small to have $E \leq E^{\max}$. Once this set of inequalities is satisfied it cannot be reversed because of the restrictions in Steps 2 and

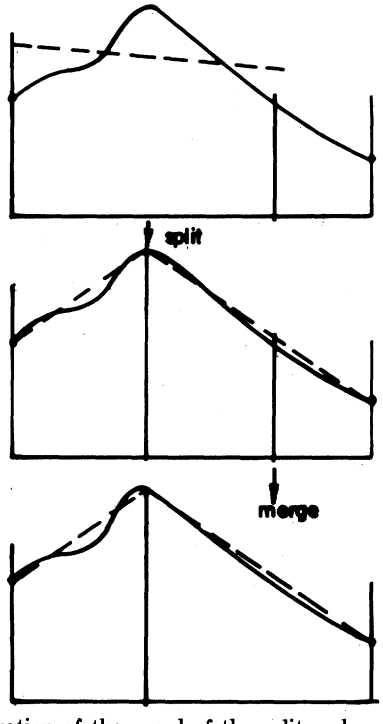


Fig. 2. Illustration of the speed of the split-and-merge algorithm. An optimum segmentation can be found in one iteration.

3. Thus eventually no modification of the segments will be made when Step 1 is applied. Step 2 will also become inactive after a certain number of iterations since at most all segments can be merged into one. Then the split-and-merge procedure will become equivalent to the R algorithm which has been assumed to terminate. Note that by the very nature of Step 2 any increase in the number of segments will cause at least one E^i to exceed E^{max} (or E to exceed E^{max}). Hence the final segmentation has a (locally) minimum number of segments.

Computational experience indicates that the local minima given by the "split-and-merge" algorithm are quite close to the global minima.

After the first few iterations the algorithm is reduced essentially to R which only adjusts the endpoints. However, it is no longer necessary to move an endpoint across the whole interval [20]. For example, consider the situation shown in Fig. 2. After one iteration the left interval will be split in two and in the next iteration two pairs will be merged to give the final answer with hardly any endpoint adjustments. Thus the "longest move" of a breakpoint will have to be made across an interval after the split-and-merge procedure is over. Let L be the number of points of the largest interval where the error norm is less than E^{max}. Then one might have a situation where the initial segmentation includes most of it with another segment as in Fig. 3. The dividing point must be moved from the position D to D' without the help of the split-and-merge procedure. Thus the number of iterations will be proportional to L (worst case). Note that L is in general independent of N since it depends on the form of data only. The average number of points per segment N/n should be of the same order as L. Since the computation per iteration in the case of the E_2 norm is proportional to n the total time of computation should be of order N. Thus the split-and-merge algorithm is linear in N, even in the worst case.

The "split" operation is computationally expensive since it requires reevaluations of integrals. However computational experience shows that it occurs only in the first couple of iterations. The "merge" operation can be controlled by the bounds given by Propositions 1 and 2. Since these bounds are not tight, mergers should be attempted even when the predicted error norm exceeds E^{max} as long as it is less than zE^{max} for some *a priori* chosen constant z. If the actual error norm after the merge is unacceptable then the merge is canceled. If z is chosen too small then many mergers will be missed while if z is too large there will be too many cancellations.

V. EXPERIMENTAL RESULTS

A series of tests was conducted using the E_2 norm given by (9).

Figs. 4–7 show a cell outline which was used as a test for various line fitting schemes. Fig. 4 shows an approximation by Ramer's algorithm [8].[1] Fig. 5 shows an approximation using Ramer's algorithm for initial segmentation and fitted lines for improving the fit. Fig. 6 shows an approximation obtained by split-and-merge algorithm. Fig. 7 shows the results of the application of the

[1] According to Duda and Hart [9] this algorithm was first suggested by G. E. Forsen.

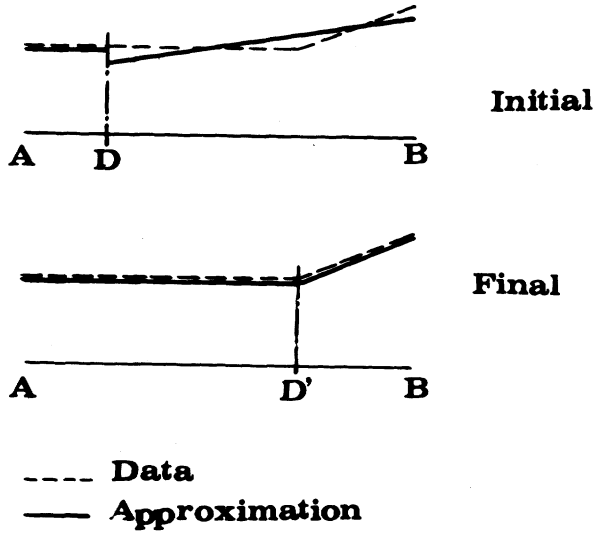


Fig. 3. Worst case of breakpoint adjustment is split-and-merge algorithm.

