MULTIPLE DYNAMIC MATCHING AND MEDIAN FILTERS APPLIED TO SONIC WELL LOGS

by

W. SCOTT POWELL LEANEY

B.Sc.(Geophysics), The University of Manitoba, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Department of Geophysics and Astronomy

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

November 1986

Department of Geophysics and Astronomy

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date Oct. 31, 1986

ABSTRACT

Nonlinear signal matching is a generalization of cross correlation in that a discrete lag between signals is replaced with a variable lag function or 'matching function', $m(x)$. Two methods are reviewed which attempt to solve for $m(x)$, namely the dynamic programming approach and the inverse theory approach. Both methods suffer from pitfalls and require the input of prior constraints to ensure convergence to the correct solution. The goal of this work has been to develop a method that can handle simple or complex matching problems and can succeed without any prior knowledge constraints. The multiple dynamic matching method is the result. It uses a significance threshold to extract a set of ridge points from a similarity matrix and applies dynamic programming to obtain a set of matched sections. These significant matched sections or 'subpaths' are then combined into a set of complete matching functions and a 'mean local confidence' norm is evaluated to determine the best overall match.

It is argued through a model of change between signals, that given that the correct matching function is known, the presence of large amplitude changes between signals can cause the correct matching function to appear suboptimal under similarity norms. Multiple dynamic matching, because it generates suboptimal solutions as well, does not overlook the correct matching function. Typically the top three interpretations as ranked by mean local confidence will contain the expert's choice for the correct matching function.

The use of median filters to preprocess the data and enhance well log features for matching has been investigated. A new 'median decomposition' is discussed as well, and in the context of a scale - space point of view to filtering, it is argued that median scale space is the proper choice for blocky waveforms. Finally, the connection between multiple dynamic matching and pattern recognition is explored, and matching iteratively through scale is proposed as a means of further generalizing the multiple dynamic matching method, making efficient high resolution matching possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

The problem of nonlinear signal correlation and the dynamic programming approach to solve it was first communicated to me by Toshifumi Matsuoka and I am very thankful to him for that. Throughout this work Ken Whittall has made countless fruitful suggestions and has helped to keep my goal in perspective. Thanks Ken. My supervisor Dr. Tad Ulrych has been a great source of inspiration and has enabled me to laugh at myself when things seemed impossible. Thanks to him for his special brand of encouragement. I would also like to thank the entire department of Geophysics and Astronomy and all my friends there.

Special thanks also to my parents for supporting my education.

Chapter 1

INTRODUCTION

1.1 Correlation and Matching Functions

The general problem of correlating two or more signals occurs in many fields of study. Examples of correlation problems come from geology (stratigraphic correlation), geophysics (seismic signal correlation), and speech processing (word recognition). Any time that sequences of data have been recorded in an experiment and a connection between signals is suspected, a correlation problem exists. Obtaining optimum signal alignment requires determining the samples of the measured response which correlate or match in two or more sequences of data. Conventionally this has been solved using a correlation or similarity functional where the signals or portions of the signals are lagged relative to one another. The unnormalized cross correlation or similarity between a reference signal $r(x)$; $x = 0, X$ and a test signal $t(x)$; $x = 0, Y$ may be written as

$$S_\xi = \int_0^X r(x)t(x+\xi)dx, \tag{1.1}$$

where S_ξ is the global similarity at lag ξ, (and the limits of integration may be changed to a window (X_1, X_2) for a local similarity measure). Finding the optimum lag simply involves computing S_ξ over a range of ξ's and determining the maximum. However, some data such as well log and seismic data may undergo variable moveout so that a continuously varying lag is necessary to recover optimum signal alignment. Thus the lag ξ is replaced with a continuous lag function or 'matching function', $m(x)$. The generalized similarity functional may now be written as

$$S = \int_0^X r(x)t[m(x)]dx. \tag{1.2}$$

where $m(x)$ determines the shift, stretches and compressions of the test signal.

One may want to determine the 'matching function' for a variety of reasons. The connection of equivalent events may be wanted for an automatic interpretation (for example connecting equivalent time horizons in well logs), or one may want to actually apply the matching function to align the data before additional processing. Another application of matching functions is to assign a waveform or portion of it to one of several classes as defined by their expected waveforms. Here one is looking for the simplest matching function to determine, for example, whether a seismic signal is due to an earthquake or nuclear blast. The application explored in this project is that of connecting structurally equivalent portions of digital sonic well logs. All applications of matching functions require an accurate and reliable method of determining the function itself, and if objectivity is required so that no *a priori* knowledge is included in the solution, the problem is particularily difficult.

1.2 Waveform Change and the Matching Problem

A simple mathematical model of change between waveforms is helpful to further explain the problem of correlation or matching. The change between a reference signal $r(x)$ and a test signal $t(x)$ may be modelled as due to a structural change $m(x)$ and an amplitude distortion $a(x)$, so that:

$$r(x) = t[m(x)] + a(x). \tag{1.3}$$

It is desired to solve for $m(x)$ and $a(x)$ and thereby quantify completely the change between signals. The trouble is that $a(x)$ can only be determined once $m(x)$ is known, and as $m(x)$ cannot be separated from the test signal, its determination is a nonlinear problem. Also, the matching function may be simple or complex, smooth or discontinuous, and there is another complication as well. For most matching problems the amplitude distortion $a(x)$ is small so that nearly identical matching is possible if only $m(x)$ is recovered. However, the presence of $|a(x)| >> 0$ for any x can cause the result of applying the correct matching function to appear suboptimal, so that any method that solves for $m(x)$ by maximizing the coherence between $r(x)$ and $t[m(x)]$ would be mislead, determining an incorrect structural change. In these severe cases it would be beneficial to generate suboptimal solutions as well.

The determination of the matching function is thus a complicated issue. The problem is nonlinear, the solution may be arbitrarily nonlinear, and amplitude distortions may cause the correct solution to appear suboptimal. One method that has been successful in the solution of nonlinear problems is that of dynamic programming (Bellman and Dreyfus 1962). However, only recently has dynamic programming been applied to matching problems in speech processing (Rabiner, Rosenberg, and Levinson 1978, Myers and Rabiner 1981), stratigraphic correlation (Gordon and Reyment 1979, Smith and Waterman 1980, Clark 1985) and seismology (Liu and Fu 1982). Dynamic programming was reviewed as a tool for geophysical data processing by Anderson and Gaby (1983). They discuss the method of dynamic waveform matching and its geophysical applications, and discuss some of the difficulties of the method with certain matching problems such as well to well correlation.

Another method which is totally distinct from dynamic matching is the inverse approach to signal correlation (Martinson, Menke and Stoffa 1982). Both methods of waveform matching and their pitfalls will be reviewed in the next two sections, followed by an introduction to multiple dynamic matching and the well log matching problem in particular.

1.3 Dynamic Waveform Matching

Nonlinear matching shifts and warps (stretches and compresses) one of the signal axes (time or depth) until the corresponding features of the two waveforms are aligned. The optimum shift and warping of the signal axis, the matching function, may be determined by dynamic programming. The following is a brief review of this approach to waveform matching based largely on a paper by Anderson and Gaby (1983).

Since it is desired to find the shift and warping that will minimize the difference between a reference waveform r_i ; $i = 1, n$ and a test waveform t_j ; $j = 1, m$, a difference or distance measure may be computed between pairs of points to form a dissimilarity matrix, d_{ij}:

$$d_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{m} |r_i - t_j|. \tag{1.4}$$

Then the path through d_{ij} which minimizes

$$D = \sum_{i=1}^{n} d_{i,p(i)} \tag{1.5}$$

is defined as the optimum path and its coordinates $i, p(i)$ determine the shift and warping of the test waveform axis for optimum waveform alignment. The matching problem is therefore viewed as a pathfinding problem through the matrix d_{ij}. Dynamic programming can efficiently determine the optimum path whenever the dynamic programming principle applies:

Whenever the path from a starting point S to an intermediate point I does not influence the optimum choice of paths for travelling from S to a goal point G, then the minimum distance from S to G is the sum of the minimum distance from S to I and the minimum distance from I to G. (Anderson and Gaby (1983)).

Dynamic programming is efficient because it assumes that the optimum local path is part of the optimum global path. In other words, only a small portion of d_{ij} local to the present path point needs to be computed to determine the next path point.

To minimize equation (1.5) with dynamic programming several things must be specified:

a) endpoint constraints - where the path may begin and end.

b) local continuity constraints - constraints on the path from one point to the next.

c) global path constraints - limitations on where the path can wander in (i, j) space.

A graphical representation of contraints a and c is shown in Figure 1.1a. In most applications of dynamic waveform matching, the starting and end points are fixed, equivalent to knowing and constraining the correlation of points near the beginning

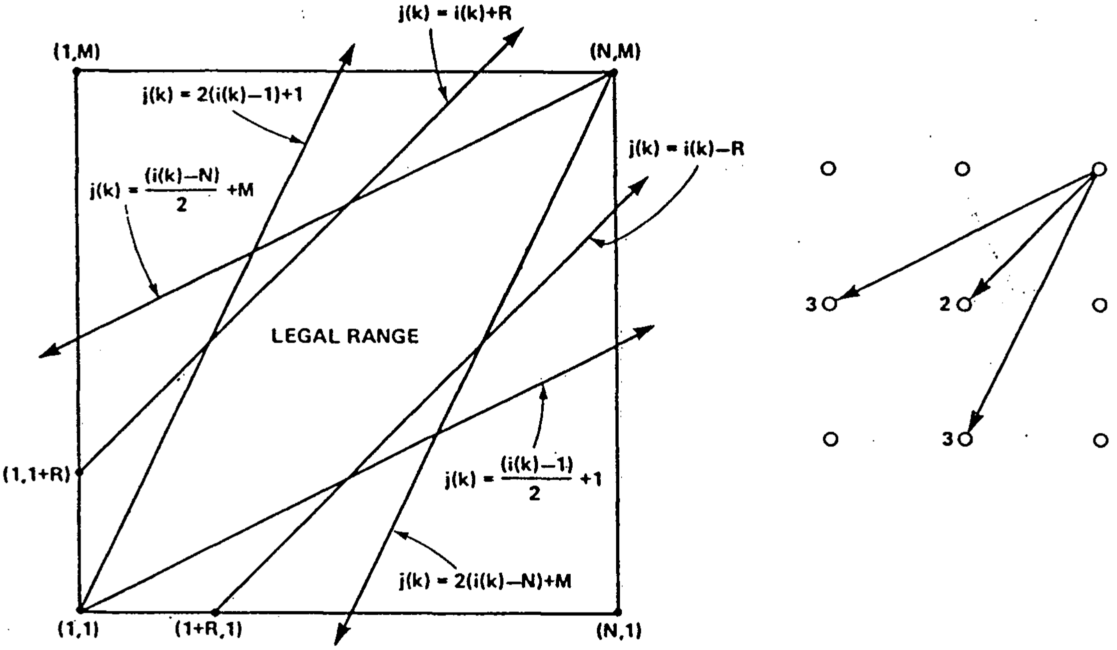

Figure 1.1 (a) Endpoint and global path constraints, (b) type IId local path constraint, (from Anderson and Gaby (1983)). End points (1,1) and (N,M) correspond to already interpreted matches between signals, and the sloped lines beginning from them are the result of the local path constraint in (b) (pathfinding may proceed from right to left or from left to right). Potential path points are commonly weighted as in (b), further restricting the simplicity of the path.

and end of the test and reference waveforms. Local constraints on the next possible path point can take the form of bounds on proximity, slope, or slope weighting, wherein a potential path point d_{ij} may be weighted proportional to the resulting deviation of the local slope from one. The commonly used type IId local path constraint is shown in Figure 1.1b. The path constraints are equivalent to restricting the overall simplicity of the function, and along with the dynamic programming principle, make dynamic waveform matching an efficient method for solving for the matching function. However, dynamic matching often fails to recover the correct match when waveforms

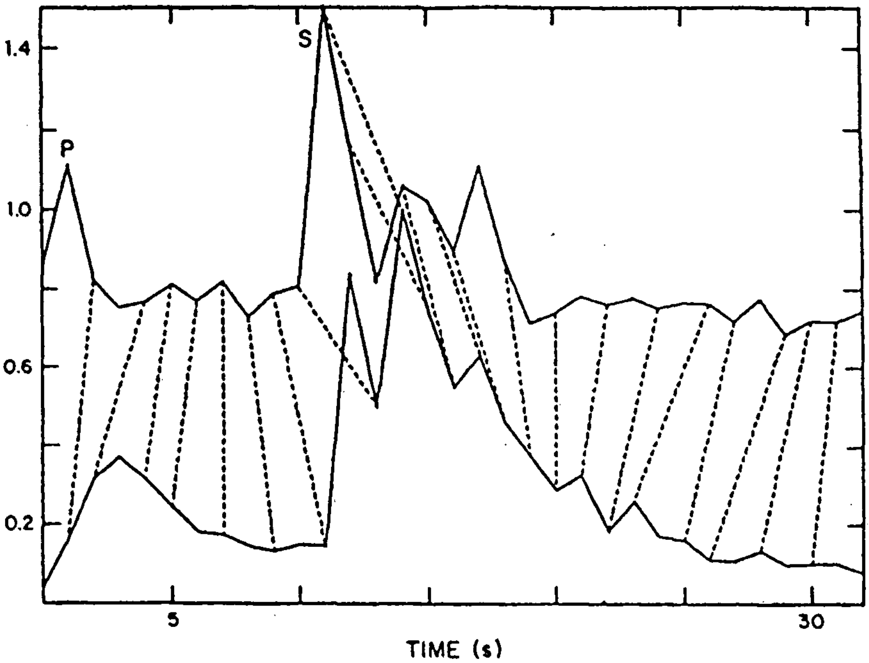

Figure 1.2 The matching of envelopes of local earthquake data using equations (1.4) and (1.5), and the path constraints of Figure 1. (From Anderson and Gaby (1983)). Dashed lines indicate matches. Notice the poor correlation of major events. The local path constraints and measure of difference d_{ij} are too restrictive to recover the correct match.

show high variability in moveout and signature. This is because sections of waveforms which have no events of significant magnitude will produce an unclear match and poorly defined minima in d_{ij}, and this can mislead a pathfinding algorithm that looks ahead only locally. Also, pronounced moveouts or disappearances of significant events can require the pathfinding algorithm to look farther ahead than the prescribed local path constraint. Figure 1.2 shows the result of applying dynamic waveform matching using equations (1.4) and (1.5), and the path constraints of Figures 1.1a and 1.1b to match the envelopes of local earthquake data. Notice the poor correlation of primary and secondary events. This is due to an inadequate measure of dissimilarity (or similarity) and local path constraints which are too restrictive. The problems with the dynamic matching approach have been overcome by using more than one metric to compare the signals (Liu and Fu 1982), and by the inclusion of user supplied constraints (Gordon and Reyment 1979, Clarke 1985), where the user may constrain the path to include points representing already interpreted matches (the correlation of 'marker' events for instance). Obviously any method that requires user interpretation is not objective or amenable to the processing of large volumes of data. What is needed is a method that is not susceptible to being mislead by insignificant or unclear matches, can handle large and variable moveout, and can succeed without any user supplied interpretation.

1.4 The Inverse Approach to Signal Correlation

The inverse theory approach to signal correlation avoids some of the pitfalls of the dynamic programming approach and can yield stable solutions to relatively simple matching problems. Martinson, Menke and Stoffa (1982) suppose that a reference signal $r(t)$ is related to its distorted version $d(x)$ through the matching function $x(t)$. Thus $r(t) = d[x(t)]$. They then parameterize $x(t)$ as the sum of a linear trend and a truncated Fourier series:

$$x(t) = a_0 t + \sum_{i=1}^{n-1} a_i \sin(i\pi t/T) \tag{1.6}$$

where T is the length of $r(t)$.

It is desired to maximize the coherence between $r(t)$ and $d[x(t)]$ which is given as:

$$C = \frac{\int_0^T r(t)d[x(t)]dt}{\int_0^T r^2(t)dt \int_0^T d^2[x(t)]dt} \tag{1.7}$$

When the coherence is maximized the gradient of C with respect to the parameterization coefficients a_i is zero:

$$\frac{\partial C}{\partial a_i} = \nabla \mathrm{C} = 0. \tag{1.8}$$

A complicated expression for the partial deivatives may be derived from (1.7). The numerical values for the coefficients are then found by starting with an initial guess for the solution a_i^0 and iterating to improve the coherence by a small amount ΔC. Using Taylor's theorem ΔC is expanded as:

$$\Delta C = \frac{\partial C}{\partial a_0}\Delta a_0 + \frac{\partial C}{\partial a_1}\Delta a_1 + ... + \frac{\partial C}{\partial a_n}\Delta a_n = \nabla \mathrm{C} \cdot \Delta \mathrm{a}. \tag{1.9}$$

This equation and $n-1$ other constraint equations which come from requiring that the increment Δa always be in the direction given by ∇C give a system of equations which when solved yield the change to the coefficients Δa. Iteration stops when C ceases to increase.

The method is applied to O^{18} profiles from two ocean drill cores to determine differential sedimentation rates. As with all nonlinear inverse methods, convergence to the correct solution is strongly dependent on the initial guess or starting function. Martinson *et al.* obtain different results given a hand picked starting function and a simple linear starting function (see Figure 1.3).

Two points should be made with regard to the inverse method of signal correlation. First, since global coherency is maximized and perturbations exist over the entire length of the matching function, local weaknesses in similarity do not greatly affect the final result and the method is more stable than dynamic matching. Secondly however, in difficult problems where the correct matching function is quite nonlinear, the inverse approach has difficulty in converging to the correct solution without a user supplied

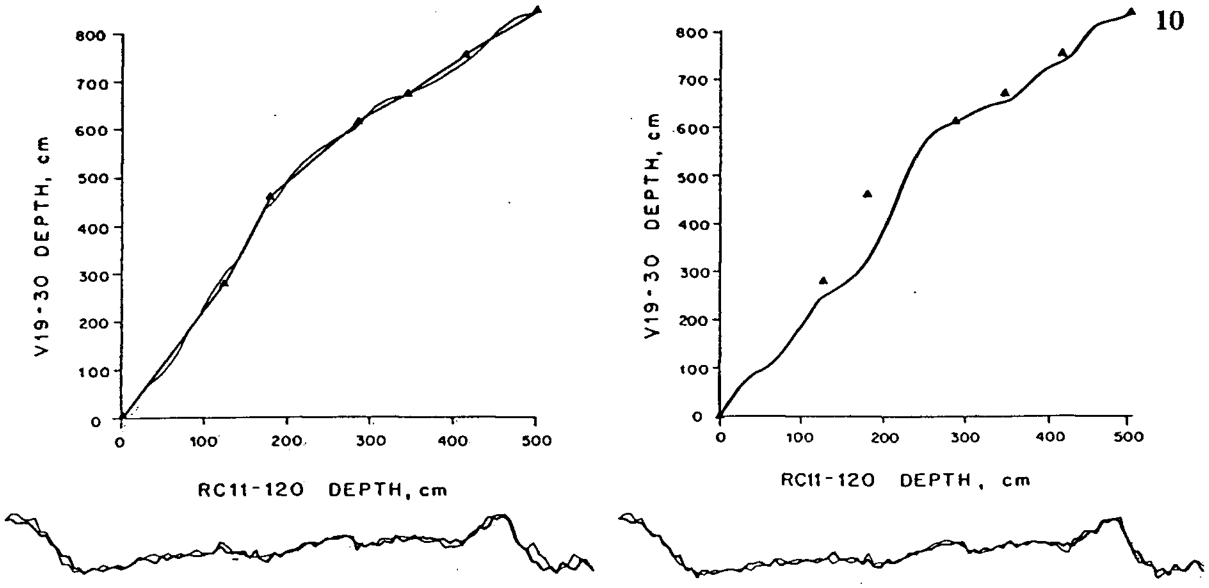

Figure 1.3 Matching function determination with the inverse approach to signal correlation applied to O^{18} data from two ocean drill cores. Solutions using (a) a hand-picked starting function (triangles) and (b) a linear starting function. (From Martinson *et al.* (1982)). Notice the sensitivity of the final solution to the starting function.

starting function. Therefore, as with dynamic waveform matching, the inverse approach when applied to the most general class of matching problems, lacks objectivity and is not suitable for automatic data matching.

1.5 Multiple Dynamic Matching

An approach to overcoming the problems of the previous two methods just discussed may now be clarified. Difficult matching problems could be reliably solved with the inverse approach if a starting function that is simple or complex as necessary could be determined automatically. Similarily, if local low confidence problems of the dynamic matching method in the presence of insignificant or unclear matches could be overcome, and local path constraints relaxed to allow for more complex matching functions, it too would satisfy the criteria of objectivity and reliability. The presence of large amplitude distortions ($|a(x)| >> 0$) between signals which may cause the correct matching function to appear suboptimal could also be handled by either method if more than one starting function was used in the inverse approach, and if additional suboptimal solutions were found with a reliable dynamic matching algorithm.

The previous two methods differ essentially in their definitions of optimality. The inverse theory approach maximizes a coherency functional over the full length of the signals and so uses 'global' optimality. Dynamic waveform matching minimizes the immediate difference between points or windows of points in the signals, and so uses 'local' optimality. It can be said that local optimality gives efficiency while global optimality gives stability. Combining the benefits of both methods would likely give superior results. Experiments with both matching schemes (Leaney 1985, 1986) have indicated the following features for a more robust waveform matching method:

1. To avoid insignificant and unclear matches, associate a significance threshold with local optimality to extract sections of the matching function where confidence is high, and use linear approximations where confidence is low.
2. Quantify final optimality over the entire function with a 'global' norm.
3. To avoid missing the correct solution due to $|a(x)| >> 0$, find suboptimal solutions as well as the optimal solution.

