Signal Processing 5 (1983) 221–227
North-Holland Publishing Company

A FAST PIECEWISE LINEAR APPROXIMATION ALGORITHM

F. GRITZALI and G. PAPAKONSTANTINOU

N.R.C. "Democritos", Department of Computers, Aghia Paraskevi, Athens, Greece

Received 8 February 1982
Revised 8 June 1982

Abstract. A piecewise linear approximation algorithm is suggested in this paper. The algorithm is fast since it works in one pass through the given data and it does not require much memory space. Moreover the peak points are preserved and become corners of the resulting piecewise linear curve. Hence the algorithm is suitable for real time applications in which the peak points are carrying important information and therefore should be calculated accurately.

Zusammenfassung. In diesem Beitrag wird ein Algorithmus zur Approximation von Kurvenverläufen durch Geradenstücke vorgeschlagen. Der Algorithmus ist schnell und benötigt wenig Speicherplatz; er arbeitet die Eingangsdaten in einem einzigen Verarbeitungsschritt ab. Hierbei bleiben die Extremwerte des Kurvenverlaufs erhalten; sie werden die Eckpunkte der sich ergebenden stückweise linearen Näherung. Der Algorithmus ist daher günstig für solche Echtzeitanwendungen, bei denen die Extremwerte wichtige Informationsträger sind und daher möglichst genau berechnet werden sollten.

Résumé. Un algorithme d'approximation linéaire par segment est suggéré. L'algorithme est rapide, car il fonctionne en un seul passage sur les données et il ne nécessite pas beaucoup de place mémoire. De plus, les pointes des pics sont conservées et deviennent les coins de la courbe, linéaire par segments, qui découle. Ainsi, l'algorithme est approprié pour des applications en temps réel où les pointes des pics transportent de l'information importante et, par conséquent, doivent être calculées avec précision.

Keywords. Digital curves, piecewise linear approximation, peak points, data compression.

1. Introduction

The problem of the approximation of functions or sampled data by piecewise linear functions appears in a number of fields e.g. encoding of information, feature generation etc.

The methods used for this purpose are usually based on heuristic algorithms, hence they usually require much time. In the cases of real time applications with fast response times this may be prohibitive.

Tomek [1] has described a fast algorithm which finds the piecewise linear approximation in one pass through the given data. Sklansky and Gonzalez [2] give another fast 'scan-along' algorithm, named RMPP, which may be applied also to digitized boundaries of images in 2-space.

These two algorithms have the disadvantage that the peak points are not accurately represented by the approximating line segments. In reference [2] an alternative version of the RMPP algorithm, named RMPP/LS, is suggested in which longer allowable segments are selected at the expense of a longer perimeter for the output curve in order to detect local and global peaks. But to get a good approximation of the peaks the error bound ε must be kept small which results in a large number of segments.

In this paper the algorithm of reference [2] is modified and combined with another algorithm which recognizes the peaks, so that the piecewise linear approximation takes place in between the peaks. This last algorithm is based on the one of reference [3]. The peak points are precisely

0165-1684/83/$03.00 © 1983 Elsevier Science Publishers

detected and included in the resulting set of end points of line segments. The precise detection of the peak points and their incorporation in the piecewise linear approximation of the waveform is important in some applications like ECG waveforms analysis.

The algorithm has been tested with real ECG waveforms and the results obtained are better than the results obtained with the RMPP or RMPP/LS algorithm.

2. Description of the algorithm

A waveform is defined as a sequence of the coordinates of its sample points (x_1, y_1), (x_2, y_2), $\ldots, (x_n, y_n)$, such that $x_i, y_i \geqslant 0$ for $1 \leqslant i \leqslant n$.

The algorithm proposed here is a combination of two algorithms, one for finding the peaks in a waveform and one for the piecewise linear approximation of the waveform (PLAW). The last one is basically the one of reference [2], properly modified so that instead of the Euclidean distance the Δy difference is used, which proved more suitable for *ECG* waveforms. The basic idea of this last algorithm is shown in Fig. 1.