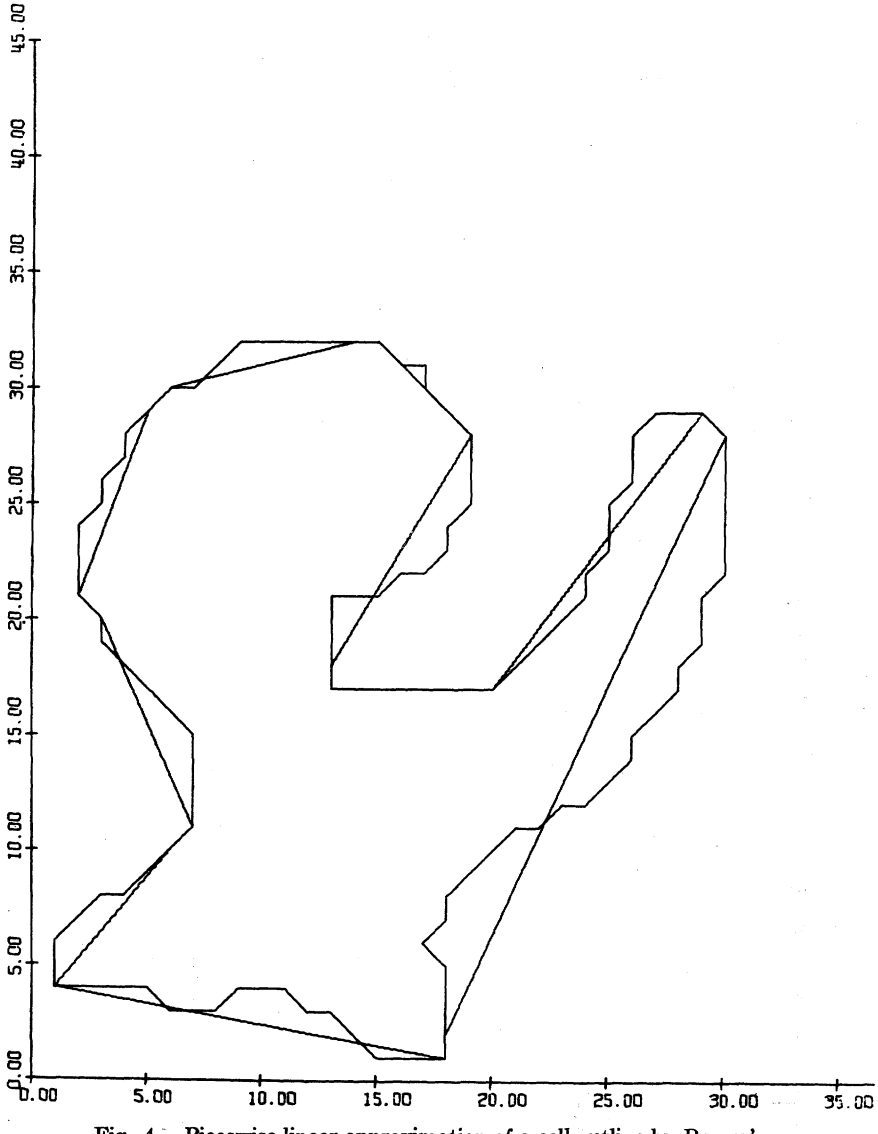


Fig. 4. Piecewise linear approximation of a cell outline by Ramer's algorithm [8].