The multiple dynamic matching method contains these features, and when applied to the problem of digital well log correlation which is known to be a difficult matching problem (Robinson 1978, Gordon and Reyment 1979, Schwarzacher 1982, Anderson and Gaby 1983), has produced good results. The method itself will be explained in detail in Chapter 3, but prior to that, a discussion of some of the peculiarities of matching well logs is needed.

1.6 The Well Log Matching Problem

The problem of matching digital well logs poses problems not encountered with other data sets. Well logs are generally separated by kilometers or tens of kilometers and structural geology can change appreciably in that distance, meaning that $m(x)$ can be considerably nonlinear. As well, the signature within a given formation may change due to differences in porosity, density, matrix composition, fluid content etc., so that amplitude distortions $a(x)$ are large. The difference between two well logs may be expected to show any one of, or combination of five classes of digital response change. The following table summarizes the geologic changes and the corresponding digital response changes.

	geologic change	digital response change
1.	dipping or faulting	shift
2.	thinning/thickening	compression/stretching
3.	pinchout	disappearance
4.	faulting/folding	repetition/inverted repetition
5.	compositional change	signature change

Table 1. Summary of geologic and digital response changes between two well logs.

The matching function $m(x)$ determines the first four geologic changes, that is, the structural change between wells, while compositional changes internal to a formation or amplitude distortions $a(x)$, may only be estimated given the correct matching function.

Figure 1.4 shows examples of matching functions corresponding to the four classes of structural change. Conventional cross correlation can handle only class 1 structural change while nonlinear matching methods can handle classes 1 - 3 of structural change. Class 4 structural changes are much more difficult to recover, but it will be seen that a simple modification to multiple dynamic matching would allow for the recovery of this class of structural change.

Another problem is associated with well logs. Since they are generally not smooth signals in that they cannot be accurately represented by a linear combination of continuous functions, conventional filtering techniques cannot be applied with an expectation of undistorted waveform features. Signal matching problems usually require some preprocessing of the data, and since well logs are commonly signals composed of blocks and spikes, an alternative filtering technique that can remove short impulses without smearing block edges is needed. Median filtering has been found appropriate for the filtering of sonic well logs. The data used in this project are five offshore sonic well logs with a sampling interval of 3 meters with each value being the weighted average of 10 readings spaced .3 meters apart. The next chapter deals with the theory of median filtering and its application to sonic well logs.

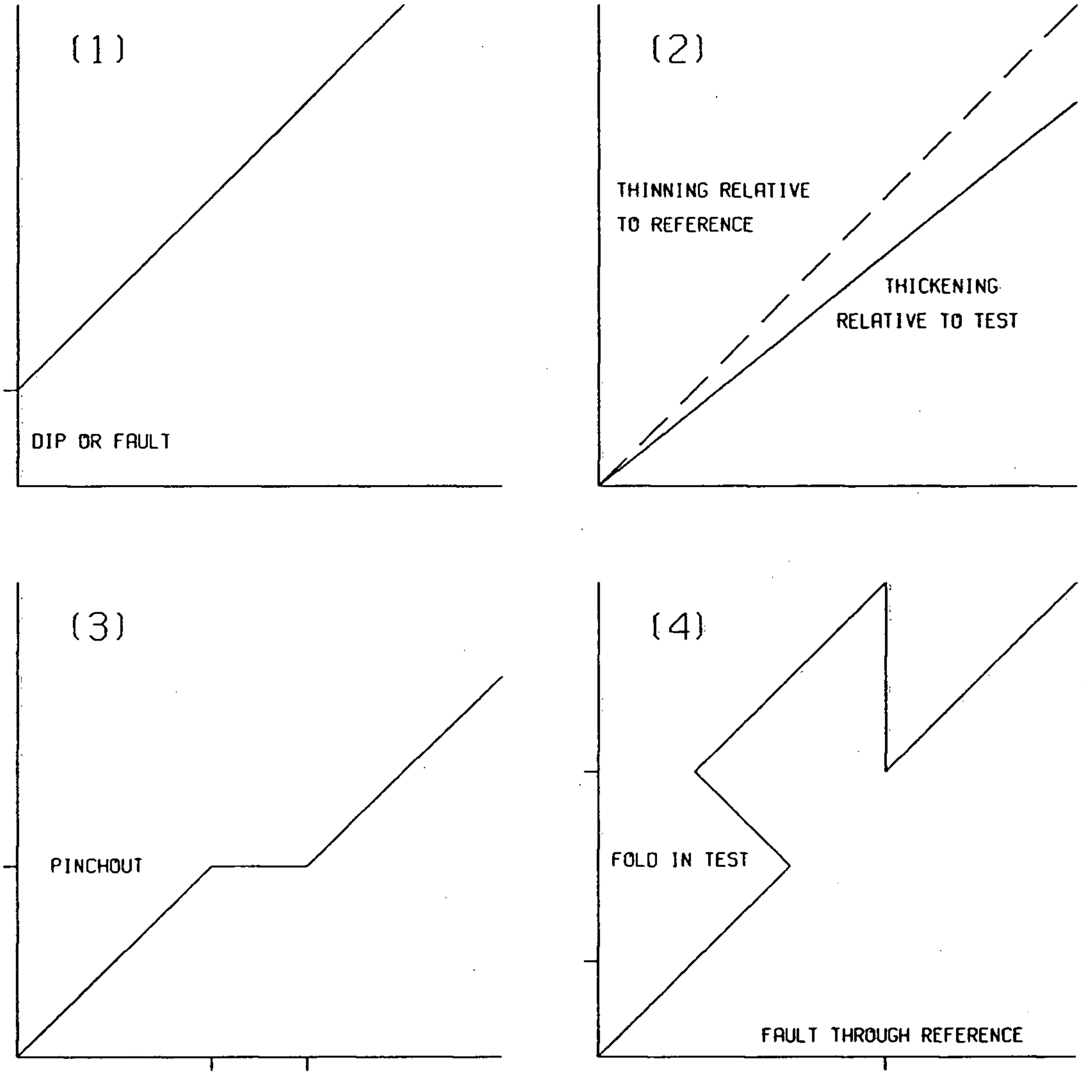

Figure 1.4 Examples of matching functions for four classes of structural change and their analogous geological situations summarized in Table 1. The reference signal is along the x axis and the test signal is along the y axis. (1) Shift (recoverable by conventional cross correlation), (2) compression or stretching, (3) appearance or disappearance, (4) inverted repetition and repetition. Nonlinear matching methods can recover classes 1 - 3 of structural change. Multiple dynamic matching can also handle class 4 structural change.

Chapter 2

PROCESSING SONIC WELL LOG DATA WITH MEDIAN FILTERS

2.1 Introduction to Median Filtering

Median filters operate by replacing the value at the middle of a sequence of numbers with the middle value of the sequence after sorting. The median of a sequence of numbers may be defined mathematically as that value that is a minimal 'distance' from all other values (Claerbout and Muir, 1973). That is, the median value minimizes the sum of the absolute differences between it and the other points of the sequence and so operates as an 'L_1' estimate of the expected value of the sequence. This is known to be a better estimate than the mean when the sequence contains wildly erratic values. The median operator M is also data dependent and is a nonlinear operator as it does not conform to the definition of linearity: $F(a\mathbf{x} + \mathbf{b}) = aF(\mathbf{x}) + F(\mathbf{b})$. For example take $\mathbf{x} = (1, 2, 3)$ and $\mathbf{b} = (2, 3, 4)$,and $a = 2$. Then $[M(a\mathbf{x} + \mathbf{b}) = 8] \neq [aM(\mathbf{x}) + M(\mathbf{b}) = 7]$ and so the median operator is nonlinear. These are properties that define the median, but of particular interest is the median as a filter, moving or sliding along a sequence of numbers.

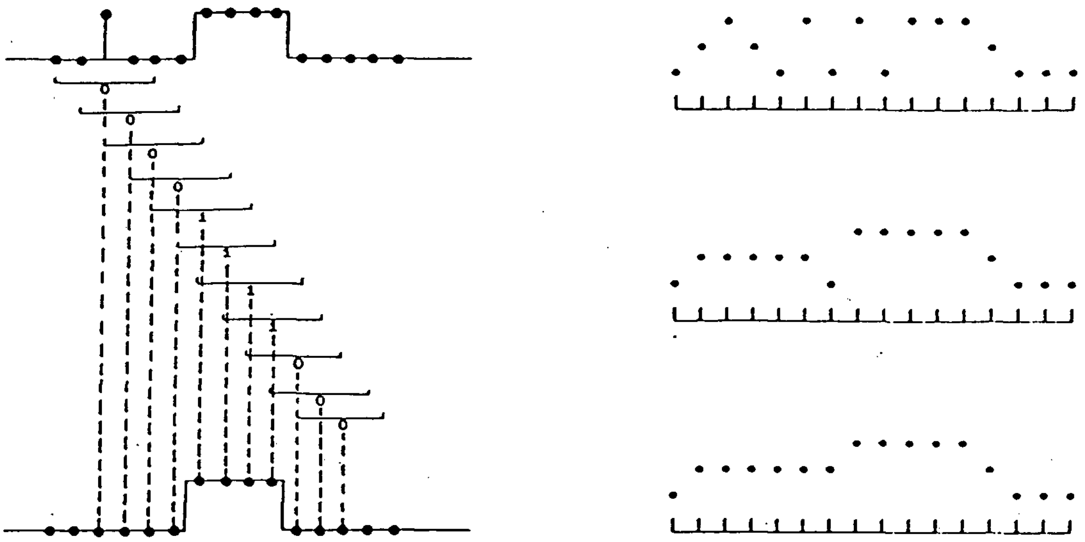

Figure 2.1 (a) Three point median filter applied to a binary signal (from Stewart (1986)), (b) convergence of a binary signal to its $N = 1$ root (from Fitch, Coyle and Gallagher (1984)).

The operation of sort - replace - slide is presented in Figure 2.1a for a three point median filter length applied to a binary signal. Notice the spike rejection and 'block pass' result. Generally any impulse that is less than $N+1$ points long where $2N+1$ is the filter length, will be rejected. However, on complex waveforms where sequence values change at each point, the ideal transfer 'behavior' of passing blocks greater than N in length is not seen. In real data examples the same length filter must be applied successively and even then the ideal transfer behavior is not observed. Gallagher and Wise (1981) have performed a theoretical analysis of the properties of median filters. They derive necessary and sufficient conditions for a signal to be invariant under successive median filtering. These conditions state that the signal must be locally monotone to pass through a median filter unchanged. A signal that is invariant under further median filtering is called a root signal, and it is shown that a root signal may contain only 'constant neighborhoods' and 'edges'. These two possible structures of root signals are defined for a median filter of length $2N + 1$ as:

1) A constant neighborhood is at least $N + 1$ identically valued points such that the constant neighborhood and the edge together are monotone.

2) An edge is a monotone region between two constant neighborhoods of different value and the connecting monotonic region cannot contain any constant neighborhood.

Two other important characteristics are noted here. For a root signal to contain both regions of increase and decrease, the points of increase and decrease must be separated by a constant neighborhood (at least $N + 1$ points of constant value). Also, a signal of length T will converge to its root signal in at most $(T - 2)/2$ successive filterings. Figure 2.1b shows the convergence of a binary signal to its root for a median filter of length 3.

2.2 Median Filtering Applied to Well Logs

The block and edge pass property of median filtering is a desired property for the filtering of sonic well logs. Well logs are composed of blocks and impulses, and it is sometimes desired to remove the impulses while preserving the blocks and edges. Bednar (1983) has investigated the editing of acoustic impedance data with a Markov algorithm, running means and median filters and concluded that median filters produced superior results. Figure 2.2 shows the convergence of a portion of a sonic well log to its root signal for a median filter length of 25. The root signal contains only constant neighborhoods or blocks of greater than 12 samples, and edges or monotonic regions between them. It is noted that an edge may be as little as one sample long. The degree to which median filtering does not preserve perfect steps in the original data,

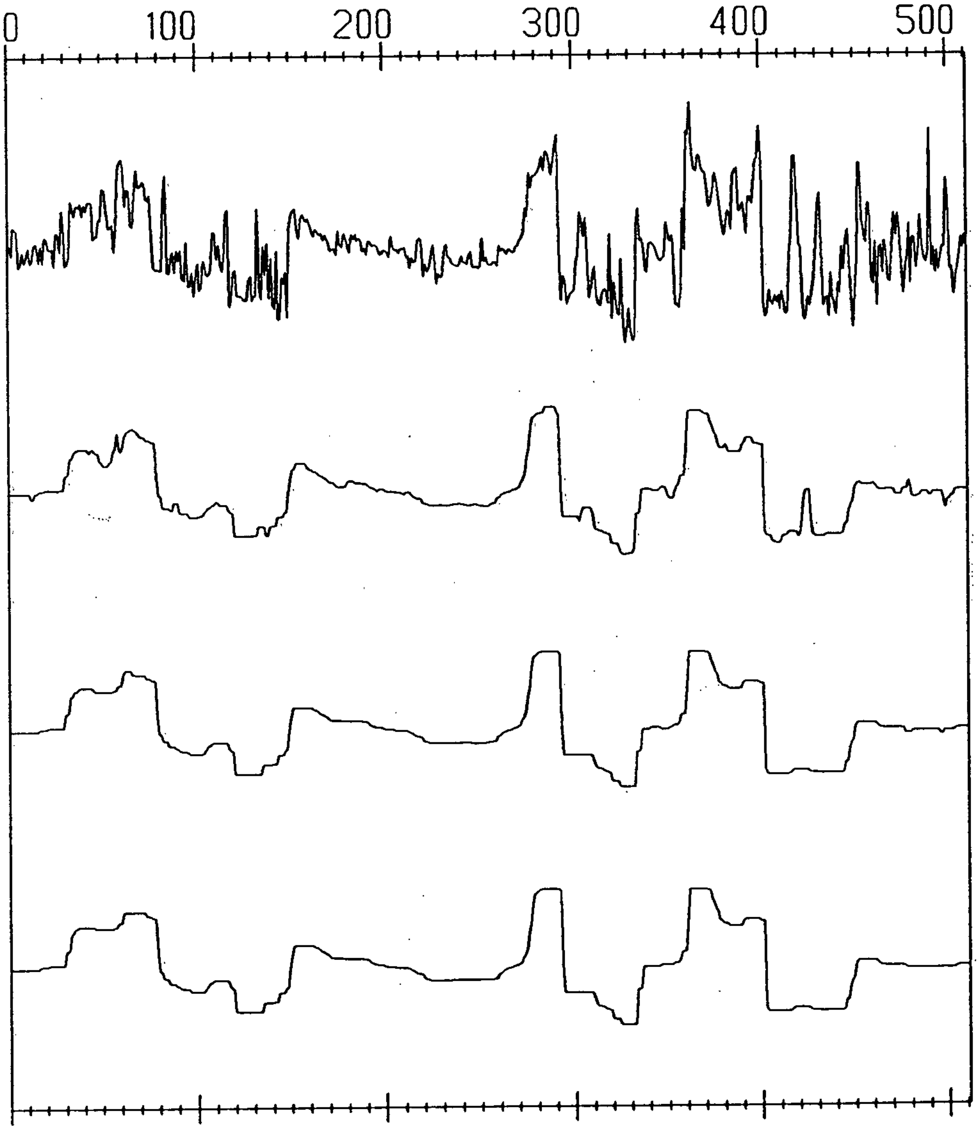

Figure 2.2 Convergence of a portion of a sonic well log for a median filter of length 25.

and produce only blocks, is dependent on the distribution of values near an edge. Evans (1982) points out that nearby spikes may be drawn into a step (making it a series of steps), or, if values are widely distributed near the step, the corner may be rounded. Median filtering when applied to sonic logs may not preserve block edges ideally due to the erratic nature of the data, but it is superior to other filtering techniques.

2.3 Compound Median Filtering

Median filtering for the removal of short impulses and the preservation of blocks and edges can be seen in Figure 2.2. Notice however the creation of multiple steps and rounded corners on some blocks. As mentioned previously, Evans (1982) attributes these artifacts to the distribution of values near the steps. It would seem that the removal of erratic values before the application of a median filter would help to reduce the multiple steps and rounded corners. An innovation to median filtering not found in the literature is employed here to remove these artifacts and is called 'compound median filtering'. Compound median filtering operates by removing all impulses of length $< N$ before application of a $2N + 1$ length median filter. This is accomplished by median filtering the input signal with a filter of length $(2n + 1)$ to its $n = 1$ root signal, then filtering this result to its $n = 2$ root signal and so on until $n = N$.

Gallagher and Wise (1981) state that the root of a signal depends on the median filter window length. For example, the root for a window of length 5 is always a root for a window of length 3, but the root for a window of length 3 is not necessarily a root for a length 5 window. Compound median filtering uses this fact and would only apply a filter of length 5 to the root signal of a length 3 filter. The result is a different root signal for the same length filter, and since any impulse of length $< N$ has been removed, multiple step and rounded corner artifacts should be reduced. Figure 2.3 shows the same original log portion as in Figure 2.2, the root signal for an $N = 12$ median filter and the root signal for the corresponding compound median filter. Also shown is the Walsh low-pass version of the original log. The use of Walsh transforms in the representation of well log data has been explored by Lanning and Johnson (1983). Walsh transforms map the data onto an orthogonal set of square wave functions. Each function is described by its 'zero crossings per second' or 'zps' (also called 'sequency'), and reconstruction to any minimum block length is then possible by zeroing energy in sequencies corresponding to blocks below a desired length and inverse Walsh transforming. The Walsh low-pass version in Figure 2.3 has blocks of length N, the same as the median and compound median filtered logs.

By computing the root mean squared difference or standard error between each of the filtering results in Figure 2.3 and the original log, the best representation of the original log at the given block length may be determined. The table below shows the standard error (SE) for each result.

Filter type	SE
median	191.1
compound median	174.8
Walsh low-pass	215.7

Table 2. Standard error between the original log and median, compound median and Walsh low-pass versions of the original.

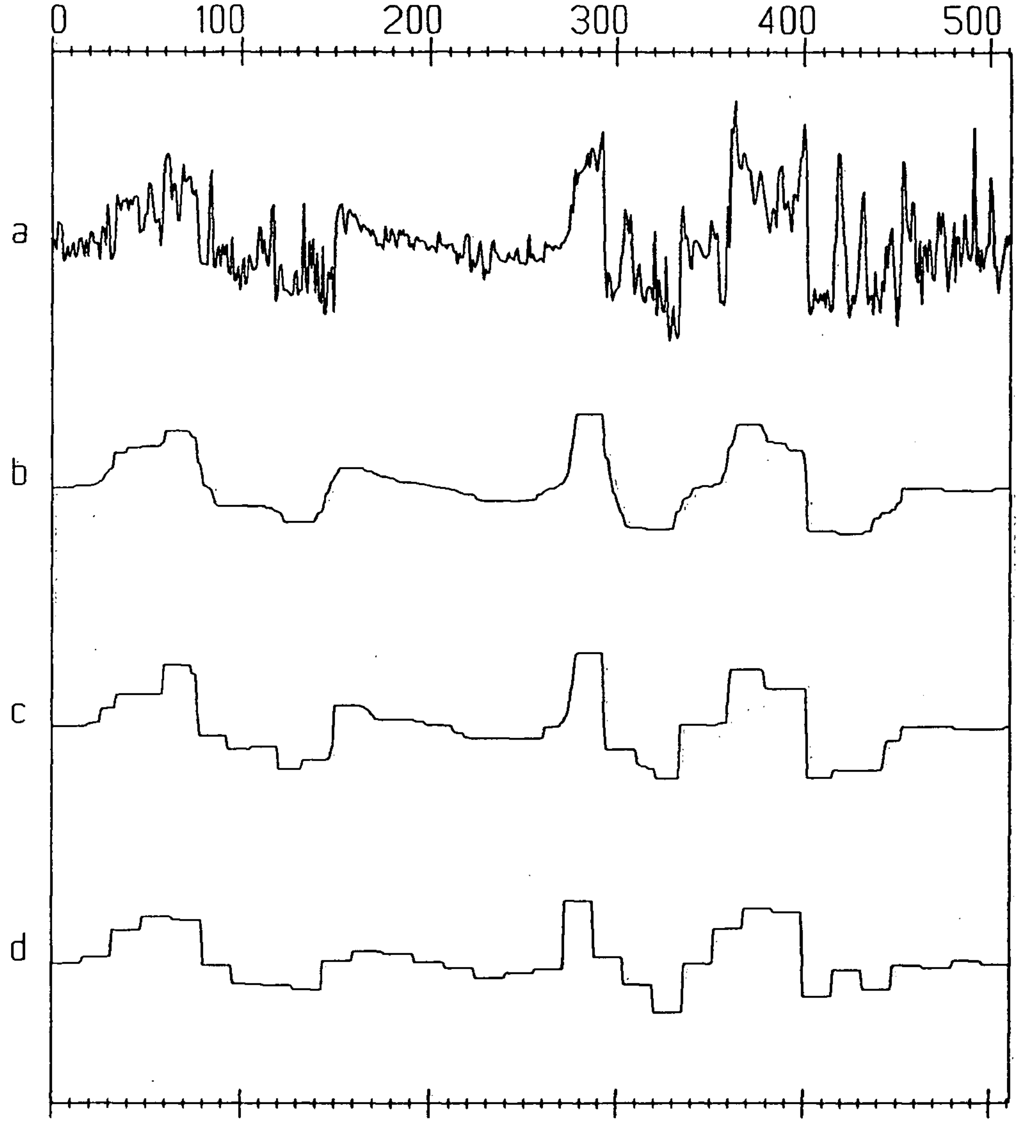

Figure 2.3 (a) Original data (same as in Figure 2.2), (b) root signal for an $N = 12$ median filter, (c) the root signal for the corresponding compound median filter, and (d) the Walsh low-pass version of (a) with equivalent shortest block length as in (b) and (c). The compound median result of (c) best represents the signal at the given minimum block length.