Starting with the point (x_1, y_1) of the waveform two angles v_1 and l_1 with initial values $+\frac{1}{2}\pi$ and $-\frac{1}{2}\pi$, respectively, are defined. The algorithm proceeds stepwise. For each sample point (x_i, y_i) two lines are drawn defined by the points (x_1, y_1), $(x_i, y_i + \varepsilon_1)$ and (x_1, y_1), $(x_i, y_i - \varepsilon_1)$, respectively, were ε_1 is a small predetermined value. These lines form with the x-axis the angles a_i, b_i as shown in Fig. 1. For each point (x_i, y_i) we define the angles v_i and l_i so that

$$\begin{aligned} v_i &:= \text{minimum of } v_{i-1}, a_i \\ l_i &:= \text{maximum of } l_{i-1}, b_i. \end{aligned} \tag{1}$$

If $v_i > l_i$ and the point (x_i, y_i) is inside its current cone, i.e. the area defined by the lines $[(x_1, y_1), \tan v_i]$ and $[(x_1, y_1), \tan l_i]$, then the point is a valid one. The point (x_i, y_i) in Fig. 1 is a valid one, while the point x_{i+1}, y_{i+1} is not since it is not inside its current cone. If a point (x_j, y_j) is not a valid one, then the line between the points (x_1, y_1) and (x_{j-1}, y_{j-1}) is the PLAW between these two points, (x_{j-1}, y_{j-1}) becomes the starting point of the next line segment with $v_{j-1} := +\frac{1}{2}\pi$ and $l_{j-1} := -\frac{1}{2}\pi$.

The algorithm guaranties that all points (x_i, y_i), $1 \leqslant i \leqslant n$, of the waveform will have a Δy difference from the line segment approximating the waveform such that $\Delta y \leqslant \varepsilon_1$. The polygonal approximation of the RMPP algorithm is not necessarily within the (error bound) ε of the original data set. Figure 4 in the Sklansky–

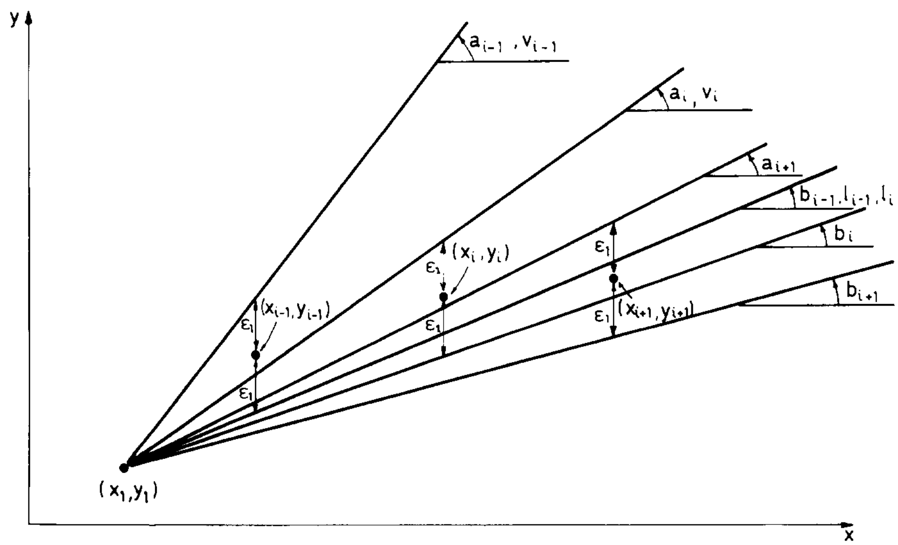

Fig. 1. An example of a PLAW.

Gonzalez paper [2] is a counter-example of their own claim. This can also be seen in Fig. 3(b) of this paper.

We shall next describe informally the algorithm for detecting peaks of the waveform.