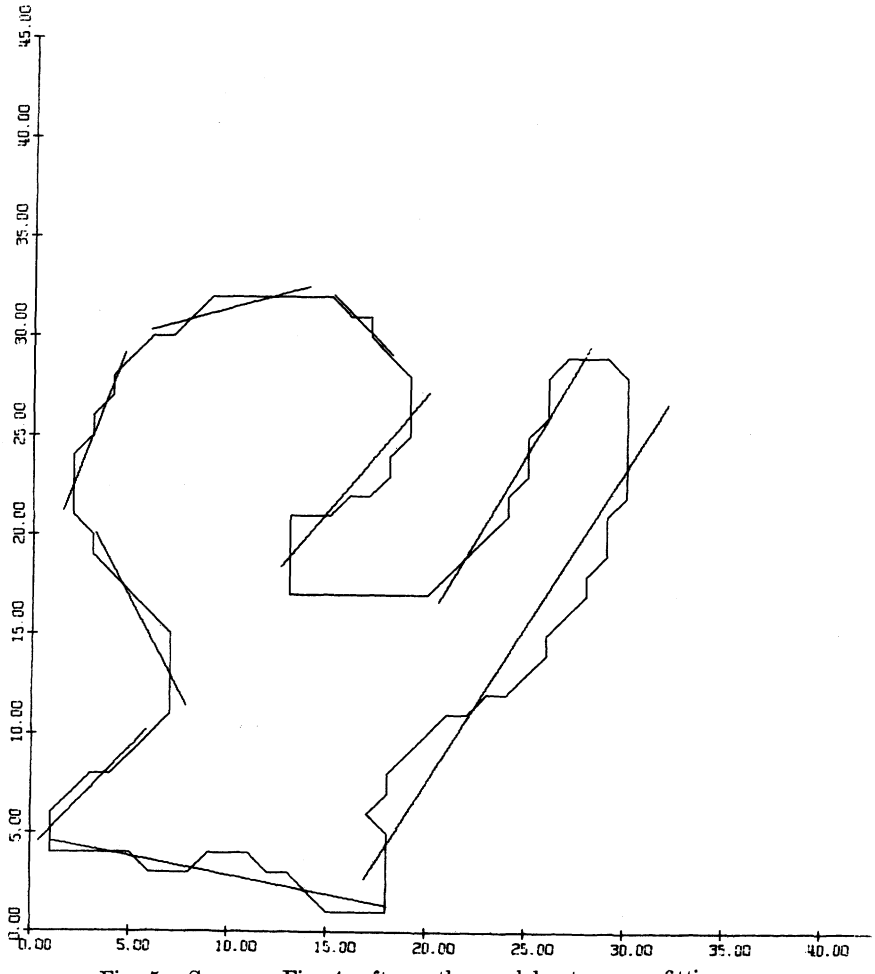


Fig. 5. Same as Fig. 4, after orthogonal least square fitting was used.

algorithm when the segments S_i are not disjoint but share the breakpoints. This results in approximations closer to continuous. Table I summarizes the results. The original outline was digitized into 124 points. In all the runs we had $E^{max} = 30$.

A second series of tests involved a piecewise linear approximation of a square, starting from a poor representation (Fig. 8). The final approximation is expected to give 4 segments with zero error. The results are shown in Table I.

Another investigation was undertaken for the approximation of an EKG. Fig. 9 shows an example. This was digitized into 240 points with an amplitude range of about 100. Split-and-merge runs were made with $E^{max} = 50$. The breakpoint adjustment algorithm used was similar to the one described in [20]. However the direction of the change for the E_2 norm is not defined *a priori* as in the case of the E_∞ norm and thus requires a check in both directions. This point as well as other variations of the algorithm have been presented in detail elsewhere [29]. The results are also summarized in Table I.

Although the CALCOMP plotter has drawn lines between discrete data points the inputs to the algorithm were only such points. Therefore, except in the case of Fig. 7, the approximations are done in disjoint segments of points and they are independent. However an inspection of Figs. 4–7 shows that in most of the breakpoints the approximations are in essence continuous. Exceptions are the points marked A in both figures.

The execution time in all instances was less than a second on an IBM 360-91 using the Fortran G compiler. For more details about time requirements see [29].

Table II illustrates the relative frequency of splits, merges, and endpoint adjustments for the example of Fig. 7. In all instances initialization was uniform with 10 segments. The only variable parameter was E^{max}. It is also of interest to observe a certain invariance among the breakpoints for different runs. These are listed in Table III. When the E^{max} is increased then the breakpoints either stay at the same location (approximately) or disappear. Thus the method tends to locate a "natural" segmentation.