This indicates that compound median filtering is the superior technique for extracting the blocky portion of a signal.

The improved result of compound median filtering over conventional median filtering brings the behavior of a compound median filter close to the 'ideal transfer behavior' attributed to median filters and mentioned earlier. A practical application of compound median filtering will be discussed in section 2.7.

2.4 Median Decomposition

The following is a new and original nonlinear signal decomposition which is founded on the innovation of compound median filtering discussed in section 2.3.

The effect of the compounding process is to successively remove impulses of length $n = 1, 2, \ldots N$. Impulses of length 2 for example are only removed after those of length 1 have been removed. Since compound filtering removes impulses of a fixed length, the difference between present and past root signals will be a signal containing impulses of only one length. The signal may thus be decomposed into its constituent 'residual component' signals, each containing impulses of one duration.

If $v_0(j); j = 1, m$ represents the original sonic log, and $f_n(\cdot)$ represents the operation of a median filter of length $2n + 1$, then the 'residual component' with only impulses of length n is:

$$a_n(j) = v_{n-1}(j) - v_n(j) \qquad \text{where} \qquad v_n(j) = f_n(v_{n-1}(j)). \qquad (2.1, 2.2)$$

When this is done for $n = 1, N$, the signal has been decomposed into a root signal and a set of N residual components $a_n(j)$, and the original signal may be reconstructed as:

$$v_0(j) = v_N(j) + \sum_{n=1}^{N} a_n(j). \tag{2.3}$$

It is noted that this is not a linear decomposition since $a_n(j)$ is an arbitrary and unknown function.

Figure 2.4 shows each compound median root or 'median component' for $N = 43$ and Figure 2.5 shows the residual components $a_n(j)$ and the original signal reconstructed from equation 2.3. It should be mentioned that the reconstruction is exact and does not suffer from round off error as no multiplications or divisions are inolved in the computation of the median or residual components.

2.5 The 'Bloctrum'

Just as there is a frequency spectrum showing energy at different sine wave periods, so may the energy or amplitude at different impulse or block lengths be computed from a median decomposition. The amplitude A_n of all impulses of length n may be computed as:

$$A_n = \sum_{j=1}^{m} |a_n(j)| \qquad \text{for} \qquad n = 1, 2, \ldots N. \tag{2.4}$$

Since compound median filtering decomposes the signal into impulses or blocks, amplitude as a function of block length may be called a 'bloctrum'. The bloctrum of a trend-removed log is shown in Figure 2.6.

The bloctrum provides quantitative insight into the block distribution within the signal and may be employed as a quality measure in matching. For example, after

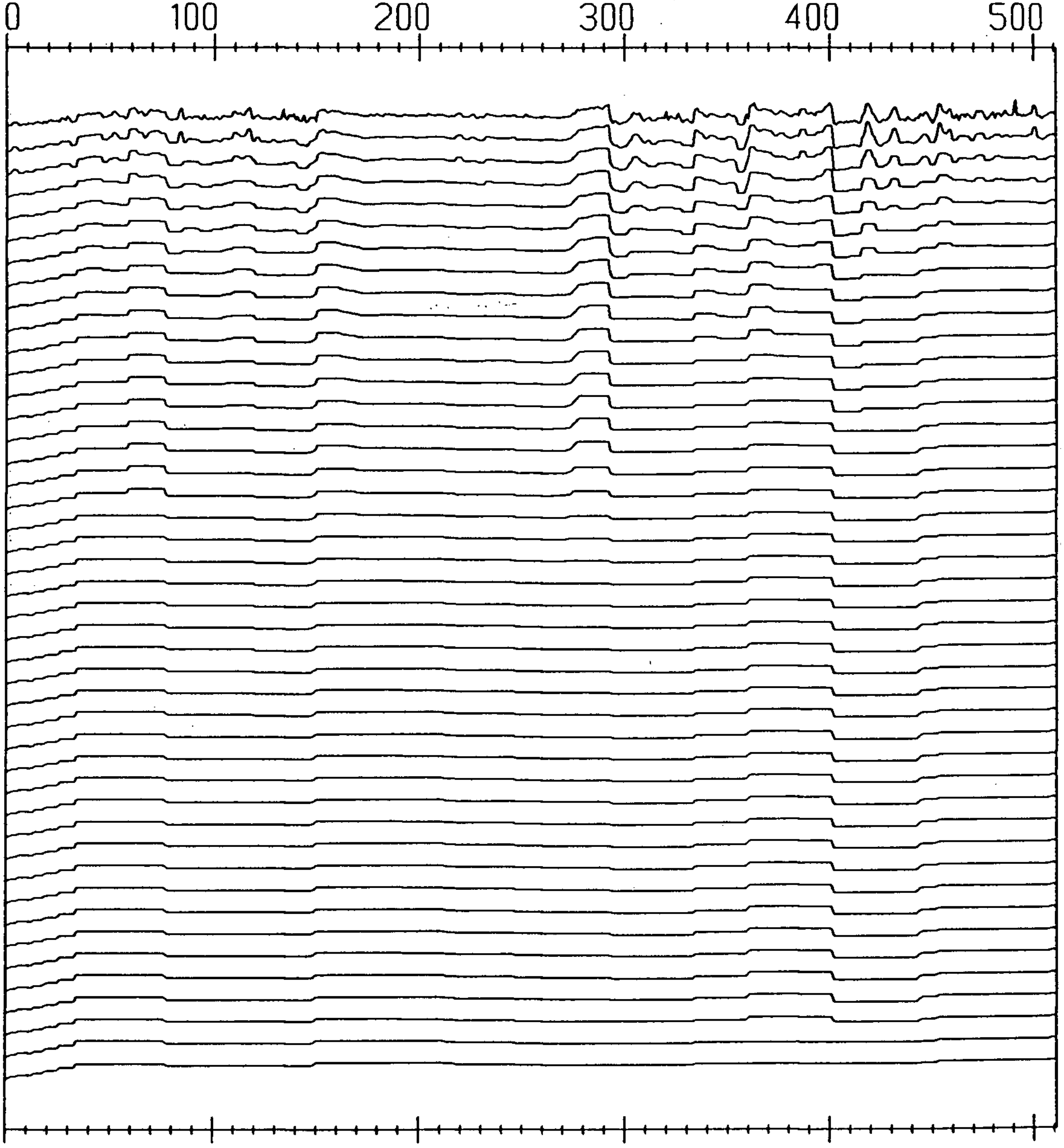

Figure 2.4 The original log and median components $v_n(j)$ for $N = 43$.

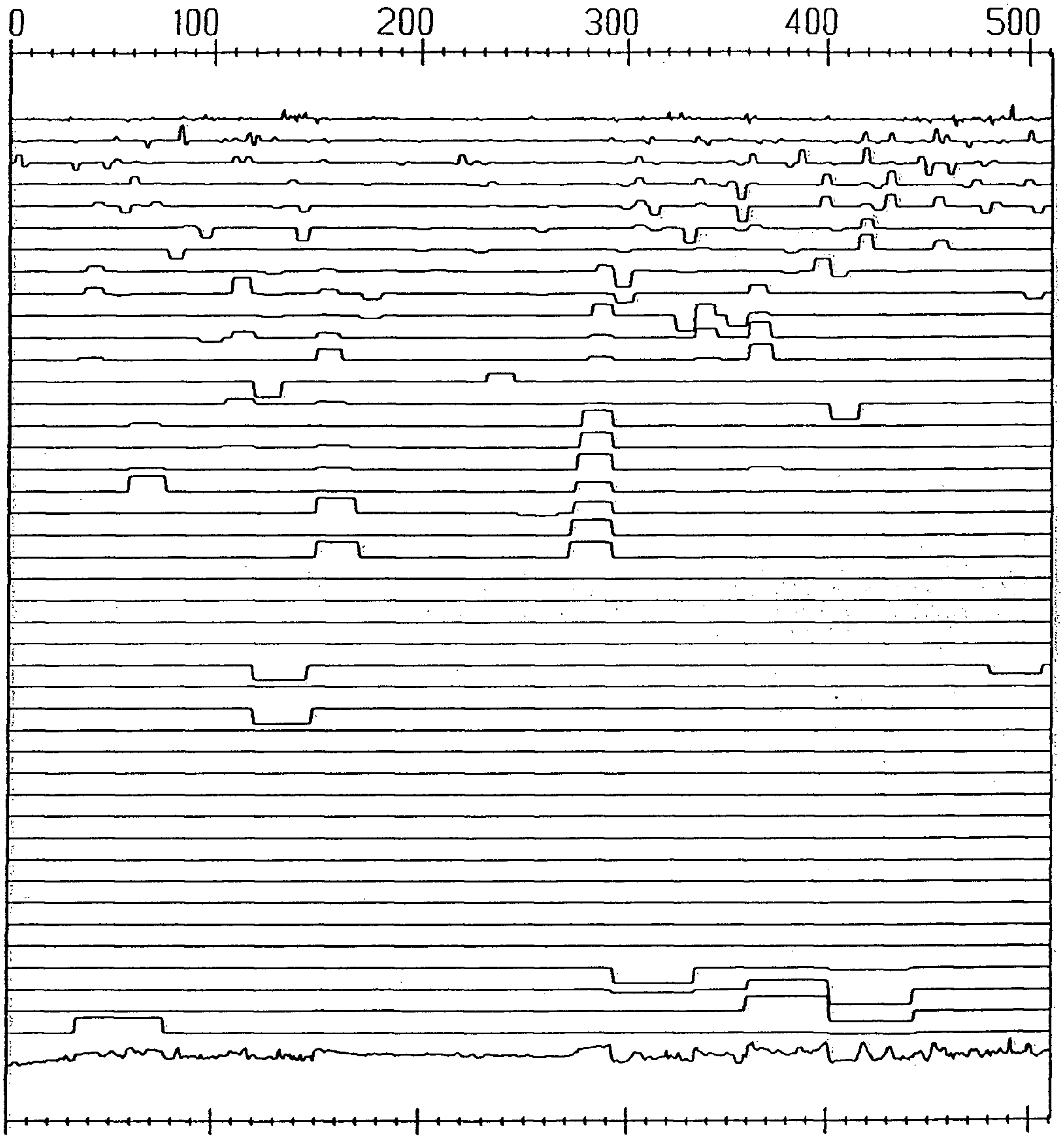

Figure 2.5 The residual components $a_n(j)$ corresponding to the median components in Figure 2.4, and the original signal reconstructed from equation (2.3).

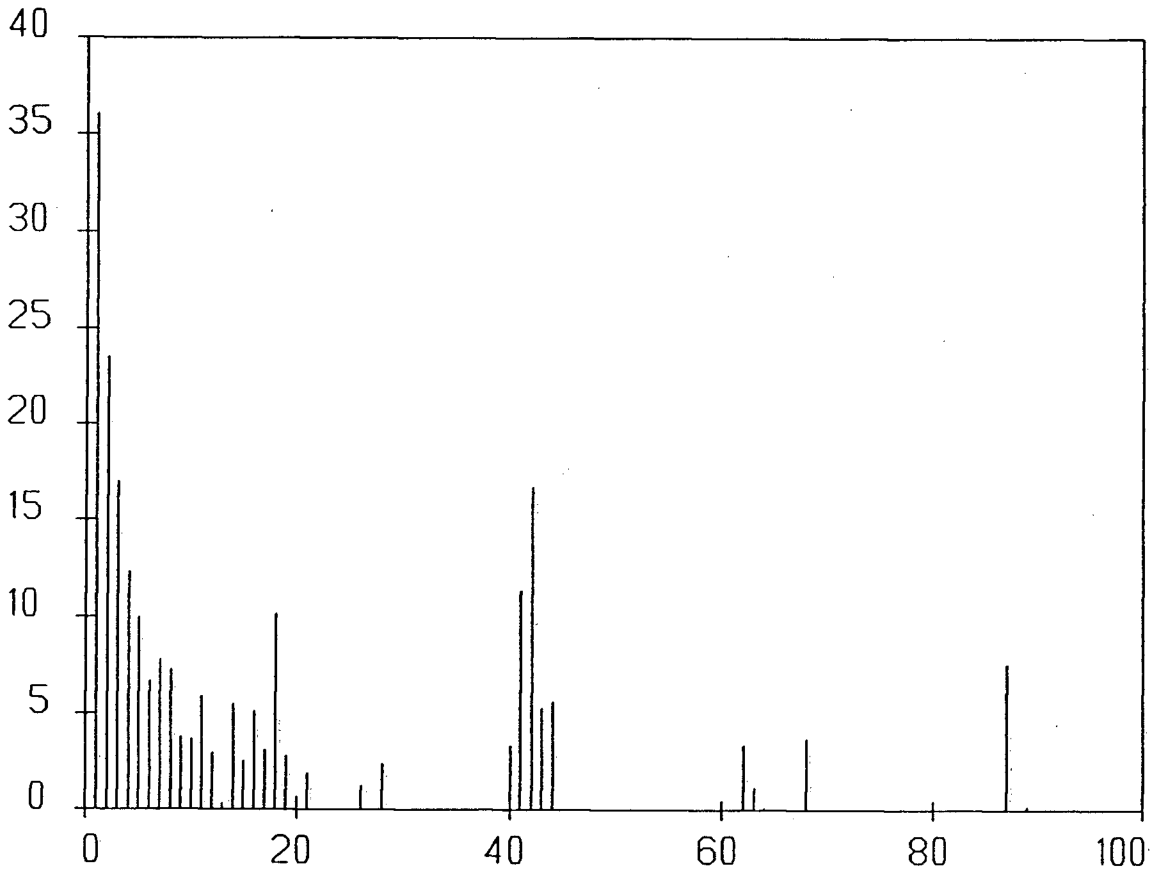

Figure 2.6 The bloctrum corresponding to the median decomposition of the original log in Figure 2.4 but with $N = 98$. Amplitudes have been normalized by the length of the signal.

matching, the block distribution within the logs should be similar, so a coherency measure between bloctrums should be higher for a better match. Also, the bloctrum may be used to reconstruct the signal with only blocks of significant energy. A threshold may be applied to the bloctrum or to each residual component individually so that only those residual components with significant amplitude would be used in a reconstruction. The signal may also be reconstructed exactly from fewer than N median components since some components have no contribution to the signal.

2.6 Median Scale Space

The operation of median filtering extracts signal from noise where the division between the two is at an exact impulse duration or block length. A longer median filter length passes longer blocks, and this may equivalently be seen as reducing resolution and producing a broader scale view of the log. The log may thus be viewed through scale by changing the length of the median filter.

The idea of scale in the areas of computer vision and pattern recognition has received a fair amount of attention in recent years. The goal is to quantitatively describe waveforms in one and two dimensions through a set of useful descriptive primitives. The local extrema of the signal and its derivatives are useful descriptive primitives, but the problem of scale consistently emerges as a source of difficulty. The location of edges for example, will vary in number and position, depending on the value of the signal at two or more points and the extent of the neighborhood around the point of measurement, so that a different description is obtained for different scales. What scale is the best to describe any particular waveform?

Witkin (1985) has come up with a multiscale approach to signal description. The idea is quite simple. He chooses as his parameter of scale the standard deviation (σ) of a Gaussian convolved with the signal. As σ is decreased continuously, beginning at a coarser scale, existing extrema move along the signal axis and new extrema occasionally appear at singular points. When the zero crossings are found along the signal axis at a number of intervals of scale, lines may be drawn in the (x, σ) plane or 'scale space' to follow this primitive descriptor through scale. As scale is coarsened, zero crossings disappear in pairs as local extrema disappear. Figure 2.7 shows the paths of zero crossings in scale space for an image intensity profile. Notice that the traces of zero crossings form arched lines at broader scales, due to the smearing affect of the Gaussian

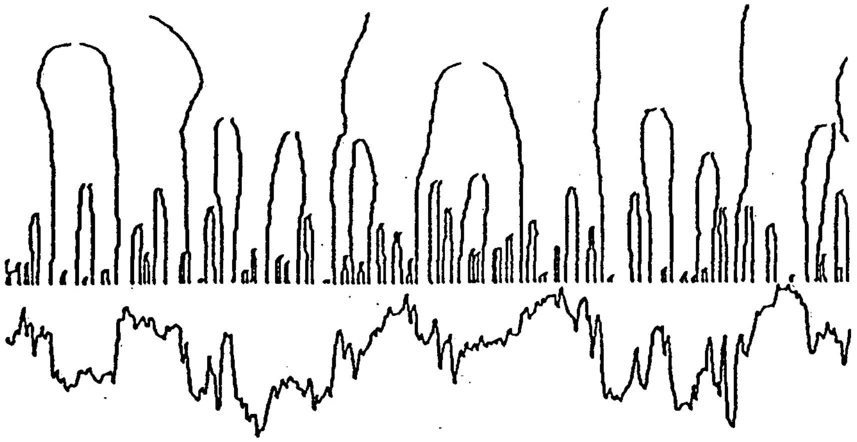

Figure 2.7 The paths of zero crossings through scale space. The scale parameter σ increases upwards. Notice the arches in the lines at broader scales due to the smearing effect of the Gaussian operator. (from Witkin (1984)) Compare with the paths of block edges through 'median scale space' in Figure 2.4.

operator. If this operator was convolved with a sonic log, the smearing would be more pronounced due to the abrupt jumps in the signal. An edge in the original signal would be shown at an incorrect location at a broad scale in (x, σ) space. A better parameter of scale in the case of blocky waveforms would be the compound median filter length. The paths of edges in 'median scale space' should remain straight, owing to the step preservation property, and should disappear rapidly over scale as the median window length is increased. Figure 2.4 shows a trend - removed sonic log through median scale space. The edges disappear at the the same x coordinate as scale is coarsened, indicating that their location has been preserved through scale. The superiority of median scale space for the description of sonic logs is clear and adds support for the use of median filters on these signals. More will be said later on the connection between a scale space point of view and pattern recognition, and in particular, matching through median scale space will be discussed.

2.7 Automatic Event Detection

The result of preserving sharp block edges provided by compound median filtering may be exploited in the form of an automatic event detector. Events may be defined as abrupt excursions from the preceeding trend, and since this also defines a block in a well log, block edges are events to be detected. The event detection algorithm proceeds as follows. After compound median filtering has been applied to the log, a differencing procedure may be applied to generate a psuedo-reflectivity sequence, with events indicated where magnitudes are large. Since some block edges may exist over more than one point, locations which show adjacent values of the same sign are merged to the location of the largest value. This merging procedure is done over the entire psuedo-reflectivity sequence so that all values become either zeros or singular spikes. A simple thresholding may then be applied to extract only the largest events.

The decision of the length of compound median filter to use is somewhat arbitrary. It must be as short as possible while still eliminating thin bed 'noise'. A filter with $N = 12$ was found suitable for the logs used in this project. Figure 2.8 shows a trend-removed log before and after compound median filtering, its psuedo-reflectivity, and the location of major events after merging and applying a threshold of 200 m/s.