An upgoing portion of the waveform is defined as a portion with positive slopes. A downgoing portion is defined as a portion with negative slopes. A noisy portion after an upgoing portion is defined as a portion with y_i coordinates such that $y_j - \varepsilon_2 \leqslant y_i \leqslant y_j$, where y_j is the y coordinate of the end point of the previous upgoing portion and ε_2 is a predetermined value approximately equal to the peak noise amplitude. A noisy portion after a downgoing portion is defined as a portion with y_i coordinates such that $y_j + \varepsilon_2 \geqslant y_i \geqslant y_j$, where y_j is the y coordinate of the end point of the previous downgoing portion.

The algorithm finds the different kinds of portions of the waveform and marks the following points as peaks:

(a) the end points of those upgoing portions which are followed by downgoing portions, ignoring possible noisy portions, and

(b) the end points of those downgoing portions which are followed by upgoing portions, ignoring possible noisy portions.

We shall now describe the proposed combination of these two algorithms using the model of a finite state automaton.

The algorithm works as follows: for each sample point a PLAW, started at a peak point is continued until another peak point is found or until a possible peak point is found. In the latter case another PLAW starts which is ignored if the possible peak point was not really a peak point but which becomes the continuation of the previous PLAW if the possible peak point was really a peak point.

The increment at the sample point (x_i, y_i) is defined as $\Delta y_i = y_i - y_{i-1}$ for $1 < i \leqslant n$. We classify the increments into five categories, namely:

small positive	(sp)	for	$0 < \Delta y_i \leqslant \varepsilon_2$,
large positive	(lp)	for	$\varepsilon_2 < \Delta y_i$,
small negative	(sn)	for	$-\varepsilon_2 \leqslant \Delta y_i < 0$,
large negative	(ln)	for	$\Delta y_i < -\varepsilon_2$,
zero	(z)	for	$\Delta y_i = 0$.

The states of the automaton are:

1. for an upgoing portion,
2. for a noisy portion after an upgoing one,
3. for a downgoing portion,
4. for a noisy portion after a downgoing one.

The transition diagram of this automaton is given in Fig. 2. The input to the automaton consists of the increments sp, lp, sn, ln, z, and the output consists of the semantic routines r_1, r_2, r_3, r_4, r_5, which are described later.

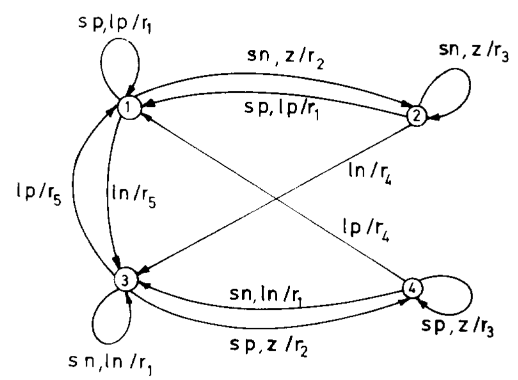

Fig. 2. Transition diagram of the finite state automaton.

Let k be the index for the quantities of the kth line segment, with $k \geqslant 0$. The zeroth line segment is an auxiliary one, not belonging to the resulting piecewise curve. Let (X_k, Y_k) be the starting point of the kth line segment, and let v_i^k, l_i^k, a_i^k, b_i^k be the angles as defined in Fig. 1 for the sample point (x_i, y_i) and with (X_k, Y_k) the starting point of the line segment.