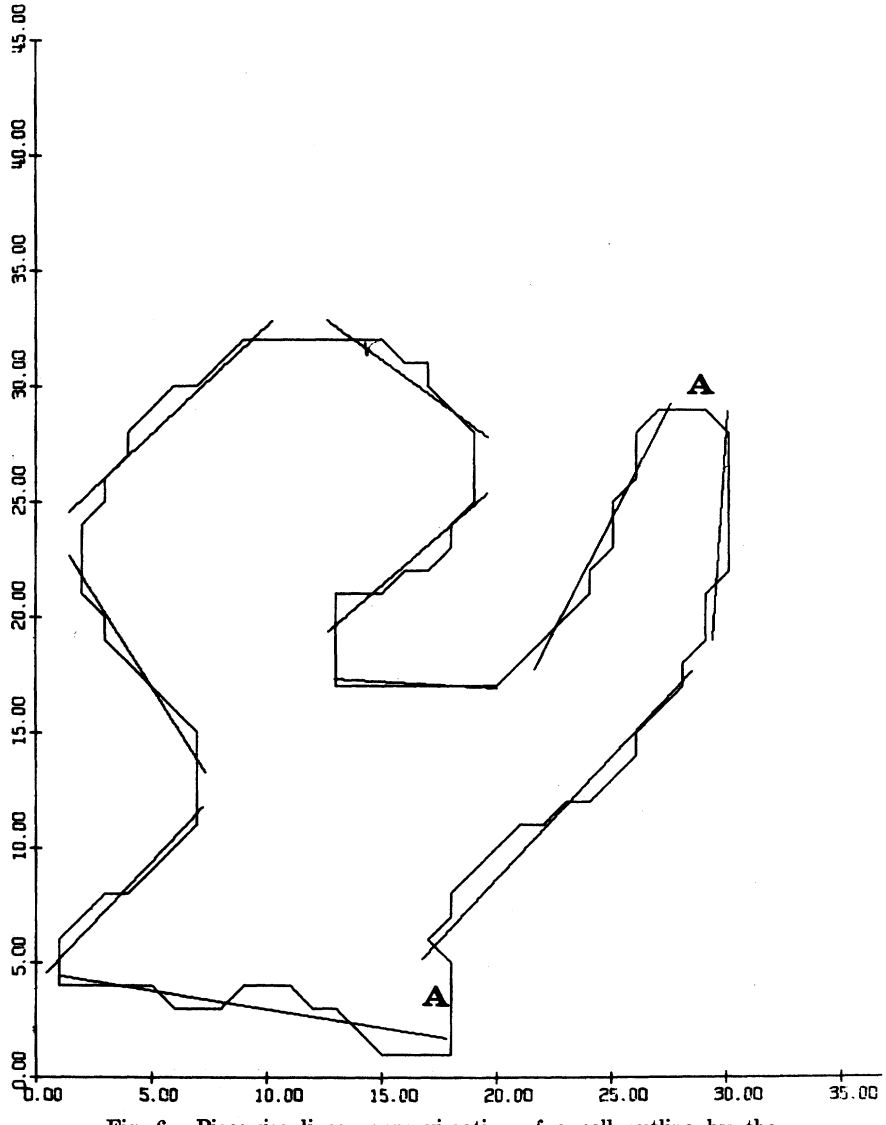


Fig. 6. Piecewise linear approximation of a cell outline by the split-and-merge algorithm with disjoint segments.

VI. CONCLUSIONS

The use of the split-and-merge algorithm removes the need for a careful *a priori* choice of the number of segments as well as the initial segmentation. In addition it results in faster curve fitting than other schemes. The resulting segmentations are not necessarily continuous but this can be easily remedied, if required for certain applications. A polygon can be readily fitted using the breakpoints found as vertices. Applications of this method in picture processing are presented elsewhere [30].

APPENDIX

PROOF OF PROPOSITION 2

Let the approximating function over $S_1 U S_2$ be

$$\sum_{i=1}^{m} \left(\frac{a_i^1 + a_i^2}{2} \right) \varphi_i(t). \tag{A1}$$

Then we have

$$E_2^U \leq \sum_j \left[f(t_j) - \sum_{i=1}^{m} \left(\frac{a_i^1 + a_i^2}{2} \right) \varphi_i(t_j) \right]^2 \tag{A2}$$

where the summation is over $S_1 U S_2$. This sum can be split into terms X_1 and X_2 the first of which is

$$X_1 = \sum_j \left[f(t_j) - \sum_{i=1}^{m} a_i^1 \varphi_i(t_j) + \sum_{i=1}^{m} \left(\frac{a_i^1 - a_i^2}{2} \right) \varphi_i(t_j) \right]^2. \tag{A3}$$

The summation is over S_1. Expanding the expression in brackets and taking into account that E_2 over S_1 is less or equal to E_2^0 we have

$$X_1 \leq E_2^0 + \sum_j \left[\sum_{i=1}^{m} \left(\frac{a_i^1 - a_i^2}{2} \right) \varphi_i(t_j) \right]^2. \tag{A4}$$