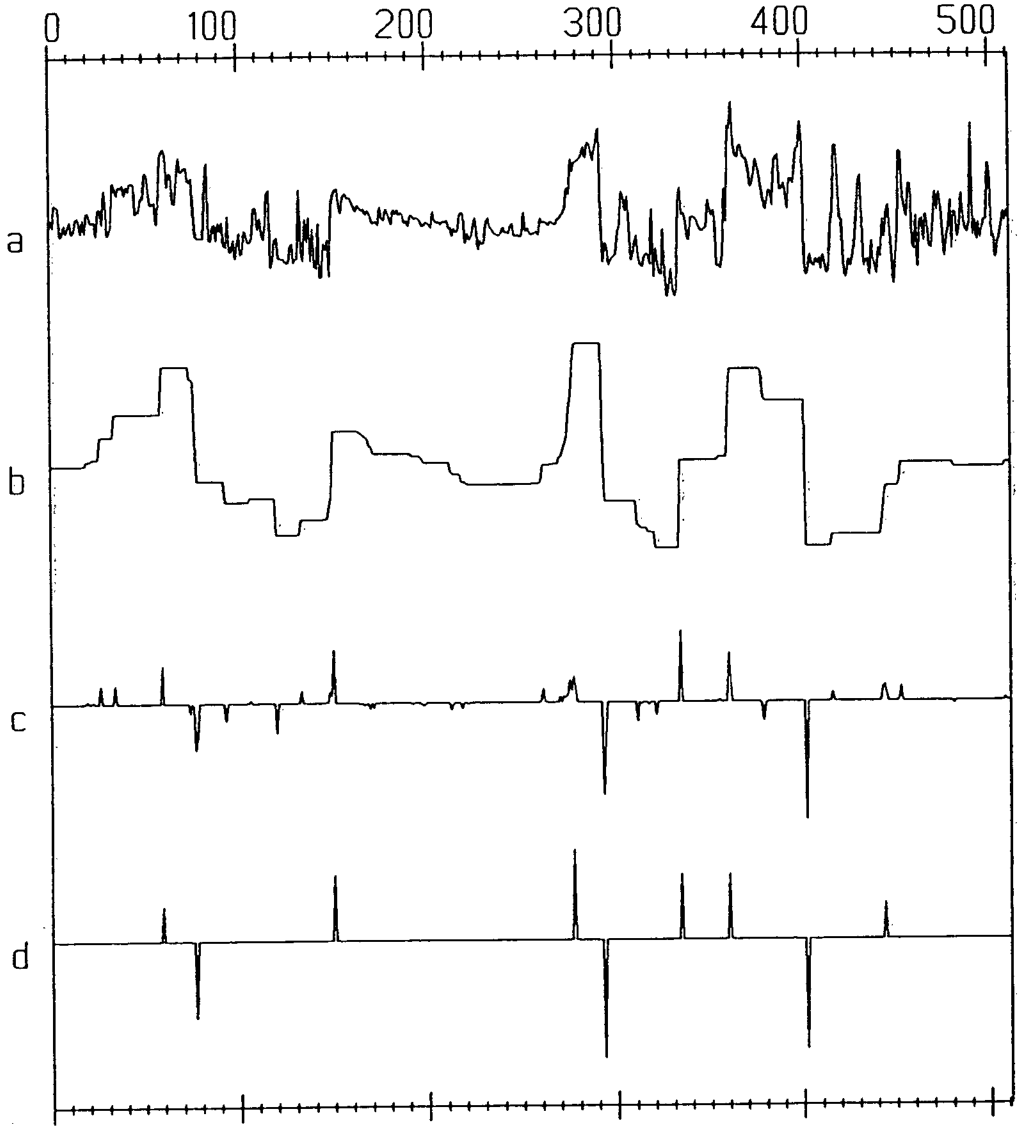

Figure 2.8 (a) original, (b) $N = 12$ compound median root signal, (c) the psuedoreflectivity and (d) the major events after merging and applying a threshold of 200.0.

2.8 Processing Sonic Logs For Matching

The goal of preprocessing the sonic data in this project is to enhance features for matching. In the case of well logs the most obvious matchable features are the block edges or steps. To enhance features for matching, two operations must be performed on the data.

First, the velocity or background gradient which varies from well to well must be removed as it can induce a d.c. bias to any similarity measure computed between portions of the logs. It was found that conventional high pass frequency domain filtering (actually the time domain analog) was unsatisfactory for trend removal. This is because narrow high amplitude blocks require low frequencies for their representation, and the removal of low frequencies induces sidelobes adjacent to these blocks (see Figure 2.9). To truely estimate the *background* trend, the trend upon which blocks sit, narrower high amplitude blocks should not be included. Median filtering can be used to estimate the background trend without 'feeling' blocks of a specified length ($\leq N$), so the removal of a sufficiently long median filter output solves the problem of sidelobes. To obtain an estimate of the velocity gradient or trend that does not include the high amplitude blocks, a 105 point median filter is applied to the logs. This is then removed from the logs to accomplish the high-pass operation. Figure 2.10 shows the filtering step and the removal of the background trend from a sonic log.

The second preprocessing operation performed on the sonic logs is low-pass filtering. Block edges are commonly imbedded in thin bed 'noise' and this can produce a greater uncertainty in any coefficient of similarity that is computed between portions of the signals, hindering the matching of block edges. Thus, a low pass operation is needed. Figure 2.11 shows the application of a 29 point median filter and an equivalent length

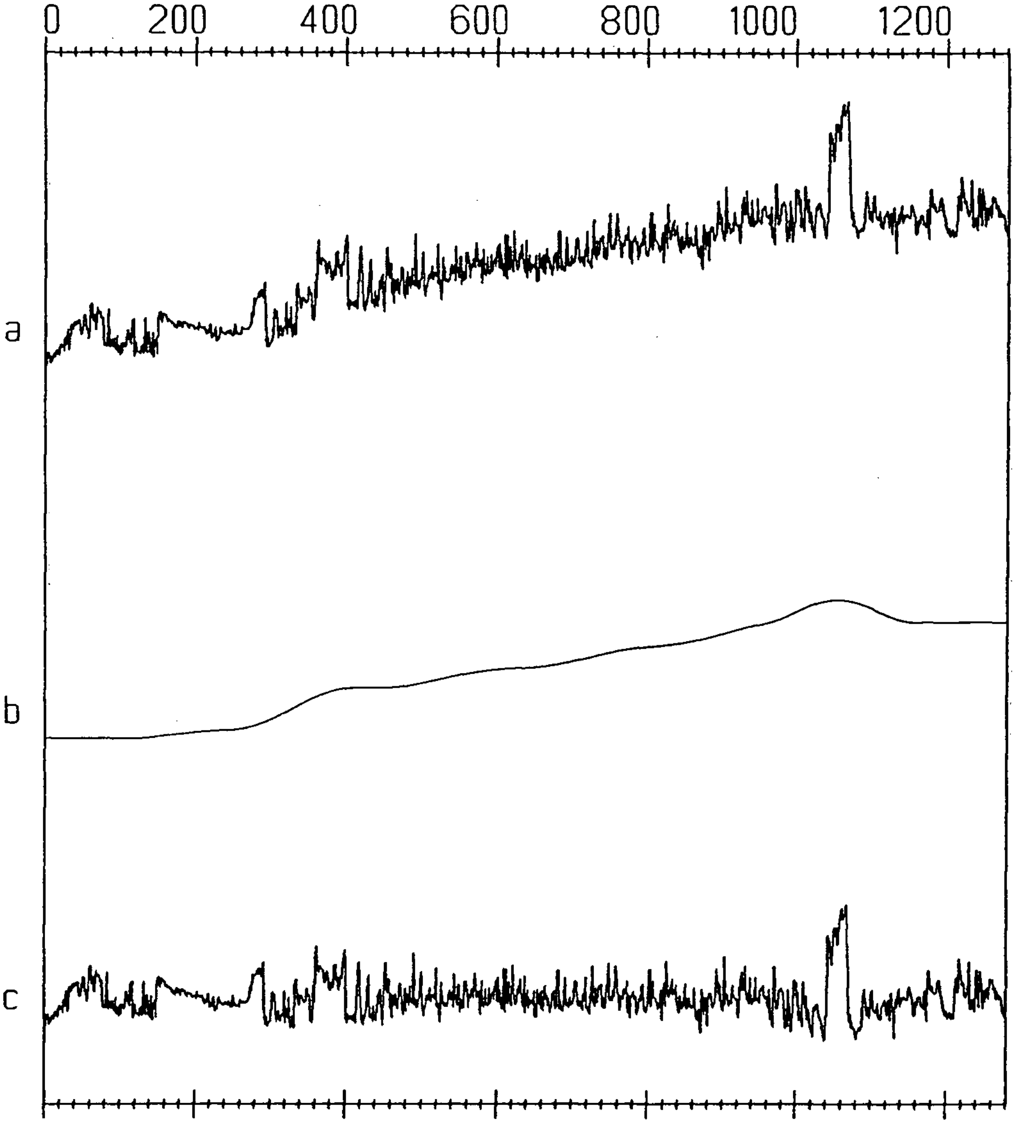

Figure 2.9 (a) original sonic log, (b) 105 point tapered mean of (a), (c) trend-removed log by subtracting (b) from (a). Notice the negative sidelobes induced adjacent to the large amplitude block located at 1020 - 1030 samples.

running mean applied an equal number of times. Both results remove the 'thin bed noise' but notice the preservation of edges in the median filter result.

The processing of the sonic logs before matching thus involves two steps. First the removal of a 105 point median filter output from the original data, and then the application of a 29 point median filter for the low pass operation. The result is data with blocks greater than or equal to approximately 15 samples or 45 metres in length. Figure 2.12 shows the preprocessing steps one sonic log and Figure 2.13 shows all five logs used for matching after pre-processing.

Unfortunately, for the purposes of processing sonic logs for matching, compound median filtering was found unsuitable, perhaps because it is too discriminating. It seems that a smoother transition between blocks (constant neighborhoods) makes matching easier. If some smoothing after compound median filtering was performed the result might be suitable for matching, but this possibility was not explored in the interest of keeping the preprocessing as simple as possible.

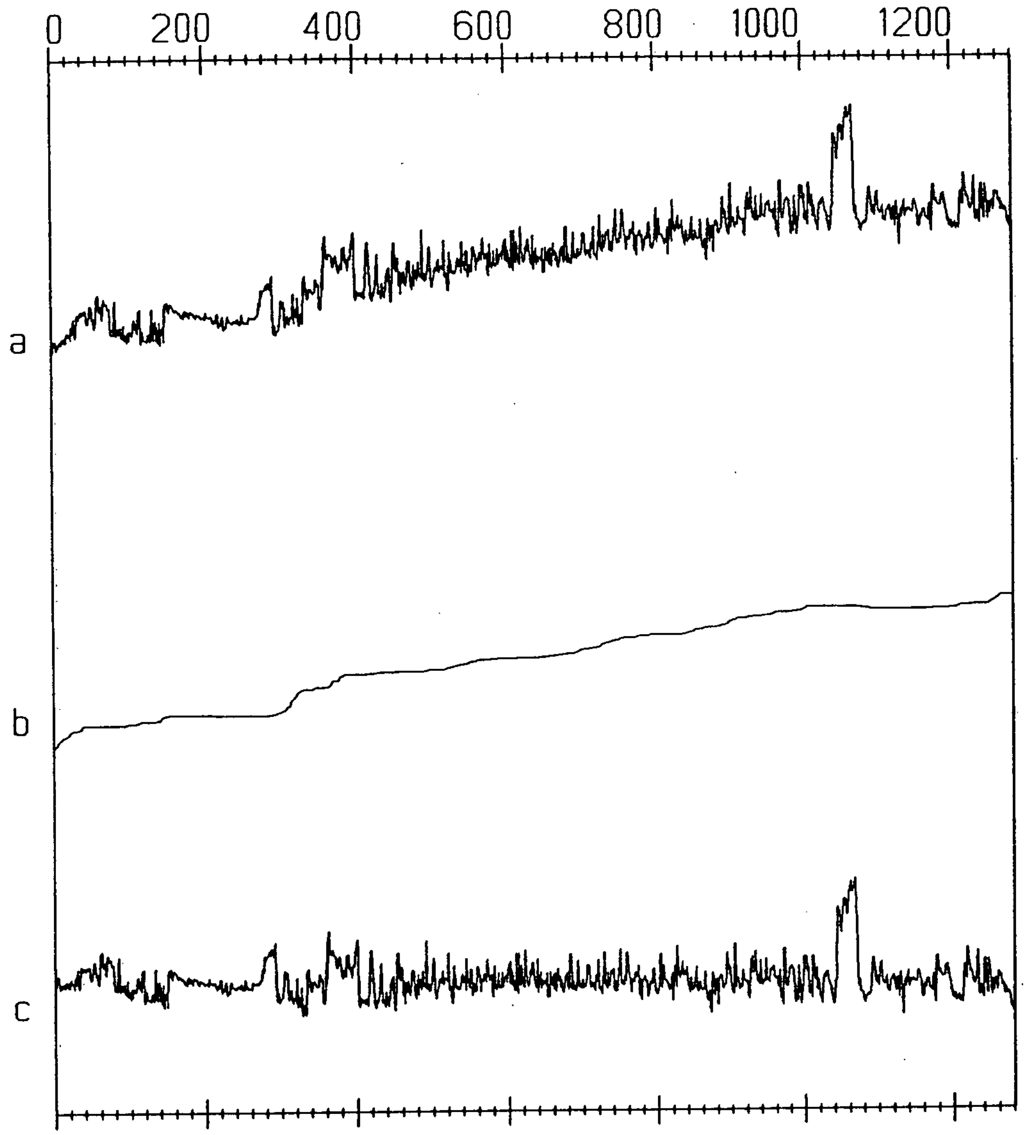

Figure 2.10 (a) original sonic log, (b) 105 point median filter of (a), (c) trend - removed log (a minus b). No negative sidelobes are seen adjacent to high amplitude blocks.

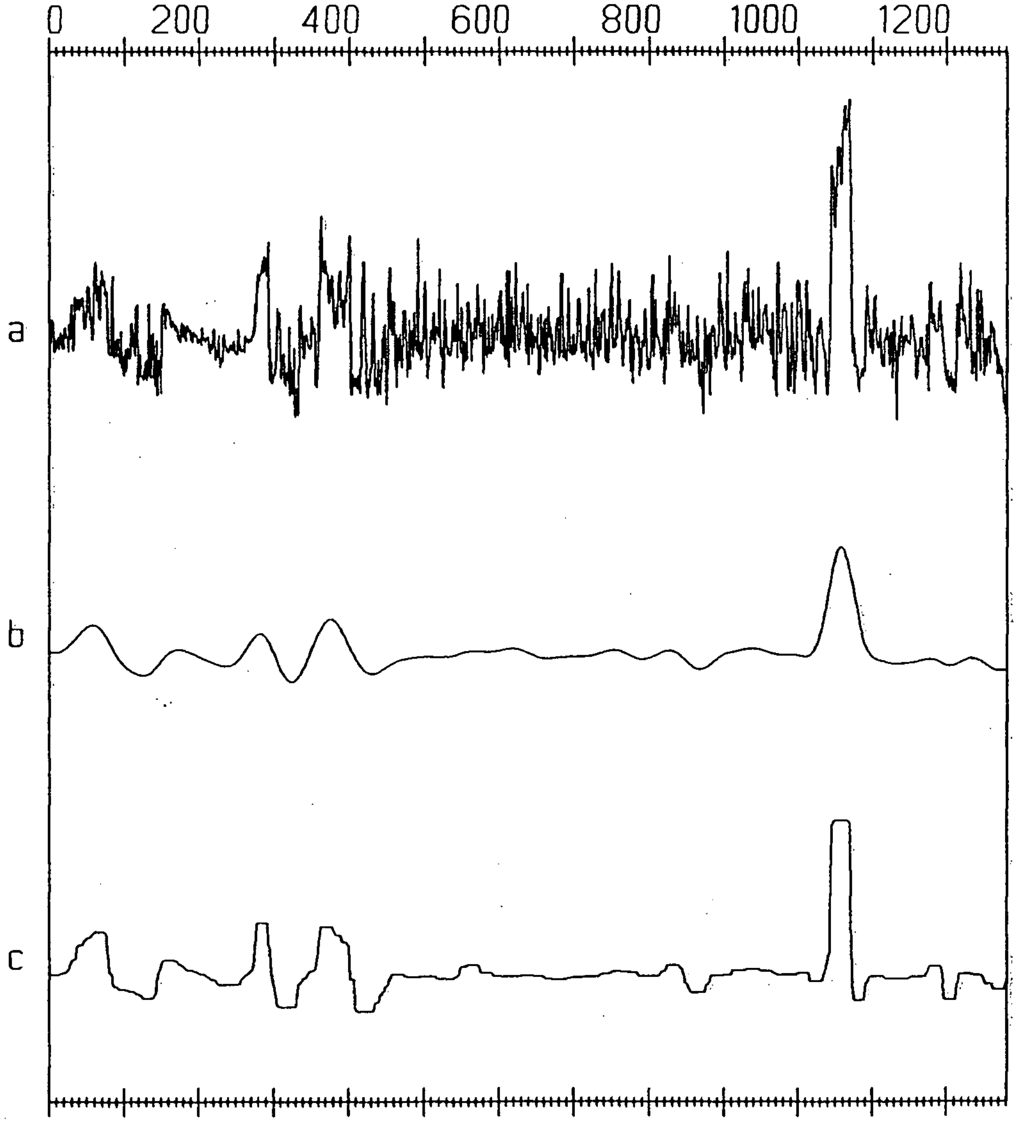

Figure 2.11 (a) The trend-removed sonic log of Figure 2.10c, (b) 29 point running mean filter applied the same number of times (3) as was the 29 point median filter in (c) for convergence to the root signal. Notice the preservation of block edges in the median filter result.

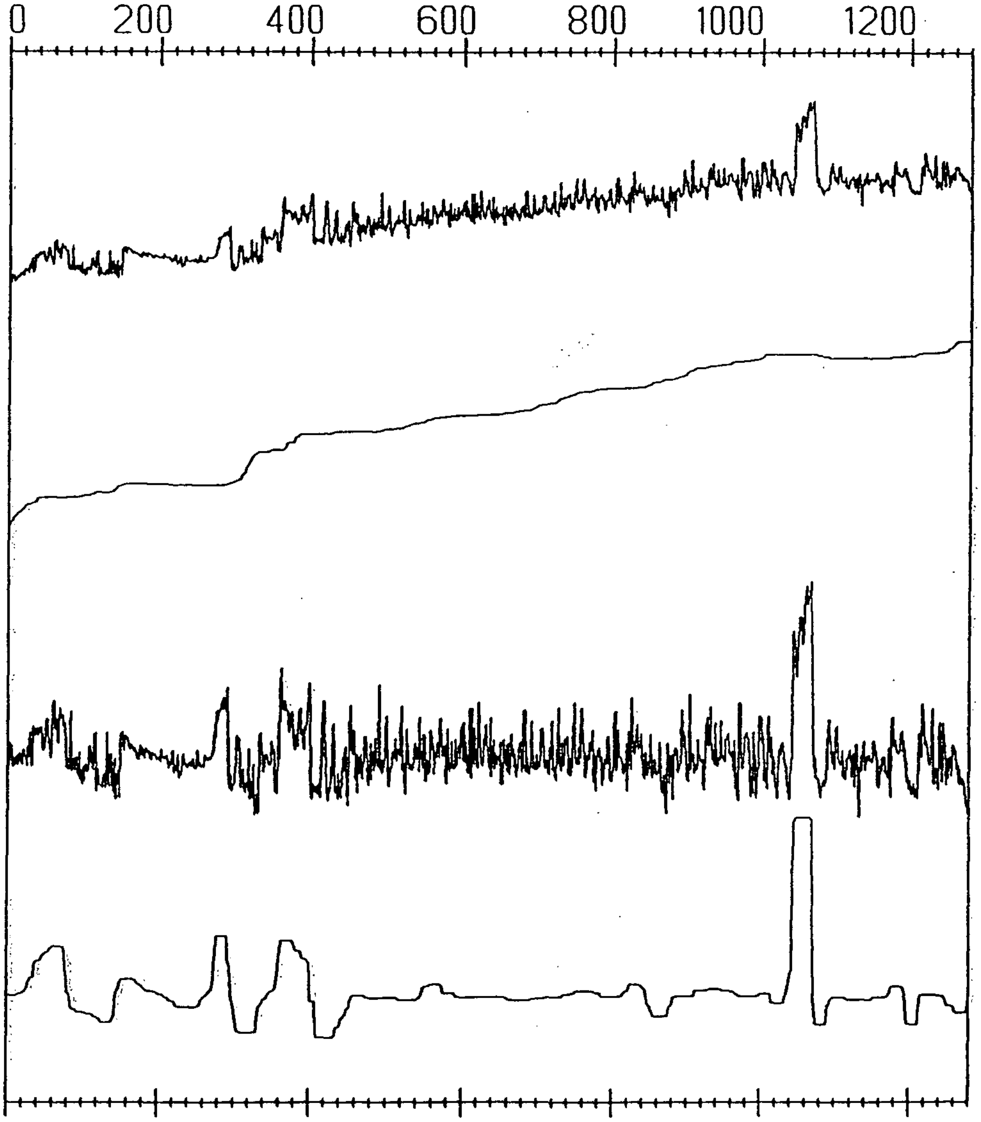

Figure 2.12 The preprocessing steps applied to one sonic log. A 105 point median filter output is subtracted from the original and then a 29 point median filter is applied to the trend removed log. The log now has blocks of length greater than 14 samples in duration.