The following procedures described in an Algol-like notation will be used in the semantic routines r_1, r_2, r_3, r_4 and r_5:

(a) Initialize(x_i, y_i, k): for the initialization of the kth line segment.

$$X_k := x_i, \quad Y_k := y_i, \quad v_i^k := \tfrac{1}{2}\pi, \quad l_i^k := -\tfrac{1}{2}\pi.$$

(b) Continue(x_i, y_i, k): for the continuation of the PLAW.

v_i^k := minimum of v_{i-1}^k, a_i^k,

l_i^k := maximum of l_{i-1}^k, b_i^k,

if (x_i, y_i) is not a valid point of line segment k **then begin** $k := k+1$, $X_k := x_{i-1}$, $Y_k := y_{i-1}$,

$$v_i^k := a_i^k, \quad l_i^k := b_i^k, \textbf{ end}.$$

(c) Peakreg(k): for the inclusion of a peak point as a corner of the resulting piecewise line curve.

if $X_k < X_0$ **then** $k := k+1$,

$X_k := X_0$, $Y_k := Y_0$, $v_{i-1}^k := v_{i-1}^0$,

$l_{i-1}^k := l_{i-1}^0$.

The initialization of the automaton is done by setting the initial state to 1, the index k to 1, the index i to 2, the X_1, Y_1 to x_1, y_1, respectively, and by calling initialize (x_1, y_1, k).

The semantic routines r_1, r_2, r_3, r_4 and r_5 can now easily be described as follows:

r_1: continue(x_i, y_i, k), $i := i+1$
(for the continuation the kth line segment).

r_2: continue(x_i, y_i, k), initialize $(x_{i-1}, y_{i-1}, 0)$, continue(x_i, y_i, 0), $i := i+1$
(for the continuation of the kth line segment and the start of an auxiliary line segment with 0 index).

r_3: continue(x_i, y_i, k), continue(x_i, y_i; 0), $y_i := y_{i-1}$, $i := i+1$
(for the continuation of both the kth and the 0th line segments if (X_0, Y_0) is a possible peak point).

r_4: peakreg(k), continue(x_i, y_i, k), $i := i+1$
(for the inclusion of the peak point (X_0, Y_0) as a corner of the resulting piecewise line curve and the continuation of the next line segment).

r_5:$k := k+1$, initialize(x_{i-1}, y_{i-1}, k), continue (x_i, y_i, k), $i := i+1$
(for the inclusion of the peak point (X_{i-1}, Y_{i-1}) as a corner of the resulting piecewise line curve and the continuation of the next line segment).

We have assumed that $\varepsilon_1 \geqslant \varepsilon_2$, hence the auxiliary PLAW which starts at a possible peak point has only one segment, and the line index k may be increased only by 1, while the automaton is in the state 2 or 4.

The quantity ε_2 must be small enough so that the peaks will not be distorted and at the same time it must be large enough so that small peaks due to noise will be eliminated.

The proposed algorithm guaranties that each point of the waveform will have a Δy difference from the resulting piecewise linear curve (pointwise error) smaller or equal to ε_1. Moreover the peak points of the waveform will be preserved in the resulting piecewise line curve.

3. Experimental results

Four algorithms namely A1, A2, A3 and A4, were implemented in FORTRAN and were tested using real ECGs. A1 was the RMPP algorithm of reference [2], algorithm A2 was the altered version RMPP/LS of reference [2], and algorithm A3 was the one proposed in this paper. Algorithm A4 was the one-pass algorithm (algorithm 1) of reference [1].

The ECGs data were taken from the CSE Project Library [4].

The same set of data was processed using the four algorithms. If the number of resulting segments was the same for A1, A2 and A3, the results obtained with A3 were much better than those of A1 and A2. In particular, A1 and A2 lead to distortions when applied to QRS waves where the slopes are very big and change abruptly. In order to have a good representation of QRS waves, a data reduction of 3 to 4 was achieved with algorithms A1 and A2, while a reduction of 20 was achieved with algorithm A3.

In Fig. 3 an illustrative ECG example (in arbitrary units) is given. Figure 3(a) shows the original ECG waveform sampled for 4 seconds with a sampling rate of 500 samples/second. Fig. 3(b), 3(c) and 3(d) show the piecewise linear approximation of the original waveform with the algorithms A1, A2 and A3 respectively. The number of the resulting line segments were 190, 215 and 95 respectively. As we can see from Fig. 3, algorithm A2 gives better results than algorithm A1 at the expense of larger number of line segments, while algorithm A3 gives better results and with less line segments.