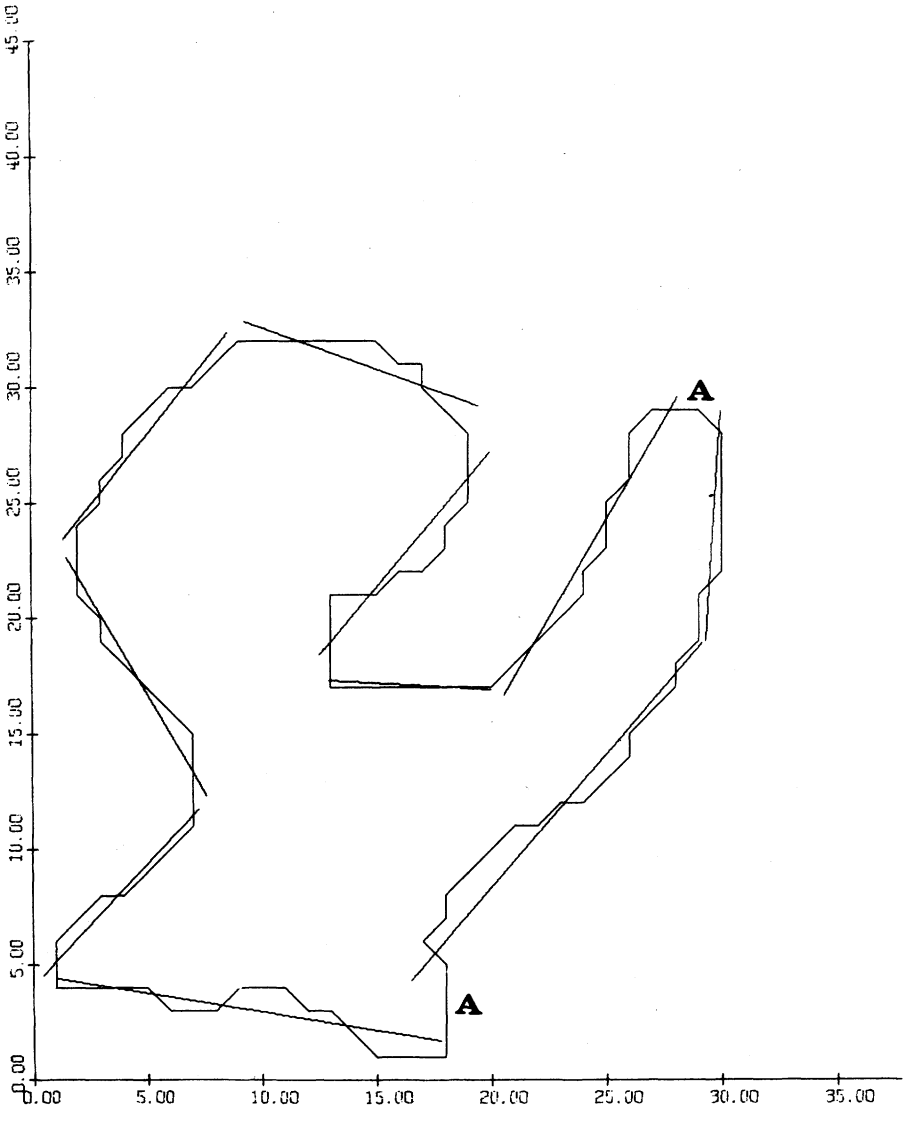


Fig. 7. Same as Fig. 6, but with overlapping segments.

TABLE I
PIECEWISE LINEAR FIT OF A CELL OUTLINE, A SQUARE, AND AN EKG

Number of Segments					Final Values of Error Norm			
Initial	After 1st Iteration	Final	Type of Initialization	Split and Merge	E_2	max E_2^1	Average E_2^1	Item
10	—	10	Ramer	no	63.67	30.44	6.37	Cell Outline
10	11	11	Ramer	yes	34.08	9.82	3.10	
10	11	11	uniform	yes	32.49	7.27	2.95	
31	17	15	uniform	yes	15.46	2.60	1.03	
4	9	9	uniform	yes	78.67	15.23	8.74	
3	6	4	"poor" initial	yes	0.0			Square
5	5	4	choice	yes	0.0			
11	—	11	uniform	no	333.8	96.40	30.35	EKG
10	—	10	Ramer	no	187.9	99.97	18.79	
10	11	11	Ramer	yes	145.8	41.42	13.25	
10	13	12	uniform	yes	151.2	41.49	12.60	
12	15	12	uniform	yes	144.5	41.42	13.14	
5	12	11	uniform	yes	154.8	41.42	14.07	
11	16	12	uniform	yes	118.9	41.42	9.91	
48	11	11	uniform	yes	145.8	41.42	13.25	

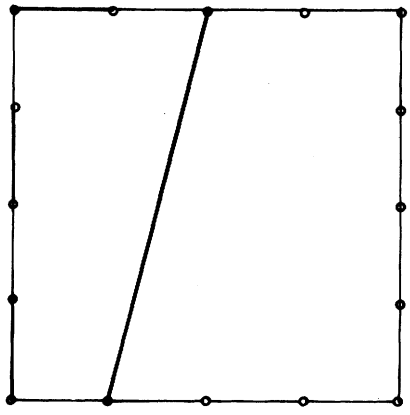


Fig. 8. Initial approximation of a square for testing the split-and-merge algorithm.