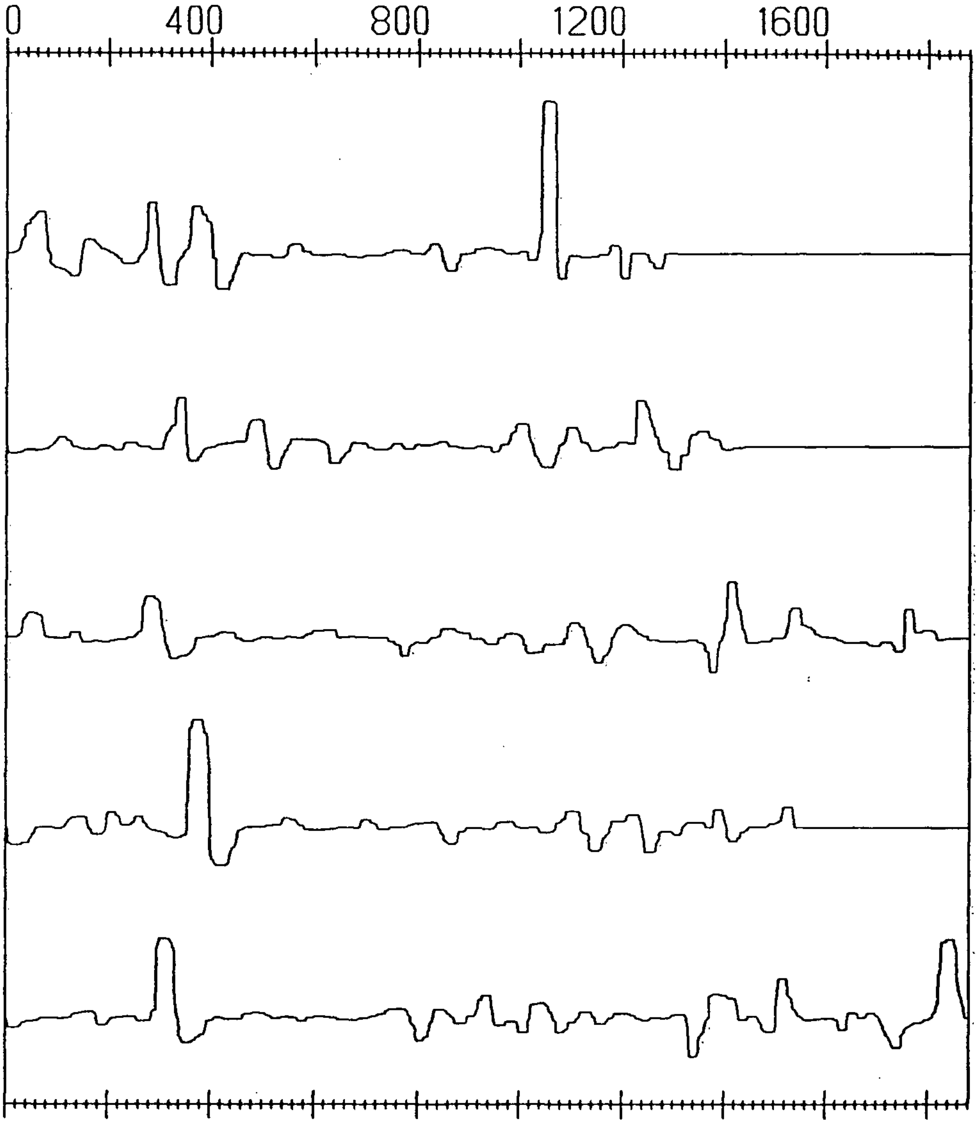

Figure 2.13 The five logs after trend removal and median low-pass preprocessing. The background trend has been removed with a 105 point median filter and low-pass median filtering has been applied with a filter of length 29. The logs now have blocks of length greater than 14 samples.

Chapter 3

MULTIPLE DYNAMIC MATCHING

3.1 Introduction

The foundation of multiple dynamic matching is in dynamic waveform matching, reviewed in section 1.3. It has as its basis a matrix known as a dissimilarity or similarity matrix depending on whether the functional values that make up the matrix are to be locally minimum or maximum for optimality. The matrix used in multiple dynamic matching is a *similarity* matrix although this is not necessary for the method to succeed. As with dynamic waveform matching, multiple dynamic matching or 'MDM' views the correlation problem as a pathfinding problem through the similarity matrix, but rather than finding only one path, many possible paths are found. The connection to the inverse approach to signal correlation (section 1.4) is that path optimality will be decided using global norms. In the sections that follow the details of MDM will be explained, beginning with the fundamentals of the similarity matrix.

3.2 The Similarity Matrix

Since it is desired to maximize the similarity between the reference signal $r(x)$; $x = 1, X$ and the test signal $t(x)$; $x = 1, Y$ everywhere along their extent, the signals may be partitioned and a local similarity measure evaluated between windows of each signal, where the relative lag between windows is given an upper bound $2L$. This bound on the relative lag is a measure of the greatest structural difference expected between the two signals and corresponds to the bandwidth of the similarity matrix S_{ij}, computed as:

$$S_{ij} = \sum_{l=1}^{L} \sum_{i=1}^{m-l} s(r_{i\Delta x}, t_{(i+l)\Delta x}, K)$$

$$S_{ij} = \sum_{l=0}^{L} \sum_{j=1}^{n-l} s(r_{(j+l)\Delta x}, t_{j\Delta x}, K), \tag{3.1}$$

where the sampling increment between functional values is Δx, $n = (X - K)/\Delta x + 1$ and $m = (Y - K)/\Delta x + 1$ are the number of points in the x and y directions respectively, s is the particular similarity functional and K is the window length over which similarity is computed. As indicated in equaton (3.1), functional values are computed for (x, y) pairs corresponding to the locations of the first point in each of the windows of the reference and test signals respectively, and computation proceeds up the diagonals of the matrix for each relative lag l. This is done to make computation of the matrix most efficient, as advantage can be taken of operations already done. Figure 3.1 illustrates the computation of the similarity matrix S_{ij}.

3.2.1 The Semblance Functional

The similarity functional used for the computation of S_{ij} in this project is semblance. The semblance functional for two signals $r(x)$ and $t(x)$ at windows i and j and for a window length K is:

$$s = \frac{\sum_{k=1}^{K} (r_{i+k-1} + t_{j+k-1})^2}{2 \cdot \sum_{k=1}^{K} (r_{i+k-1}^2 + t_{j+k-1}^2)}. \tag{3.2}$$

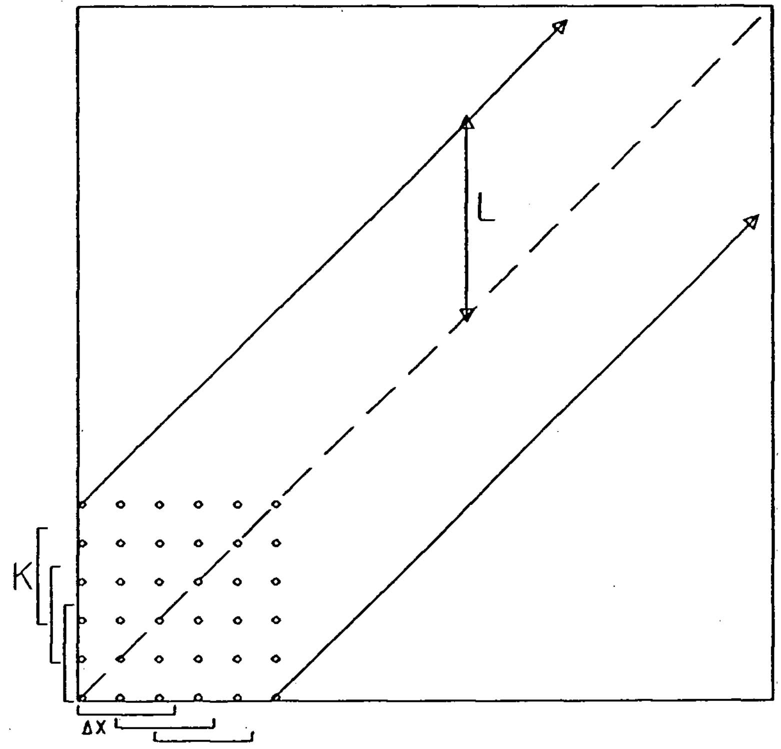

Figure 3.1 Computation of the similarity matrix. The half bandwidth is L, the similarity window length is K, the matrix sampling increment is Δx, and computation proceeds up the diagonals to take advantage of operations already done.

Semblance is a measure of the output/input energy ratio and is related to energy normalized cross correlation and summation or stacking. Neidell and Taner (1971) conclude from tests on multichannel seismic data that semblance has greater discriminating power in the case of ambiguously coherent events. Semblance has the range $(0,1)$ and not $(-1,1)$ as does cross correlation, and so has no built in phase or anticorrelation discrimination ability. In fact this is desired since in the case of matching sonic logs there is no utility in considering anticorrelations. Semblance is also related to the F-statistic

(Douze and Laster 1979). Although their experiments were performed on synthetic examples, their work indicates that larger semblance values indicate greater confidence in there being coherent energy between signals.

3.2.2 Smoothness and Sampling

Since the matching problem will be solved by pathfinding through the similarity matrix, it must be sufficiently smooth to avoid errors. It is still desired to sample the matrix (every Δx) to reduce its size and the cost of the algorithm, but this must not be done at the expense of accuracy. The question then becomes what is the largest Δx that can be used to provide greatest efficiency while still retaining acceptable accuracy. The answer depends on the low pass median filter length.

Median filtering leaves blocks of greater than N in length (for a filter of length $2N + 1$), so from sampling theory each block must be sampled at least once to avoid aliasing. This puts an upper bound on the matrix sampling increment Δx, and means that $\Delta x < N$. Now that Δx can be fixed, the matrix must be computed such that it is sufficiently smooth. It turns out that the smoothness of the similarity matrix can be controlled by the window length K of the semblance functional. The exact transfer function of the semblance functional is difficult to quantify, but as it is known to behave similarily to a stack or mean, it can be said that longer windows produce smoother matrices.

In an attempt to establish the best value of K given Δx and N, a 2D Fourier analysis has been done for similarity matrices computed using different semblance window lengths. It may be conjectured that below some value of K, noise of significant energy might appear in the matrix, making it unreliable for pathfinding. The data from which the matrices were computed for the 2D analysis are logs 1 and 2 after preprocessing

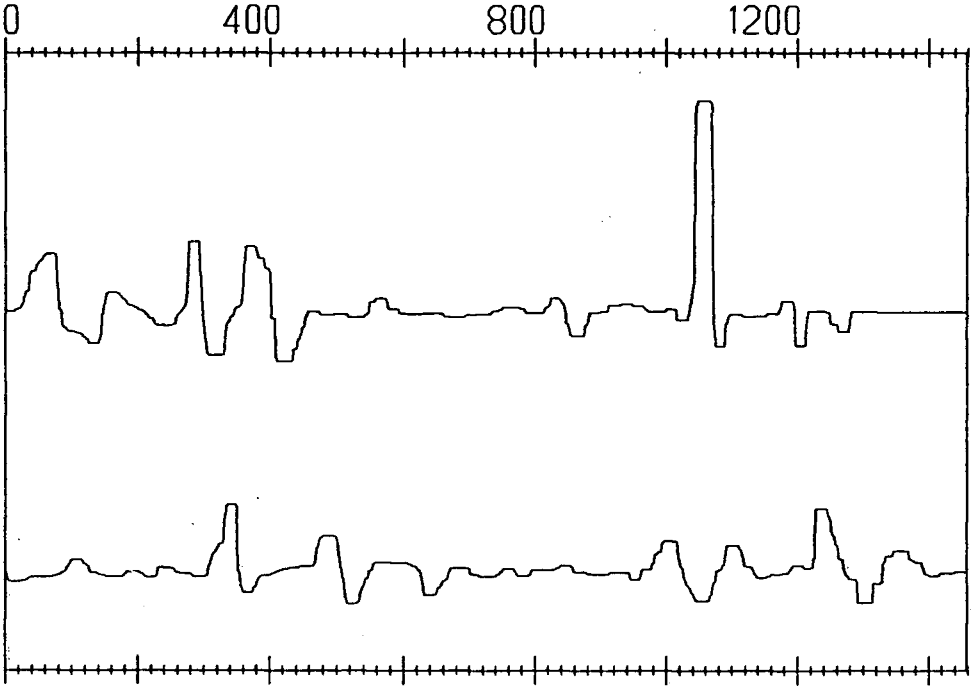

Figure 3.2 Logs 1 and 2 after preprocessing. These two logs will be used in an example of multiple dynamic matching which will be referred to as $m_{2/1}$.

(see Figure 2.12). These logs will also be used as an example to illustrate the multiple dynamic matching method and will be referred to as example $m_{2/1}$, with log 2 as the reference log. Figure 3.2 shows logs 1 and 2 after the preprocessing steps of trend removal and low-pass median filtering ($N = 14$). Figure 3.3 shows similarity matrices computed for example $m_{2/1}$ with Δx =10 and $K = 20, 60, 100$ and their corresponding 2D power spectra. Notice that significant energy remains below the Nyquist frequency for all examples so this cannot be used to determine the best semblance window length. The 'character' of the matrices is different however. It is known that shorter windows allow for the matching of finer scale features. Given the shortest block length in the sonic logs (N), one might want the semblance window to be of approximately the same length. However, too short a window leads to an unreliable estimate of similarity (Douze

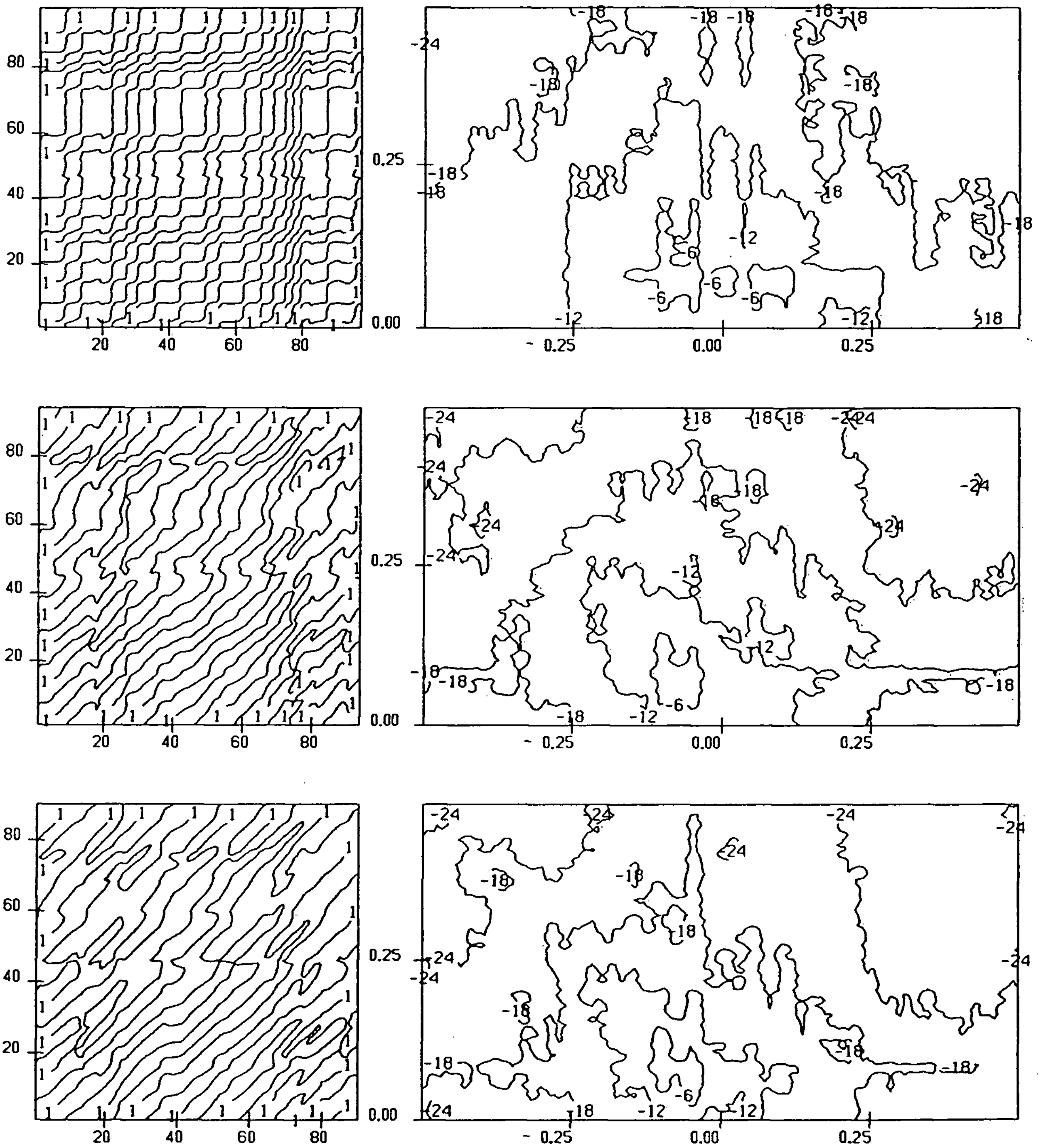

Figure 3.3 The similarity matrices (left) and 2D power spectra (right) for (a) $K = 20$, (b) $K = 60$, and (c) $K = 100$. All matrices show energy dropping to a very low value well below the Nyquist frequency, so this can not be used as a criterion for choosing the best value for K. (The spatial Nyquist frequency is expressed in terms of cycles per sample and so equals 0.5) The similarity matrix in (b) shows the best character relative to *a priori* knowledge.

and Laster 1979), so longer window lengths are needed for reliable similarity estimates. It is also known that trends in similarity should appear at roughly 45 degrees in the matrix, and should be clearly defined. The matrices of Figure 3.3 may now be viewed in a different light. For $K = 20$, contours of similarity follow block edges closely, with higher values where blocks of similar amplitude coincide. This matrix does not conform to the expected 'good character' matrix and would clearly be unreliable for the extraction of trends in similarity. The $K = 60$ matrix shows the good character of well defined trends in similarity at roughly 45 degrees. The $K = 100$ matrix also shows coherent trends, but since it is desired to keep K as small as possible while still preserving the coherence of similarity trends, $K = 60$ is a better choice. The selection of K is thus somewhat arbitrary. Given that $\Delta x \approx 2/3\ N$ to satisfy the block sampling criterion, $K \approx 6\Delta x$ has been found appropriate to ensure the smoothness and character of the similarity matrix.

3.3 Local Confidence

The semblance coefficient between windows of points in the logs will be large (close to 1) when portions of the logs have similar amplitude and shape. In the similarity matrix semblance coefficients are located side by side, defining a surface, and different structures on this surface correspond to different situations of matching. For example, if amplitudes are roughly constant over a portion of the logs and events are small, any coefficient of similarity will not change appreciably between any two windows of these 'quiet' portions of the logs. The result is a flat region or plateau in the similarity matrix and a poorly defined match. Low confidence should therefore be placed in matches between portions of the logs with events of insignificant magnitude. Conversely the concept of high local confidence may be explained by the following illustration.

Imagine windows of each log 'sitting' on blocks of large and similar amplitude. As one window remains stationary and the other is slid forward over the other log, it passes over the block edge and the amplitude drops rapidly. Similarity values drop also due to the sudden large difference in amplitudes between the windows. This will occur if the stationary and sliding windows are interchanged, so that a peak will appear in the similarity matrix in both x and y directions at some coordinates (x, y) corresponding to the vicinity of the two block edges. If the windows are slid at constant increments Δx along both logs (following a diagonal in the matrix), the peaks will flatten where a match becomes unclear, and will sharpen where a significant match can be seen. The occurance of repeated significant matches appears in the similarity matrix as a well defined trend or ridge. Local confidence in a match is therefore related to the magnitude of local similarity S, at a point on the matching function $(x, m(x)$, that is to $S_{x,m(x)}$, and to the quality of the ridge in the direction of $m(x)$. To quantify the quality of a ridge, the second derivative may be computed perpendicular to the direction of $m(x)$ as it measures the downward curvature. The local confidence (LC) norm is now defined as:

$$\mathrm{LC} = -S_{x,m(x)} S''_{x,m(x)^{\perp}} \tag{3.3}$$

Figure 3.4 shows portions of the logs of example $m_{2/1}$ and the corresponding similarity matrix. Where event amplitudes are large coherent trends in similarity are seen, and where events are smaller trends are less well defined. Each well defined ridge represents a region where matching confidence is high, and may be viewed as a portion of the matching function, $m(x)$. Notice that ridges can run roughly parallel to one another, indicating more than one possible significant match for any one event on a log. This is due to the periodicity in the blocks of the logs.

The best overall matching function will be the function for which local confidence remains high throughout its length. This is easily expressed in terms of the mean local confidence (MLC) norm:

$$\text{MLC} = -\frac{1}{P}\sum_{p=0}^{P} S_{x,m(x)} S''_{x,m(x)^{\perp}}, \tag{3.4}$$

where P is the length of the matching function. This is a global norm as it assesses the optimality of the entire matching function. More will be said on the computation of MLC in section 3.5.1.

3.4 Ridge Points

Since nothing is known *a priori* about the optimum $m(x)$, scanning S_{ij} along all possible paths to find the maximum MLC is impractical. To handle this problem, points are located for which the local confidence is known to be large. The magnitude of S_{ij} may be constrained to be large by requiring that it exceeds a significance threshold. The negative of the second derivative may be constrained to be large by locating only points which are peaks perpendicular to $m(x)$. Nothing is known *a priori* about the local direction of $m(x)$, but it can be assumed to be in one of four directions defined by the eight points surrounding a central point. A simple test is then implemented to ensure that the point appears as a peak in three out of four directions. These two criteria make the location of local confidence maxima or 'ridge points' efficient, as the matrix S_{ij} is simply scanned to find the points for which the above criteria are satisfied. The following summarizes the test for ridge points.