Both algorithms A3 and A4 guarantee that the pointwise error is smaller than a predetermined value ε_1. Figs. 3(d) and 3(e) show the piecewise linear approximation of the original waveform with the algorithms A3 and A4 respectively, for the same value ε_1. As we can see from Fig. 3 the results are better with algorithm A3. Moreover, algorithm A3 finds the prominent peaks (i.e. those not due to noise) without distortion at all, while

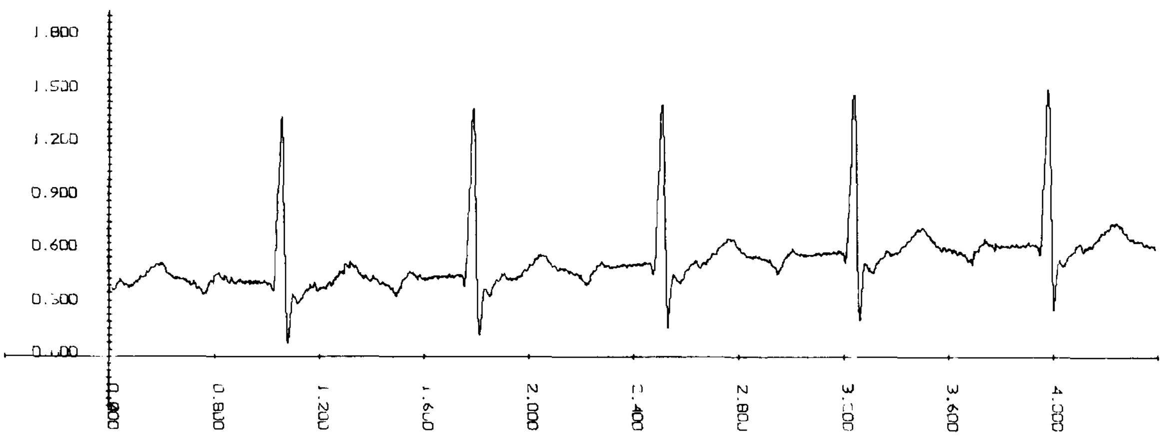

Fig. 3(a). An illustrative ECG waveform.

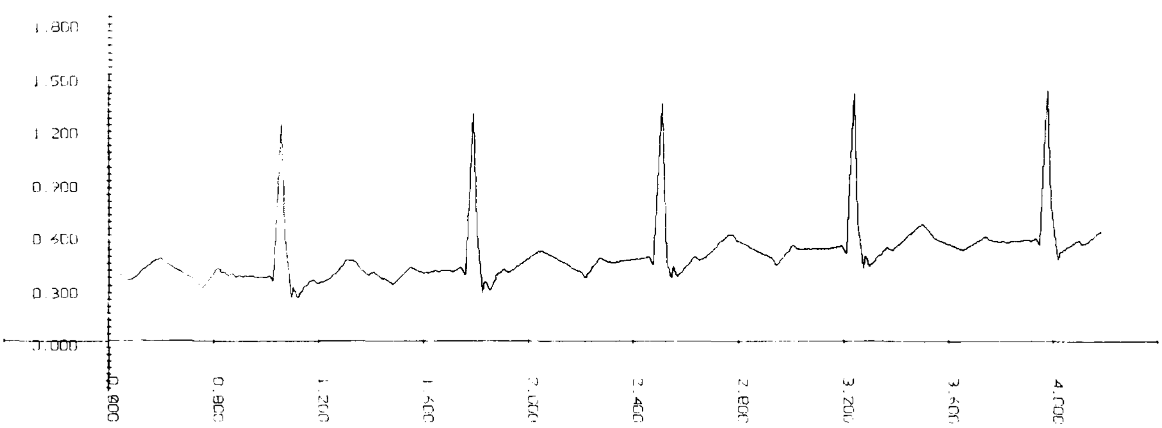

Fig. 3(b). The ECG waveform processed with algorithm A1.