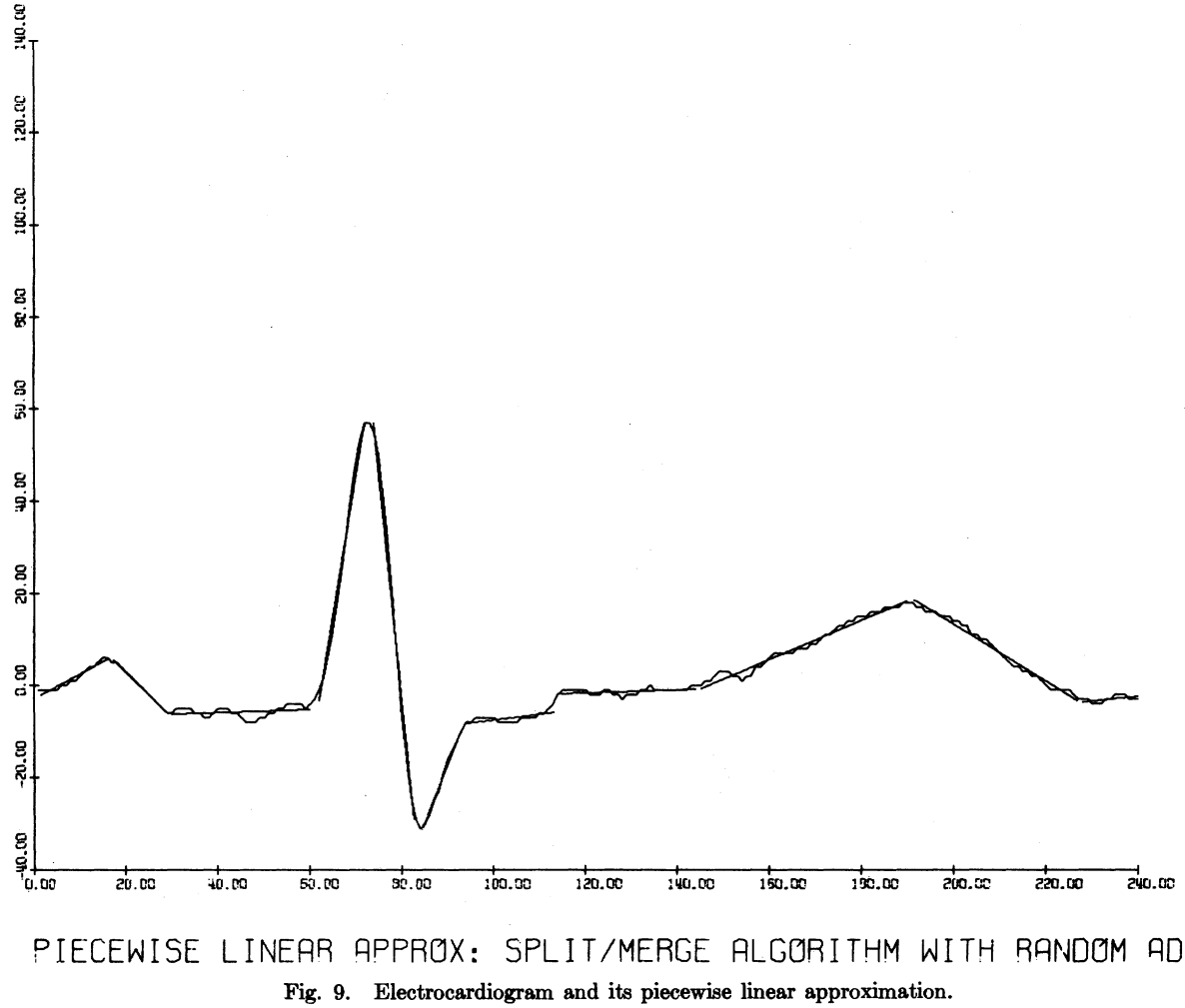


Fig. 9. Electrocardiogram and its piecewise linear approximation.

Using the Schwartz inequality and (13) for the last term we obtain:

$$X_1 \leq E_2^0 + \delta^2 \sum_j \Big[\sum_{i=1}^{m} \varphi_i(t_j)\Big]^2. \tag{A5}$$

If $|\, S_1 \,|$ denotes the number of points in S_1 we have, taking into account (14):

$$X_1 \leq E_2^0 + |\, S_1 \,| \delta^2 L^2. \tag{A6}$$

A similar expression can be obtained for X_2 so that

$$E_2^n \leq 2E_2^0 + (|\, S_1 \,| + |\, S_2 \,|)\delta^2 L^2. \tag{A7}$$

For the linear approximation $\varphi_1(t) = 1$ and $\varphi_2(t) = t$ so that L is approximately equal to the maximum length of S_1 or S_2. Therefore we have

TABLE II
EFFECTS OF E^{max} ON THE PIECEWISE LINEAR FIT OF A CELL OUTLINE

E^{max}	Final n	Number of Iterations	Splits	Mergers	Adjustments	Final Values of Error Norm E_2	min E_2^i	max E_2^i	over E_2^i	Execution Time on 360-91 in Seconds
300	6	7	0	4	19	229	5.8	112	38	0.31
150	8	8	1	3	29	130	0.0	44	16	0.36
120	9	5	1	2	21	88	0.7	44	9.8	0.34
100	9	6	2	3	22	85	0.7	37	9.4	0.35
90	10	6	2	2	19					0.35
80	10	6	2	2	19	54	0.7	14	5.4	0.35
70	10	6	2	2	20					0.36

TABLE III
EFFECTS OF E^{max} ON THE LOCATION OF THE BREAKPOINTS

E^{max}	Final n	Location of Breakpoints											
300	6	15	—	46		66		84		97		—	124
150	8	15	—	46	62		69	84		98		111	124
120	9	15	—	46	62		70	83	94		104	115	124
100	9	20	—	46	62		70	83	94		104	115	124
70, 80, 90	10	20	38	48	62		70	83	94		104	115	124

$$E_2^n \leq 2E_2^0 + 2\delta^2 L^3. \tag{A8}$$

For other polynomial approximations

$$L = \max_t \Big[\sum_{i=0}^{m-1} (t^i)^2\Big]^{1/2} = \max_t \left[\frac{t^{2m}-1}{t^2-1}\right]^{1/2} \approx \max_t t^{m-1}.$$