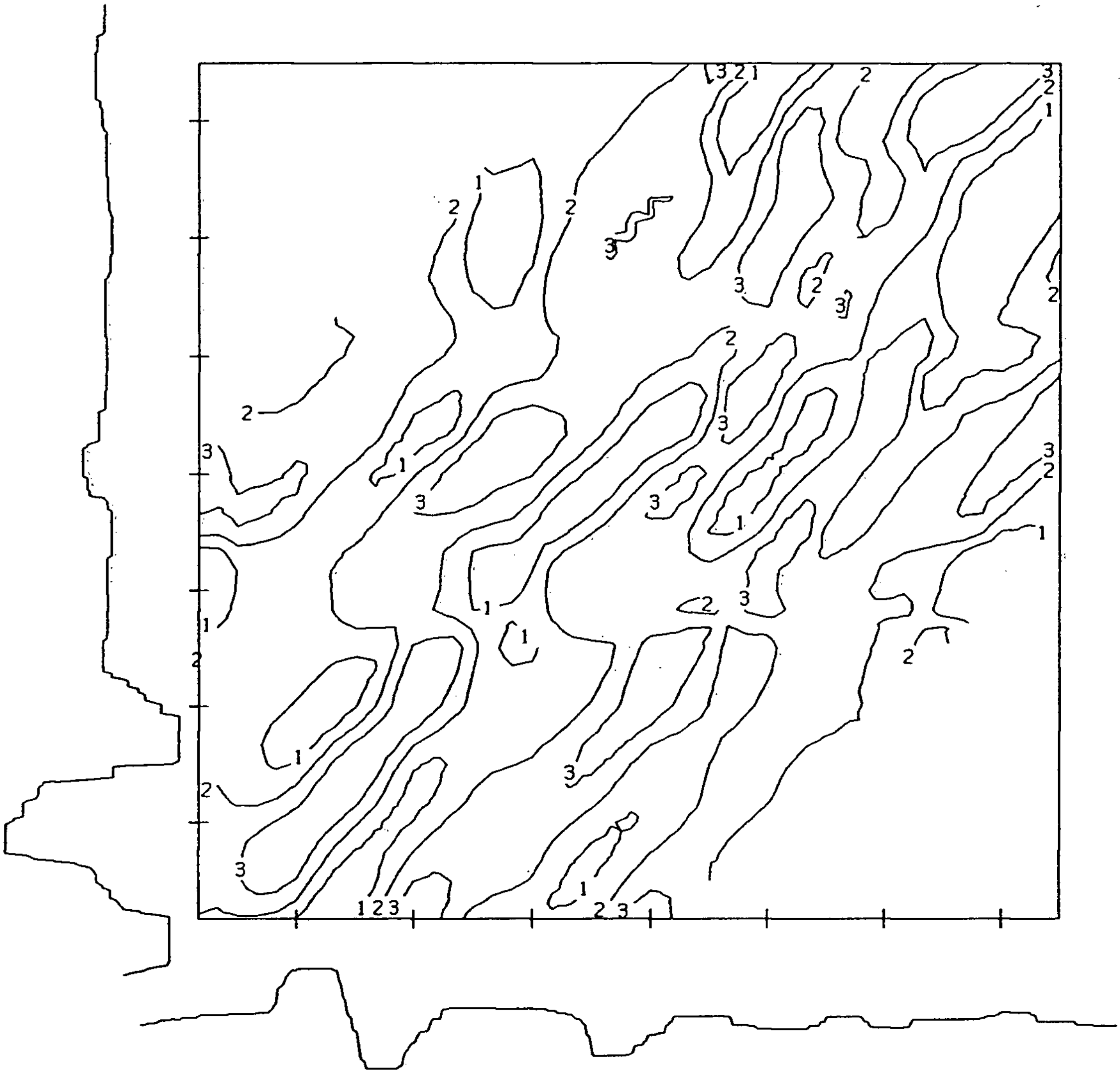

Figure 3.4 Portions of the preprocessed logs and the similarity matrix with contours of similarity labelled so that 1 is low and 3 is high. Notice that where event amplitudes are large, at the beginning of the traces, trends in the similarity matrix are well defined. Toward the end of the traces where events are smaller, trends in similarity are less coherent.

1. $S_{ij} > .65$ (significance threshold).
2. 3 of the following are true
   (a) $S_{i,j} > S_{i,j-1}$ and $S_{i,j} > S_{i,j+1}$
   (b) $S_{i,j} > S_{i-1,j}$ and $S_{i,j} > S_{i+1,j}$
   (c) $S_{i,j} > S_{i-1,j-1}$ and $S_{i,j} > S_{i+1,j+1}$
   (d) $S_{i,j} > S_{i-1,j+1}$ and $S_{i,j} > S_{i+1,j-1}$

These criteria allow the ridge points to be located adjacent to one another along the top of coherent and significant trends in the similarity matrix. Figure 3.5 shows the similarity matrix for example $m_{2/1}$ with one contour at 0.5 and ridge points as x's.

3.5 Subpath Determination (Dynamic Matching)

Establishing a set of potential pieces of the correct matching function requires joining the ridge points into 'subfunctions' or 'subpaths'. This is done with a simple dynamic programming scheme.

A ridge point is defined to be a member of a subpath if (1) it is within a prescribed distance D from the previous ridge point and (2) the slope of the segment that joins them is non-negative. The subpaths are thus constrained to be non-decreasing. Figure 3.6 shows the local path constraint used for subpath finding. The algorithm to determine the set of subpaths proceeds as follows.

The ridge points are sorted by their distance from the origin, and subpath finding begins with the ridge point nearest the origin. If the distance d from the last ridge point in the subpath to the next nearest ridge point exceeds D, that is if $d > D$, then the subpath is terminated. When the first subpath is terminated, all ridge points belonging

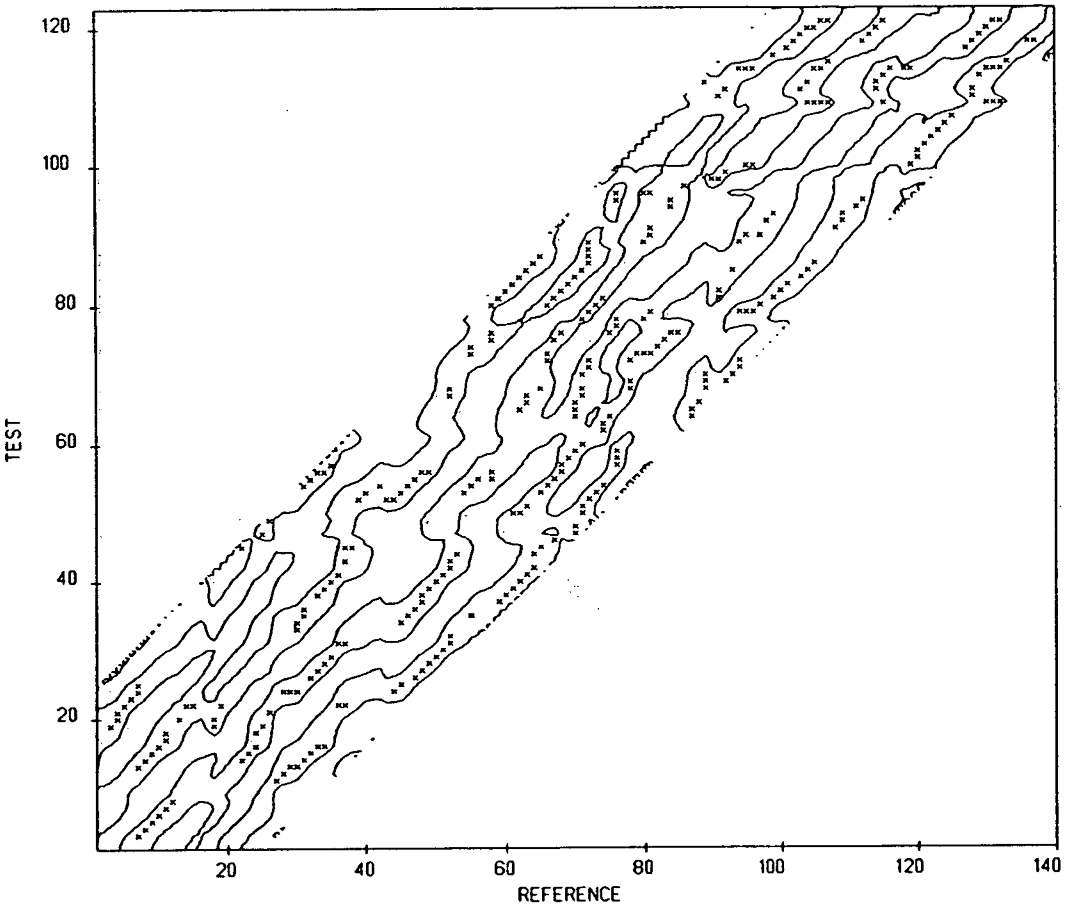

Figure 3.5 Location of ridge points as x's with one contour at 0.5.

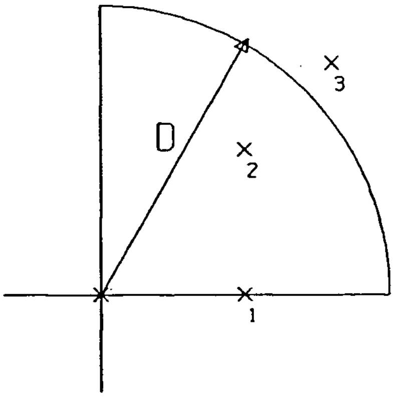

Figure 3.6 The local path constraint used in dynamic matching to connect ridge points into subpaths. Ridge points 1 and 2 are legal choices, ridge point 3 is not.

to it are removed from the set of 'possible subpath points', and the next ridge point nearest the origin begins a new subpath. The procedure is repeated and all subpaths have been found when the set of possible subpath points is empty. Also, if a subpath is less than a prescribed length L it is deemed insignificant and is removed from the set of subpaths. This puts a minimum significant length on the subpaths as prescribed by the definition of local confidence discussed in section 3.3. Figure 3.7 shows the ridge points of Figure 3.5 connected into subpaths with $D = 3.5$ and $L = 4.2$. These parameter values were established from trial and error and may be converted to well log samples for generality by multiplication by Δx. More will be said on the determination of these parameters in Chapter 4.

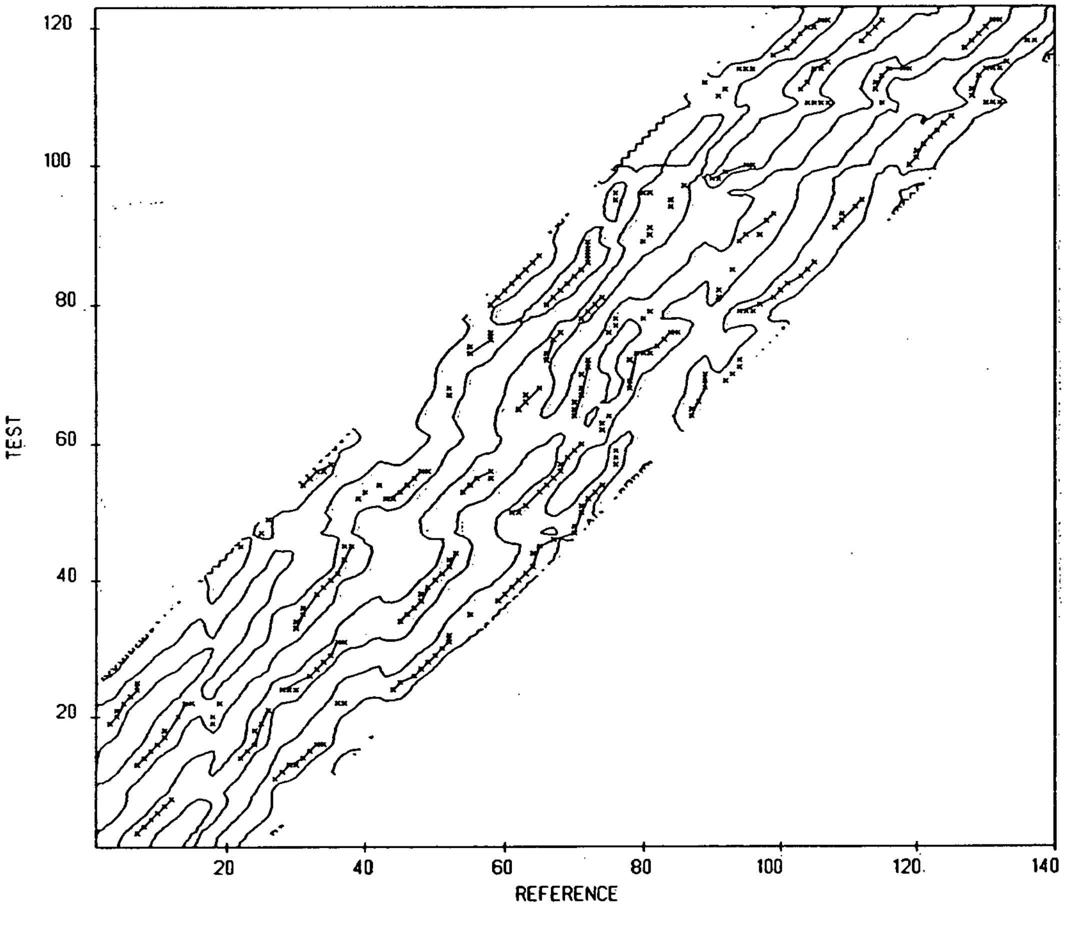

Figure 3.7 The connection of ridge points into subpaths with dynamic matching. Ridge points were connected using $D = 3.5$ and $L = 4.2$.

3.6 The Function Network (Multiple Dynamic Matching)

The set of subpaths found by dynamic matching must now be combined into a set of complete matching functions or 'function network'. Since confidence in a match is low between subpaths, linear approximations are made to join the subpaths into complete functions. To determine which subpaths to join, only those which when joined preserve a non-decreasing function are considered. Joining a subpath with its nearest non-decreasing neighbor may be too restrictive, so the nearest two subpaths are included as possible. The algorithm to build complete functions from subpaths is nontrivial when two choices are allowed at the end of every subpath.

The approach taken to build the function network is to view each subpath as a state and allow two possible transitions from each state. A transition network is established in this manner, where in the case of subpaths, the two next nearest subpaths are the transitions allowed from each subpath. The subpath nearest the origin starts the function building process. A new function is added for each branch from a subpath, and has attributed to it, all the subpaths of the function built up to that branch. Paths are terminated with a subpath that has no possible next subpaths. To ensure that all paths have been found, all subpaths are tested for their membership in a complete path, and any subpath that is not part of a complete path becomes the starting subpath for a new set of paths. An estimate of the total number of paths is discussed in section 3.9. Figure 3.8 shows the function network for the set of subpaths shown in Figure 3.7.

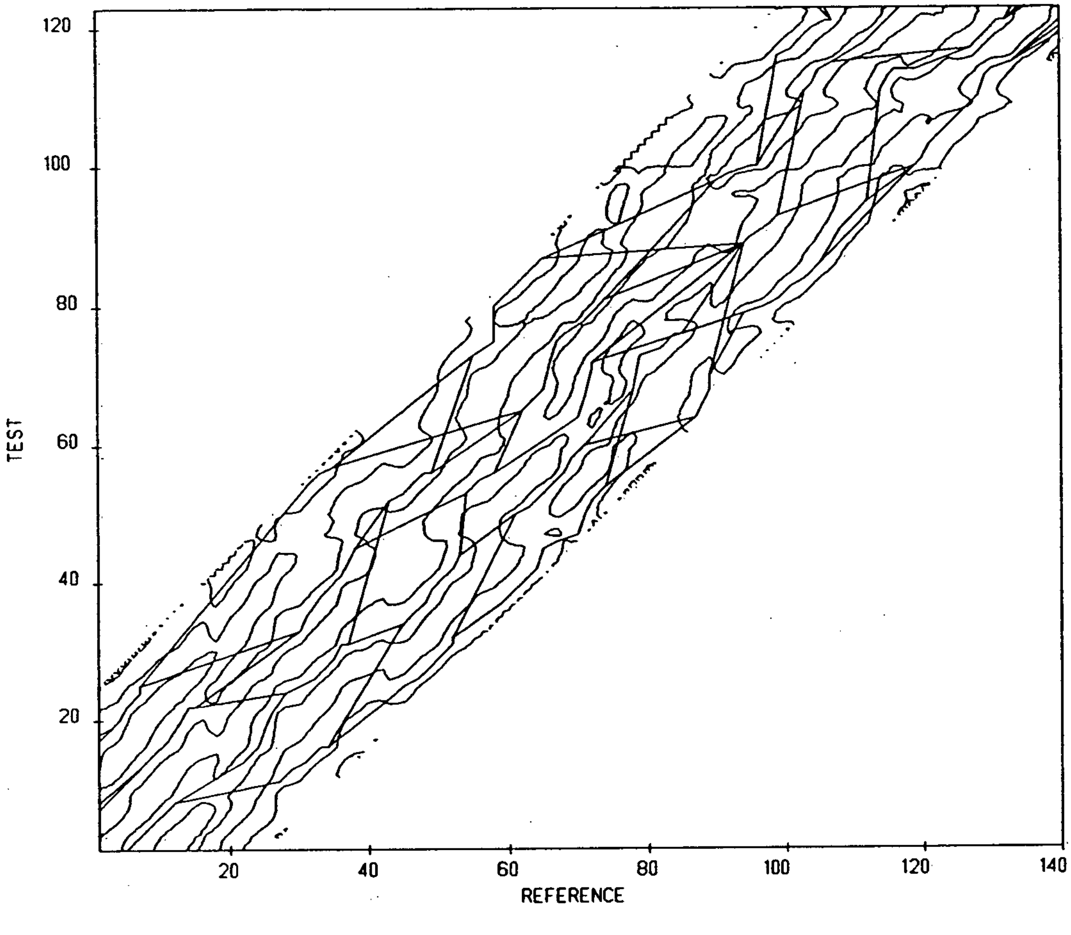

Figure 3.8 The function network found from the subpaths in Figure 3.7. Each subpath may branch to the next two nearest subpaths. There are a total of 180 functions.

3.7 Ranking the Functions

The set of possible matching functions found from multiple dynamic matching must now be ranked to determine the optimum matching function.

3.7.1 Mean Local Confidence

It was argued in section 3.3 that significant trends in correlation appear as well defined ridges in the similarity matrix. The concept of local confidence was developed to measure the quality of the local matches, and now that a set of complete functions has been found, the mean local confidence (MLC) norm may be computed to measure the quality of each entire matching function. Equation (3.4) is restated here.

$$\mathrm{MLC} = -\frac{1}{P}\sum_{p=0}^{P} S_{x,m(x)} S''_{x,m(x)^{\perp}}.$$

It requires computing $S_{x,m(x)}$ and $-S''_{x,m(x)^{\perp}}$ along each function $m(x)$ which must be stepped along at a fixed increment to avoid biasing. Since this will generally lead to coordinates not on a lattice point (i,j) of the matrix, interpolation is necessary. Simple distance-weighted bilinear interpolation is employed to compute $S_{x,m(x)}$ and $-S''_{x,m(x)^{\perp}}$. The MLC norm thus requires interpolation of values at the coordinates $(x, m(x))$, $(x - \Delta x, m(x - \Delta x) + \Delta y)$, and $(x + \Delta x, m(x + \Delta x) - \Delta y)$, where $(\Delta x^2 + \Delta y^2)^{1/2} = 1$. A three point finite difference is then used to compute the second derivative part of local confidence as:

$$-S''_{x,m(x)^{\perp}} = S_{x-\Delta x, m(x-\Delta x)+\Delta y} - 2 \cdot S_{x,m(x)} + S_{x+\Delta x, m(x+\Delta x)-\Delta y} \qquad (3.5)$$

All matching functions are now ranked by their MLC's to determine the best functions. Figure 3.9 shows the top ten functions based on MLC and the top ranking function as a thickened line. Figure 3.10 shows the results of applying the top three matching functions to the test log. Due to the relatively high amplitude distortion $a(x)$ between well logs, additional constraints are helpful to determine the interpreter's optimal $m(x)$.

3.7.2 Simplicity

Other properties beside the mean local confidence are important in determining the optimum matching function. The goal in most problems is to find the *simplest* solution. In the case of matching well logs the simplest structural change that will produce optimal matching is a reasonable requirement.

A measure of simplicity which is easy to compute is the ratio of linear path length to total path length. The linear path length may be easily approximated by the distance from the first to last points in the matching function. The simplicity (SIMP) norm for the kth matching function with ridge points $j = 1, M$ is:

$$\mathrm{SIMP} = \frac{\left[(x_M^k - x_1^k)^2 + (m_k(x_M^k) - m_k(x_1^k))^2\right]}{\sum_{j=2}^{M}\left[(x_j^k - x_{j-1}^k)^2 + (m_k(x_j^k) - m_k(x_{j-1}^k))^2\right]^{1/2}} \tag{3.6}$$

The top 10 functions as ranked by MLC are then ranked by SIMP. The ten matching functions after this ranking are shown Figure 3.11, and the top three are shown applied to the test log in Figure 3.12.