Fig. 3(c). The ECG waveform processed with algorithm A2.

Fig. 3(d). The ECG waveform processed with algorithm A3.

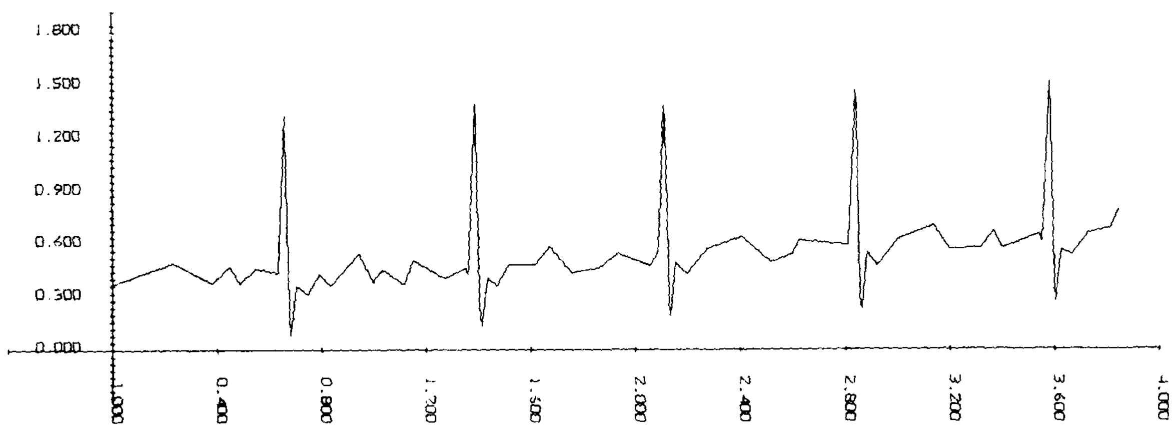

Fig. 3(e). The ECG waveform processed with algorithm A4.

algorithm A4 finds them with amplitude distortion (about 10%) and phase distortion (about 6 sample points). The above distortion is due to the fact that the endpoints of the line segments computed by algorithm A4, are not in general points of the given digitized curve. Finally, the 'percent rms difference' as defined in reference [5] is smaller with algorithm A3 (3% for A3 and 6% for A4).

The number of the resulting line segments for the same ε_1, is less with algorithm A4 (95 for A3 and 80 for A4). When different values of ε_1 are used such that the number of the resulting line segments is about the same for both algorithms, the corresponding piecewise linear approximations are visually comparable. But still algorithm A4 distorts the peaks and gives larger percent rms differences.

The computational effort depends linearly on the amount of data, i.e. is $O(n)$. The memory space required is constant independent of the data, i.e. $O(1)$. In this sense, the algorithm is fast and has low storage requirements. In practice, it is well suited for real-time applications.

Other piecewise linear approximating methods can also be utilized with the model proposed here provided they are 'scan-along' methods, like those of references [1], [2] and [3].

References

[1] I. Tomek, "Two algorithms for piecewise linear continuous approximations of functions of one variable", *IEEE Trans. Comput.*, Vol. C-23, 1974, pp. 445–448.

[2] J. Sklansky and V. Gonzalez, "Fast polygonal approximation of digitized curves", *Pattern Recognition*, Vol. 12, 1980, pp. 327–331.

[3] G. Papakonstantinou and E. Skordalakis, "An algorithm for recognizing characteristic points in waveforms", *Informatica*, Vol. 78, 6/103, 1978.

[4] J.L. Willems *et al.*, "Common standards for quantitative electrocardiography: The CSE pilot study", in: F. Cremy et al., eds., *Proc. Medical Informatics Europe 81*, Springer-Verlag, Berlin, 1981, pp. 319–326.

[5] J.P. Abenstein and W.J. Tompkins, "A new data-reduction algorithm for real-time ECG analysis", *IEEE Trans. Biomedical Engineering*, Vol. BME-29, 1982, pp. 43-48.