If K denotes the maximum length of S_1 or S_2 we have

$$L = K^{m-1} \tag{A10}$$

and the above equation becomes

$$E_2^n \leq 2E_2^0 + 2K\delta^2 L^2 = 2E_2^0\delta^2 L^2 + \frac{1}{m-1}. \tag{A11}$$

Note that in deriving (A4) from (A3) there is no need to include the cross product

$$\sum_j \Big[f(t_j) - \sum_{i=1}^{m} a_i^1 \varphi_i(t_j)\Big]\Big[\sum_{k=1}^{m} (a_k^1 - a_k^2)\varphi_k(t_j)\Big] \tag{A12}$$

since optimal E_2 approximations are characterized by [25]

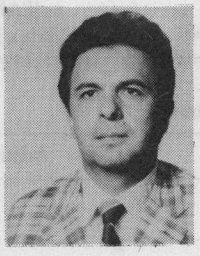
$$\sum_j \Big[f(t_j) - \sum_{i=1}^{m} a_i^1 \varphi_i(t_j)\Big]\varphi_k(t_j) = 0. \tag{A13}$$

(This can be shown by setting equal to zero the partial derivatives with respect to a_i in the equation defining E_2.)

REFERENCES

[1] T. Pavlidis, "Analysis of set patterns," *Pattern Recognition*, vol. 1, pp. 165–178, 1968.
[2] ——, "Linguistic analysis of waveforms," in *Software Engineering*, J. Tou, Ed. New York: Academic, 1971, pp. 203–225.
[3] D. E. McClure, "Feature selection for the analysis of line patterns," Ph.D. dissertation, Div. Appl. Math., Brown Univ., Providence, R. I., 1970.
[4] J. Sklansky, "Recognition of convex blobs," *Pattern Recognition*, vol. 2, pp. 3–10, 1970.
[5] U. Montanari, "A note on minimal length polygonal approximation to a digitized contour," *Commun. Ass. Comput. Mach.* vol. 13, pp. 4–47, 1970.
[6] T. Kaneko and R. Mancini, "Straight line approximation for boundary of left ventricular chamber from a cardiac cineangiogram," in *Proc. 6th Annu. Princeton Conf.*, Princeton, N. J., pp. 337–341, 1972.
[7] T. Pavlidis, "Piecewise approximation of functions of two variables through regions with variable boundaries," in *Proc. of Ass. Comput. Mach. Annu. Conf.*, Aug. 1972, pp. 652–662.
[8] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *J. Comput. Graphics and Image Processing*, vol. 1, pp. 244–256, 1972.
[9] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973, pp. 328–339.
[10] D. E. McClure, "Nonlinear segmented function approximation and analysis of line patterns," Div. Appl. Math., Brown Univ., Providence, R. I., Tech. Rep., 1973.
[11] G. Birkhoff and C. R. DeBoor, "Piecewise polynomial interpolation and approximation," in *Approximation of Functions*, H. L. Garabedian, Ed. Amsterdam, The Netherlands: Elsevier, 1965, pp. 164–190.
[12] J. R. Rice, *The Approximation of Functions*, vol. I, vol. II. Reading, Mass.: Addison-Wesley, 1964, 1969.
[13] C. E. DeBoor and J. R. Rice, "Least square cubic spline approximation: Variable knots," Dep. Comput. Sci., Purdue Univ., Lafayette, Ind., Tech. Rep. 81.
[14] R. E. Esch and W. L. Eastman, "Computational methods for best spline approximation," *J. Approx. Theory*, vol. 2, pp. 85–96, 1969.
[15] D. D. Braess, "Chebyshev approximation by spline functions with free knots," *Numer. Math.*, vol. 17, pp. 357–366, 1971.
[16] T. Pavlidis, "Optimal piecewise polynomial L_2 approximation of functions of one and two variables," Comput. Sci. Lab., Dep. Electr. Eng., Princeton Univ., Princeton, N. J., Tech. Rep. 143, Oct. 1973.
[17] G. M. Phillips, "Algorithms for piecewise straight line approximations," *Comput. J.*, vol. II, pp. 211–212, 1968.
[18] M. G. Cox, "An algorithm for approximating convex functions by means of first degree splines," *Comput. J.*, vol. 14, pp. 272–275, 1971.
[19] A. Cantoni, "Optimal curve fitting with piecewise linear functions," *IEEE Trans. Comput.*, vol. C-20, pp. 59–67, Jan. 1971.
[20] T. Pavlidis, "Waveform segmentation through functional approximation," *IEEE Trans. Comput.*, vol. C-22, pp. 689–697, July 1973.
[21] H. Vogt, *Element de Mathematiques Superieures*, 17th ed., Vuibert, 1947, p. 140.
[22] R. E. Esch and W. L. Eastman, "Computational methods for best approximation," Sperry Rand Res. Cen., Sudbury, Mass., Tech. Rep. SEG-TR-67-30, Dec. 1967.