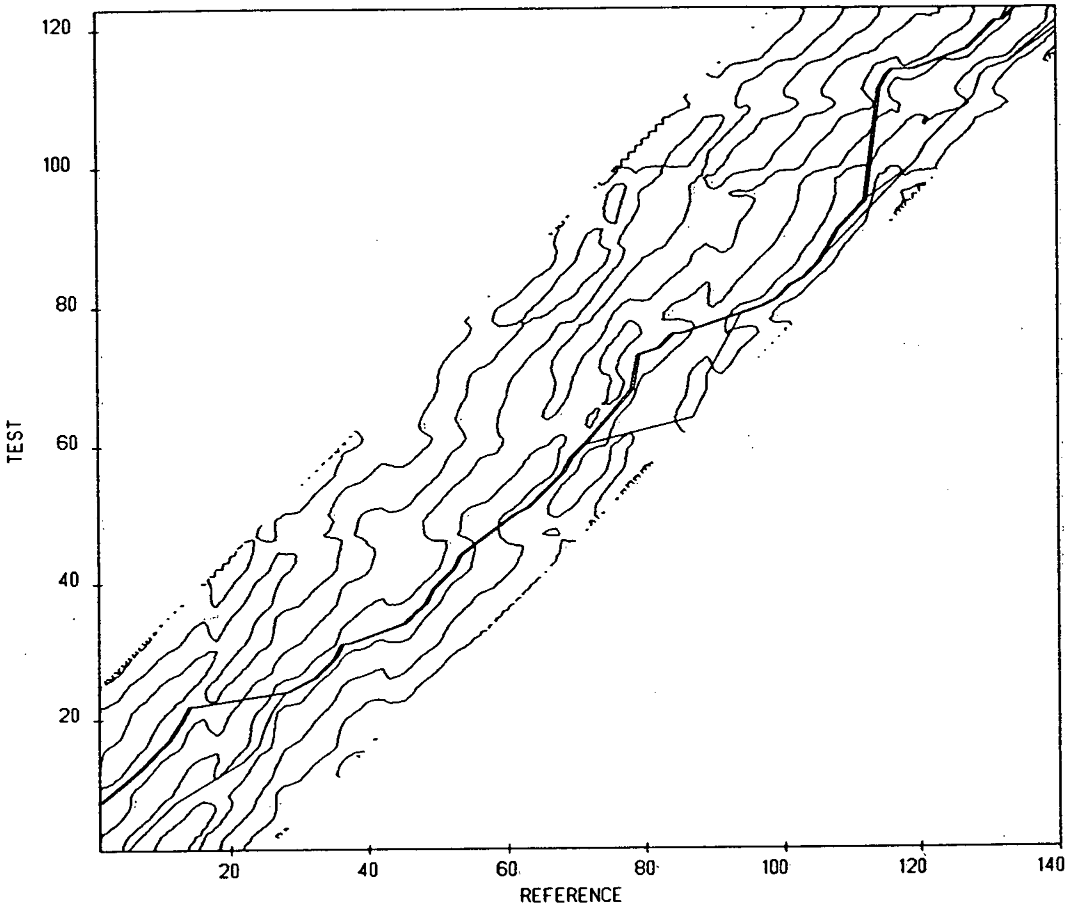

Figure 3.9 The top ten functions based on MLC with the top ranking function thickened.

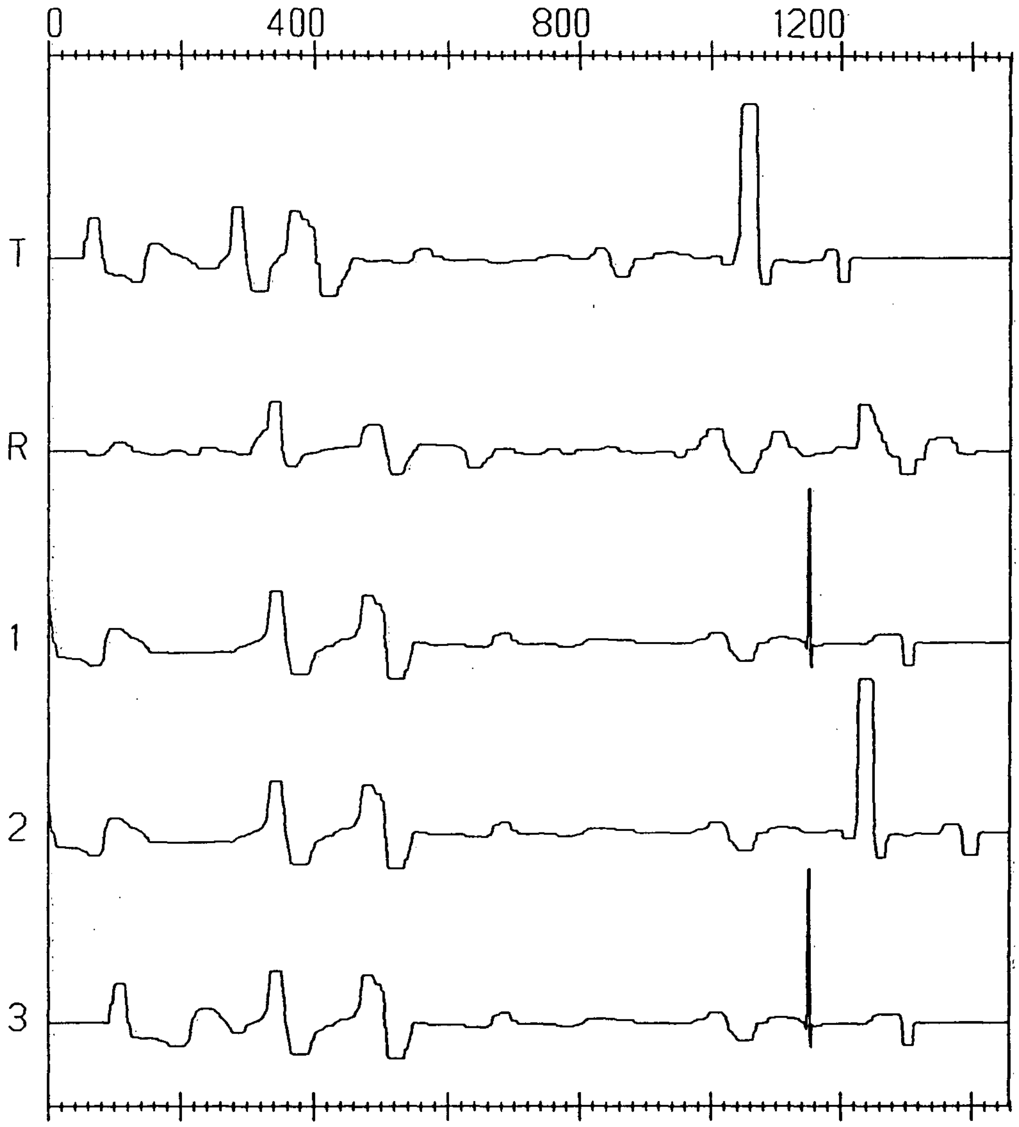

Figure 3.10 The results of applying the top 3 matching functions to the test log. r=reference, t=test, 1,2,3=matching function ranking.

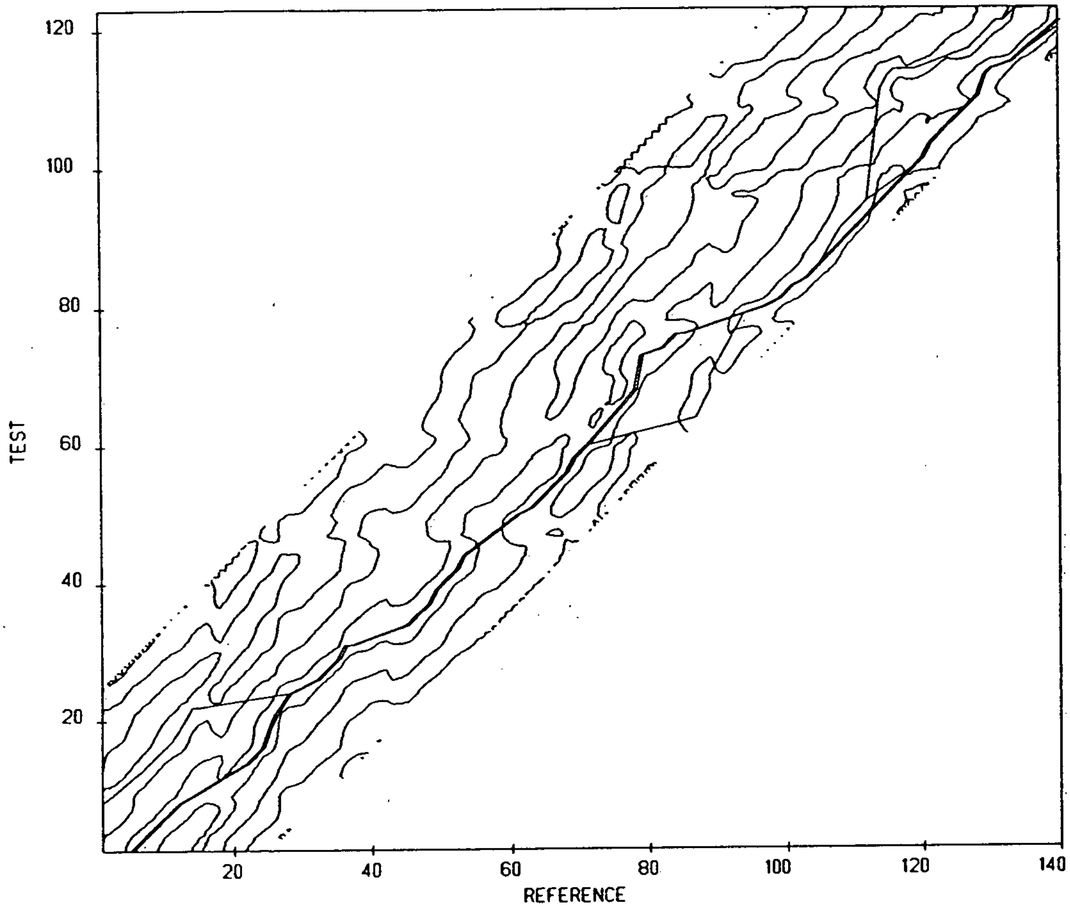

Figure 3.11 The top ten functions after ranking by MLC and SIMP. The top ranking function is shown thickened. Compare with Figure 3.9.

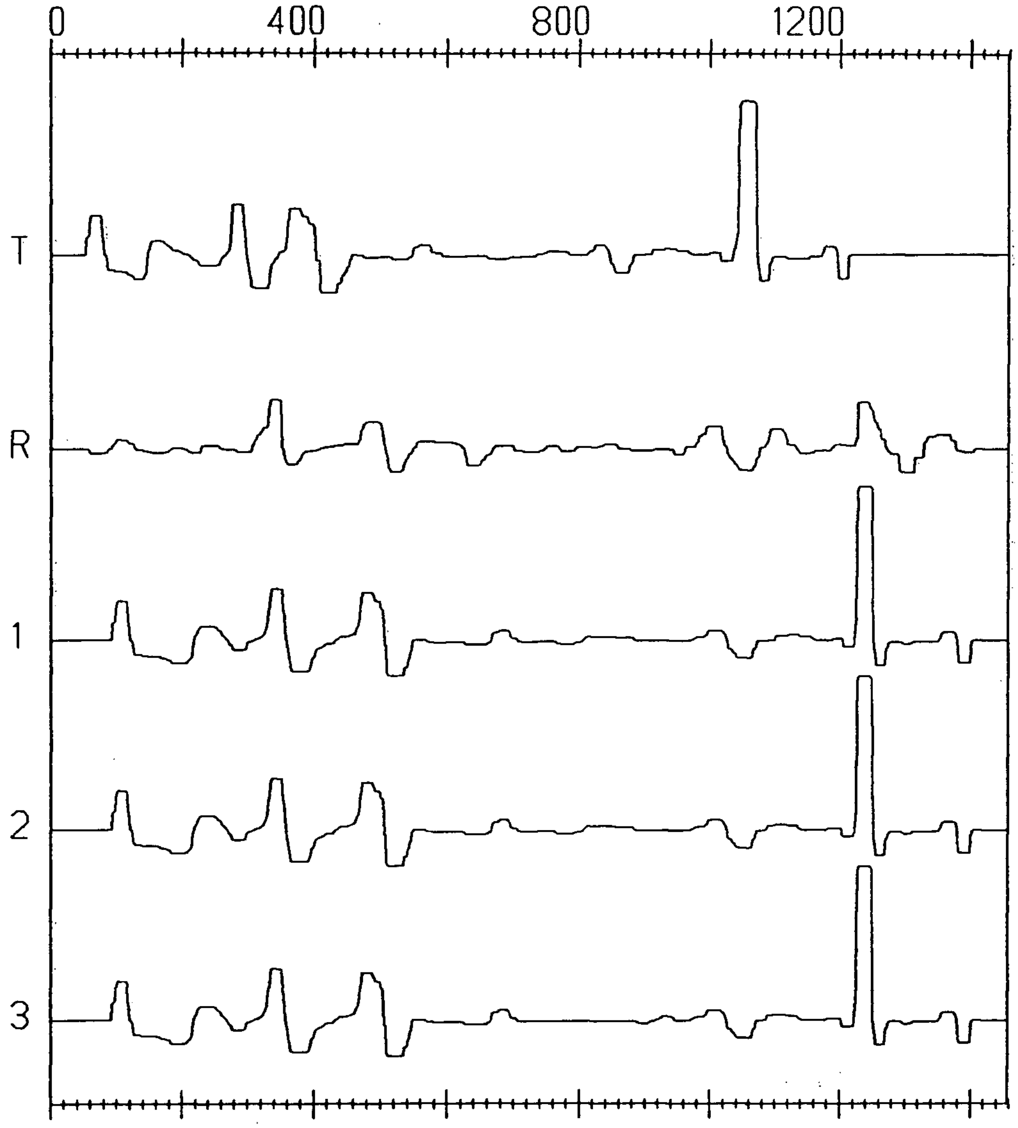

Figure 3.12 The top three functions after MLC and SIMP ranking applied to the test log. Compare with figure 3.10.

3.7.3. Sufficient Diversity

Supplying alternative interpretations requires that they be sufficiently different from one another if they are to provide useful information. Two of the matching functions in the top three are usually so similar that another more diverse interpretation is desired. To ensure the sufficient diversity of the optimal matching functions, some measure of difference between them should be exceeded. A measure of difference between the best ten matching functions as ranked by MLC and SIMP would be:

$$D_{ij} = \sum_{x=0}^{X} |m_i(x) - m_j(x)| \qquad \text{for} \quad i = 1,10 \; ; \; j = i+1,10. \tag{3.7}$$

Cluster analysis may be done using this dissimilarity matrix to separate similar matching functions from sufficiently diverse ones. The top ranking functions from each of the clusters may then be taken to form the set of best three sufficiently diverse functions.

Instead of implementing cluster analysis, the half-matrix of equation 3.4 was computed for the top ten functions in example $m_{2/1}$, and new functions were brought into the top three if any two showed $D_{ij} < 2.0$. The result of implementing this method of sufficient diversity is shown in Figure 3.13. The three functions now provide the interpreter with more information.

3.8 Applying the Matching Function

Once the best functions have been found, a visual check as to the quality of the matches is useful. Figures 3.10, 3.12 and 3.13 were produced by applying the top three matching functions (after different rankings) to the test log. The application of a matching function requires the stretching or compressing (warping) of the log axis as

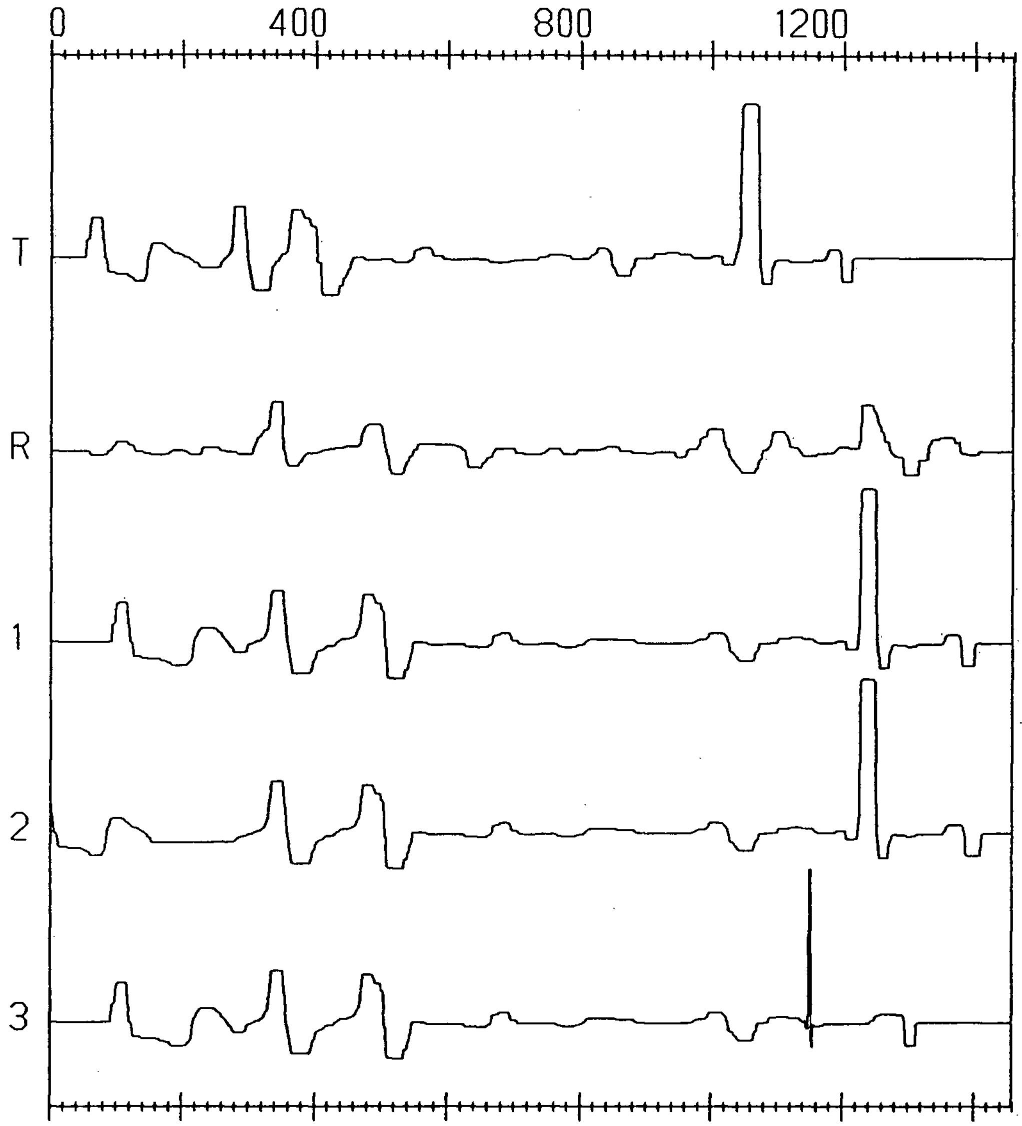

Figure 3.13 The best three functions applied to the test log after they are constrained to be sufficiently diverse.

determined by a segment of the piecewise linear function. If the local slope exceeds 1.0, the test log is locally compressed, and if the local slope is less than 1.0, the test log is locally stretched. Then the log is resampled to maintain a constant sampling interval. The resampling of the log after warping the signal axis requires the inperpolation of log values. Interpolation has been done for examples in this project using a simple distance - weighted linear scheme. It has been found quite satisfactory given the rather heavy filtering of the sonic data, but for higher resolution matching some other (*ie.* cubic spline) interpolation may be needed.

Another method of display, other than applying the matching function, is to connect matched locations in each log with a dashed line (see Figure 1.2). Graphically this is equivalent to locating a point on the axis of one of the logs and projecting it on to the matching function. This point is then projected on to the other signal axis to determine the matched location. This mode of display is particularily useful when matching structurally equivalent portions of well logs and will be used in section 3.12.

3.9 Computing Time

Factors that affect the computing time necessary for a solution are the similarity matrix bandwidth b, the matrix sampling increment (Δx), and the number of subpaths (n). The matrix bandwidth b is set by the user based on *a priori* information about the complexity of the geology over the separation of the wells. Some practical upper bound is easily estimated. The size of the matrix is related directly to the bandwidth b and inversely to the matrix sampling increment Δx. Since higher resolution means a shorter low-pass median filter length and finer matrix sampling, and since matrix size is proportional to $(b/\Delta x)^2$, the cost of the algorithm rises rapidly with resolution.

The effects of bandwidth and resolution on computing time are easy to quantify. Another cost factor is the total number of matching functions found. This is an important factor of cost since the MLC norm is computed for each function. The total number of functions is related to the number of subpaths. The number of subpaths in turn is a function of the local path constraints and the underlying similarity of the signals being matched. Predicting the number of subpaths is therefore a difficult task.

Given that the number of subpaths is known however, the total number of functions may be roughly estimated. The function building algorithm progresses forward through the similarity matrix, combining subpaths with two others at each 'level' of the network. As levels are passed, all subpaths behind the forward most points in the functions are eliminated from possible combination. The power to which 2 should be raised is thus reduced by the number of subpaths passed at each level. The number of subpaths passed at any level is related to the spatial period of the ridges and the bandwidth of the matrix. Once the number of ridges r across the matrix is estimated, the number of paths for each starting subpath is then estimated by $2^{(n/r-(e-1))}$ where n is the number of subpaths and $e-1$ is subtracted from the power for the subpaths in the ending level. This number is then multiplied by the number of starting subpaths. In example $m_{2/1}$with $n = 30$, $r = 4$, and $e = 3$, the estimate of the total number of paths is 128 and the actual number is 135.

Memory requirements should also be mentioned. The similarity matrix S_{ij} is not fully used since only the central band needs to be stored for pathfinding. A more efficient use of memory would eliminate the unused 'triangles' above and below the central band, making S_{ij} rectangular, but this would require a rotation of the matching function before application. This saving in memory has not been implemented in the interest of preserving clarity in similarity matrix figures.

3.10 Sources of Error

The error in any matching function may only be thought of relative to the resolution or scale of structures left in the logs. A function which is correct at one scale will probably not be correct at another scale. The matching of significant events - the block edges or steps, is the most important aspect of matching and will be considered first.