[23] P. Rabinowitz, "Application of linear programming to numerical analysis," *SIAM Rev.*, vol. 10, pp. 121–159, Apr. 1968.
[24] D. C. Handscomb, Ed., *Methods of Numerical Approximation.* Elmsford, N. Y.: Pergamon, 1966.
[25] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods.* New York: Wiley, 1966.
[26] J. R. Rice, "Running orthogonalization," *J. Approx. Theory*, vol. 4, pp. 332–338, 1971.
[27] C. M. Kortman, "Redundancy reduction—A practical method of data compression," *Proc. IEEE*, vol. 55, pp. 253–263, Mar. 1967.
[28] T. Pavlidis and A. P. Maika "Uniform piecewise polynomial approximation with variable joints," *J. Approx. Theory*, to be published.
[29] T. Pavlidis and S. L. Horowitz "Piecewise approximation of plane curves," in *Proc. 1st Int. Joint Conf. Pattern Recognition*, Washington, D. C., Oct. 30–Nov. 1, 1973, pp. 396–405.
[30] H. Y. Feng and T. Pavlidis, "The generation of polygonal outlines of objects from gray level pictures," Comput. Sci. Lab., Dep. Elec. Eng., Princeton Univ., Princeton, N. J., Tech. Rep. 150, Apr. 1974.



Theodosios Pavlidis (S'62–M'64) was born in Salonica, Greece, on September 8, 1934. He received the diploma from the National Technical University of Athens, Athens, Greece, in 1957, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1962 and 1964, respectively.

Since 1964 he has been with the Department of Electrical Engineering, Princeton University, Princeton, N. J., and he is presently an Associate Professor. He is a consultant with the U. S. Army (Frankford Arsenal). His research interests include pattern recognition, image processing, approximation theory, and the application of mathematical techniques in the life sciences. He is the author of *Biological Oscillators: Their Mathematical Analysis* (New York: Academic, 1973).

Dr. Pavlidis is a member of the Association for Computing Machinery, the Pattern Recognition Society, and Sigma Xi.

Steven L. Horowitz was born in New York City, N. Y., on March 4, 1953. He received the B.S. degree in electrical engineering from Princeton University, Princeton, N. J., in 1974.

His present plans include pursuing research in picture processing while attending graduate school in Computer Science at Princeton University.

Correspondence

On the Modeling of Demand Paging Algorithms by Finite Automata

C. C. YANG

***Abstract*—Two finite automata are devised for modeling two classes of demand paging algorithms. The first one of one input and three outputs models the class of algorithms with a constant amount of allocated space. The second one of one input and six outputs models the class of algorithms with a variable amount of allocated space. Some evaluation techniques are developed following each model. The memory states of the first class algorithm with the Least Recently Used (LRU) replacement policy and the working set model of the second class are recursively defined by strings of the loaded pages. The adopted replacement policy and the state string updating procedure are imbedded in the recursive definition of memory states. Properties of some algorithms are developed to fit the finiteness assumption of a reference string.**

***Index Terms*—Demand paging algorithm, finite automaton, Least Recently Used replacement policy, memory management, missing-page rate, page-fault rate, page-success rate, sequential machine, working set model.**

INTRODUCTION

In a virtual storage computer system [1], the memory allocation problem includes three aspects [2]–[3]: 1) page fetching or loading, 2) page placement and 3) page replacement. Placement is irrelevant in a paging environment [3]. Loading is on demand only in demand paging system. Therefore, replacement is all that is left. In addition, besides the page size and the number of allocated page frames, the replacement policy is one of the most important parameters [1] in designing a virtual storage computer system. Aho *et al.* [4] discovered that the minimum cost is always achieved by a demand paging algorithm under usual assumptions about memory system organization. Thus, a systematic study and a unified approach of demand paging algorithms should deserve considerable interest and attention.

Although analytic studies of the algorithms just mentioned have established some important formal models like the working set model [5], the stack algorithm [6], the independent reference model [4], the probabilistic model based on a finite-state first-order Markov chain [7] and the formal treatment of the anomalous behavior of the algorithm with the FIFO (First-In First-Out) replacement policy [8], the well established automata theory [9] did not fruitfully serve in this area until the appearance of Gelenbe's recent article [10] in which he uses a stochastic automaton to formalize the class of the random partially preloaded algorithms.

This correspondence presents two finite automata that formalize two classes of demand paging algorithms: one involves constant allocated space like the algorithm with the LRU (Least-Recently Used) replacement policy and the other involves variable allocated

Manuscript received September 13, 1973; revised December 13, 1973. This work was supported in part by IBM Taiwan Corporation, IBM World Trade Corporation, IBM San Jose Research Laboratory, and by N.I.H. Grant 5-501-RR05300-12 at the University of Alabama in Birmingham, Birmingham, Ala.

The author is with the Department of Information Sciences, University of Alabama in Birmingham, University Station, Birmingham, Ala. 35294.