The error in $m(x)$ for any x is at a minimum when local confidence is at a maximum. This occurs over subpaths where ridge points have been located. As explained in section 3.3, ridge points (and subpaths) are found where events of significant magnitude have been matched, so the matching of major events is done with minimal error over subpaths. The errors in matching over subpaths are due only to the matrix sampling increment. Any significant block edge will be matched to within Δx, but $\pm\Delta x/2$ is a better estimate.

The greatest errors occur where confidence is low. This happens over the linear approximations between subpaths. Over these portions of the function, error in matching is not easily quantified in terms of samples. However, a plot of the local confidence *vs* reference depth can indicate where errors are greatest. Figure 3.14 shows match $m_{2/1}(x)$ and LC *vs* x.

Another source of error should be mentioned also. The degree to which median filtering does not accurately represent the underlying blockiness of the original signal can also induce errors in the match. Matching errors of this type and those due to low local confidence may be reduced by increasing resolution and matching finer scale structures. Using a matching function found from broad scale matching as a constraint on finer scale matching can reduce errors and make matching of small scale structures possible.

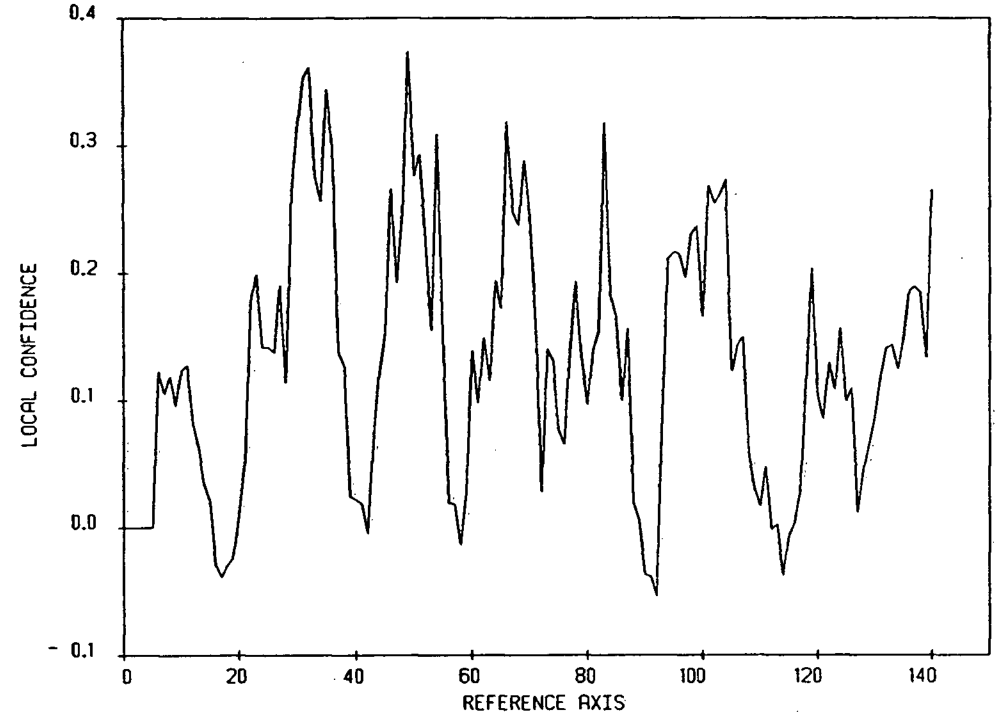

Figure 3.14 Local confidence vs. x for the top ranking matching function in figure 3.12.

3.11 Multiple Dynamic Matching and Pattern Recognition

Exploring the analogy between multiple dynamic matching or 'MDM' and pattern recognition affords insight into how MDM works. At a given scale obvious matches of major events 'jump out' at the interpreter. He/she scans along the signals thinking something like 'this matches with this and that matches with that...' etc. This pattern matching process is represented in the similarity matrix by coherent ridge trends where each ridge represents a distinct local interpretation. When the interpreter's thought becomes 'or that could match with that', this is equivalent to skipping to a new ridge in the similarity matrix, and another interpretation is included as possible. MDM shows this doubt in interpretation as a branch in the 'interpretation tree' of most likely matching functions (see Figure 3.8). All the most probable interpretations are found automatically by this process of connecting local interpretations together. Finally the MLC and SIMP norms mimic the confidence criteria of the expert and the best match is chosen.

The analogy may be continued through scale as well. Just as the interpreter matches 'marker' events first and then looks for subtleties, so may MDM be able to 'fine tune' a broad scale match in subsequent processing. High resolution matching down to a few samples should be possible using a 'scale down' procedure where the $m(x)$ at one scale is used as a constraint on the $m(x)$ at the next scale down, effectively restricting the bandwidth of S_{ij} and so also the possible change in $m(x)$.

3.12 Automatic Stratigraphic Correlation

The major objective of matching well logs is to ascertain an estimate of the geologic change between wells. This means estimating the local dip or structure through the correlation of equivalent time horizons or events, and estimating the change in the measured response. The local dip or structure is given, as is discussed in section 1.2 and 1.5, by $m(x)$, while the change in the measured response is given by $a(x)$. Since the matching function quantifies the correlation between signals, it may be used to connect structurally equivalent depths in two wells. To illustrate this connection, matching functions were found between the five sonic well logs in Figure 2.12. The method of automatic event detection discussed in section 2.5 was implemented to find the major events in sonic log 1. Figure 3.15 shows these major events followed between wells using the matching functions for logs 1 and 2, 2 and 3, 3 and 4, 4 and 5. Each of the matching functions used was top ranked based on the MLC and SIMP norms, and since event detection was also done automatically, the resulting structural interpretation is totally objective.

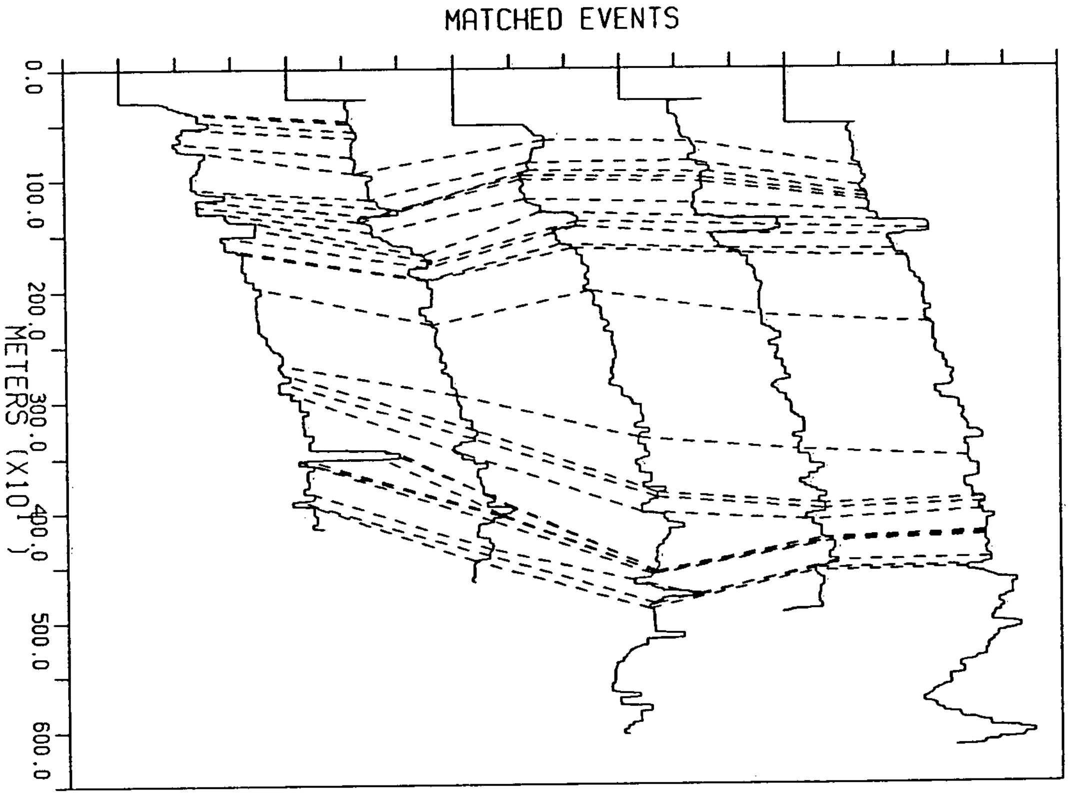

Figure 3.15 The automatic correlation of five well logs from four matching functions. Major events were found in log 1 and followed across the section. Equal spacing of the logs is arbitrary.

Chapter 4

CONCLUSIONS

4.1 Improvements

The method of multiple dynamic matching has been shown to be successful in the problem of matching trend - removed and median filtered sonic well log data. Nevertheless some improvements are needed to increase matching resolution and to further generalize the method. As discussed in section 3.9, the error in the match is greatest where linear matches are assumed between subpaths. Reducing the errors in the matching function requires higher resolution matching, but as discussed in section 3.9, this causes a rapid increase in the cost of the algorithm. One possibility that should allow high resolution matching at reasonable efficiency is that of matching through scale. The idea is to obtain a broad scale matching function, apply the function to less severely filtered versions of the logs and proceed to determine the higher resolution matching function. Since the broad scale estimate of the function has been applied to the data, it has been mapped to the diagonal of the next higher resolution similarity matrix which is now square, and the matrix bandwidth may be reduced dramatically since large scale structural changes have been removed. Thus matching at higher resolution may be done more efficiently due to the 'smaller' matrix. The matching function at the next resolution may be viewed in terms of its deviations from the diagonal, and once the previous warping of the test signal axis has been accounted for, the deviations may be added to the broadest scale function. Resolution in the matching function may be iteratively built up by this method, and with the bandwidth of the similarity matrix reduced at each iteration, the cost is kept at a minimum.

There is a problem with this approach however. The broad scale match has been obtained from filtered versions of the logs that have flat regions (constant neighborhoods). Sections of severe stretching and compressing in the matching function when applied to a flat region of the test log, do not affect the quality of the match over these regions. Unfortunately, severe slopes in a matching function when applied to a less filtered test log will produce pronounced and unreasonable distortions in the signal, making it unsuitable for further matching. Thus the application of a broad scale matching function may not be appropriate before higher resolution matching.

Two possible alternatives may help this situation and allow for iteratively matching through scale. The first involves smoothing the broader scale function before application, reducing the severe slopes and so also any unreasonable distortions in the less filtered logs. The second alternative does not apply the broader scale mapping at all, but instead, a narrow band of a higher resolution similarity matrix would be computed around the already determined broad scale function. The next matching function is thus restricted to be close to the broad scale function. This approach complicates the computation of S_{ij} but may prove superior to smoothing the matching function before its application. It is believed that higher resolution matching is possible with either of the approaches discussed above, but they must be tried and assessed based on the usual criteria of accuracy and efficiency.

4.2 Parameter Estimation

The method of multiple dynamic matching requires the fixing of the parameters N (half the low-pass median filter length), K (the similarity window length), and D (the proximity threshold for subpath determination). A rough estimate of K has been established (given that $\Delta x \approx 2/3N$) as $\approx 4N$ or $6\Delta x$, as this provides a sufficiently smooth similarity matrix with the proper character. The median filter length $(2N+1)$ determines the resolution of the match and is selected by the user (as is the matrix bandwidth b), but the proper choice for D is difficult to establish given the other parameters. The proximity threshold D is the maximum allowable separation of ridge points in any subpath, and although a value of 3.5 (or 35 log samples) has worked well for the matching examples in this project and tests have indicated that broader scale matching using $N = 29$, $\Delta x = 20$ $K = 120$ and $D = 35$ have produced similarily good results, the value of D in other matching problems will likely have to be altered to suit the particular application. The generality of the method thus breaks down if the local path constraint (D) cannot be determined automatically in every application. Where one subpath becomes two is a crucial question in multiple dynamic matching. The method requires enough subpaths to fully explore the space of possible matching functions, but not too many to make the solution prohibitively costly. Intuitively it seems that there should be a range of values for D for which the number of subpaths should remain constant. This kind of testing could be done to establish D for matching. The minimum subpath length L has been fixed at 42 log samples and this has also proven to be a good value at different scales. But the fixing of D, L and K will necessarily depend on the particular application and perhaps most strongly on the bandwidth and nature (eg. blocky or continuous) of the input signals. The automatic fixing of these parameters and ultimate generalization of MDM requires experiments with other data sets and other matching problems.

4.3 Speculations on the Future and Related Ideas

4.3.1 Extending Well Log Matching to 3D

The structural interpetation of Figure 3.15 shows the logs arbitrarily spaced an equal distance appart. In reality they are distributed as shown in Figure 4.1. It is evident that only 4 matching functions have been found and that there are still 6 remaining to be determined since there are a total of 10 possible matches. Figure 4.1b shows the 'net' of matching combinations possible with this areal distribution of five wells. The intersection points indicate locations where more control is possible on the structural interpretation. For each major event a surface may be approximated allowing errors at the intersection points. In this way an objective three dimensional interpretation is possible using only well log information. More complex geologies would require more sophisticated fitting schemes, and due to nonuniqueness, more than one interpretation should be generated, but each interpretation may be ranked based on optimality criteria, thus giving the interpreter an objective basis with which to compare his own interpretation.

Once a three dimensional structural interpretation has been settled upon, by whatever method, the spatial variation of the measured response, (*ie.* acoustic velocity), may be estimated. The determination of matching functions for a set of well logs can therefore automatically estimate the structural and response change in an area.

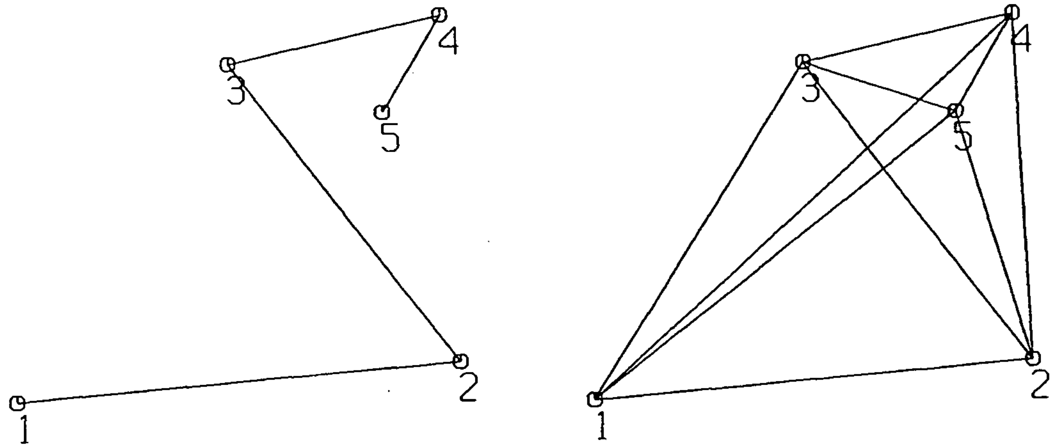

Figure 4.1 (a) The areal distribution of wells and the matches corresponding to those in Figure 3.14. (b) The matching 'net' possible with the five wells. Intersection points indicate where more control is possible for the mapping of structural events over an area.

4.3.2 Seismic Signal Matching

In seismology, the importance of signal correlation was recognized early on, especially for the determination of velocity functions. Indeed, the interpretation of seismic sections depends on the spatial correlation of events at different times. The spatial - temporal correlation function implicitly determined by the interpreter approximates the geology, so any artificial intelligence system that attempts to automatically interpret a seismic section must first determine the correlation functions between traces.

The matching of seismic traces is a considerably easier problem than matching well logs, owing to the closer proximity of adjacent measurements and the lower frequency content of the signals. These factors mean that the matching function is constrained to be quite simple. In the context of multiple dynamic matching this means a narrower similarity matrix and efficient high resolution matching. The multiple interpretation advantage of MDM could be effectively used in seismic interpretation, since locations where a second or third interpretation is probable may be flagged and further investigation focussed on that area, perhaps allowing the computer to continue the interpretation with the second best choice.

Nonlinear signal matching thus provides automatic picking of events across a section. As a bonus, storing adjacent matching functions gives the spatial - temporal correlation function, making improved trace interpolation possible. Also, multichannel deconvolution, dip moveout filtering, and automatic velocity analysis can be approached from a new and different point of view.

4.4 Summary

The nonlinear signal matching problem has been introduced by contrasting it with the simple idea of cross correlation, and the discrete lag in the cross correlation expression was replaced with a continuously varying lag or matching function, $m(x)$. The matching problem was further explained by a model of change between waveforms that decomposes the problem into a structural change $m(x)$, and an amplitude distortion $a(x)$. It was argued that in the presence of large amplitude distortions, the correct matching function may appear suboptimal under similarity norms and as large amplitude distortions are likely to occur in any difficult matching problem, a method is needed that can provide suboptimal as well as optimal solutions.

Two distinct methods for the solution of the nonlinear signal matching problem were reviewed, namely the dynamic programming approach and the inverse theory approach. It was found that both methods have advantages and pitfalls, and that neither method is satisfactory when the goal is to objectively determine optimal and suboptimal solutions.

Multiple dynamic matching was introduced as a method that overcomes the pitfalls of both of the other methods. It is founded on a dynamic programming philosophy in that pathfinding through a similarity matrix is done, but with the innovation of a significance threshold to extract only the matches where confidence is high. Each of these 'subpaths' is then joined in combination with others to generate a network of possible complete interpretations. The advantage of the inverse approach is that optimality is based on global rather than local norms, and this feature has been incorporated into MDM in that the objective function to be maximized measures the quality of the entire matching function. Due to the large amplitude distortions between well logs the correct or interpreter's matching function may be ranked as suboptimal, so a simplicity constraint can be invoked to improve the ranking of the 'interpeter's choice'. The results of MDM are good for the example explored here, and given the difficulty of the well log matching problem, the method should perform well in other simpler matching problems.

It was also conjectured that higher resolution matching with reduced error would be possible by matching through scale, where a broader scale match is used as a constraint on finer scale matching. The 'scale space' approach is receiving considerable interest in the artificial intelligence fields of computer vision and pattern recognition, and since signal matching is essentially a pattern recognition process, matching through scale is a promising prospect to improve the results of multiple dynamic matching. In fact the idea of scale space was discussed in chapter two, and it was argued that the median is a superior parameter of scale than that employed by Witkin (1985), especially when looking at blocky waveforms. Median scale space and the new signal decomposition presented for the first time in section 2.4, median decomposition may prove to be suitable for the description of other intricate waveforms as well.

Although the subject of median filtering was secondary in the present work, it represents, as does multiple dynamic matching, the need for new nonlinear approaches to solve difficult problems in signal processing.

Biblography

Anderson K.R. and Gaby J.E., 1983, Dynamic waveform matching, Information Sciences,**31**, p221-242.

Bednar T.B., 1983, Applications of median filtering to deconvolution, pulse estimation, and statistical editing of seismic data, Geophysics **48**, No. 12, p1598-1610.

Claerbout J.F. and Muir F., 1973, Robust modelling with erratic data, Geophysics, **38**, p826.

Clarke, R.M., 1985, A fortran program for constrained sequence slotting based on minimum combined path length, Computers and Geosciences, **11**, No. 3, p605-617.

Bellman R., and Dreyfus S., 1962, Applied dynamic programming, Princeton U.P., Princeton N.J.

Douze E.J. and Laster S.J., 1979, Statistics of semblance, Geophysics, **44**, No. 12, p1999-2003.

Gordon A.D. and Reyment R.A., 1979, Slotting of borehole sequences, Journal of Mathematical Geology, **11**, No. 3, p309-327.

Evans J.R., 1982, Running median filters and a general despiker, Bulletin of the Seismological Society of America, **72**, No. 1, p331-338.

Fitch J.P., Coyle E.J. and Gallgher N.C., 1984, Median filtering by threshold decomposition, IEEE transactions on Acoustics, Speech and Signal Processing, ASSP-**32**, No. 6, p1183,1188.

Lanning E.N. and Johnson D.M., 1983, Automated identification of rock boundaries: An application of the Walsh transform to geophysical well log analysis, Geophysics, **48**, No. 2, p197-205.

Liu H.H. and Fu K.S., 1982, A syntactic pattern recognition approach to seismic discrimination, Geoexploration, **20**, p183-196.

Martinson D.G., Menke W. and Stoffa P., 1982, An inverse approach to signal correlation, Journal of Geophysical Research, **87**, No. B6, p4807-4818.

Myers C.S. and Rabiner L.R., 1981, A comparitive study of several dynamic time-warping algorithms for connected word recognition, Bell system technical journal, **60**, No. 7, p1389-1409.

Neidell N.S. and Taner M.T., 1971, Semblance and other coherency measures for multichannel data, Geophysics **36**, p482-497.

Rabiner L.R., Rosinberg A.E. and Levinson S.E., 1978, Considerations in dynamic time warping of discrete word recognition, IEEE transactions on Acoustics, Speech and Signal Processing, ASSP-**26**, No. 6, p575-582.

Robinson J.E., 1978, Pitfalls in automatic lithostratigraphic correlation, Computers and Geosciences, **4**, p273-275.

Schwarzacher W., 1980, Models for the study of stratigraphic correlation, Mathematical Geology, **12**, No. 3, p213-234.

Smith T.F. and Waterman M.S., 1980, New stratigraphic correlation techniques, Journal of Geology, **88**, p451-457.

Stewart R.R., 1985, Median filtering: review and a new F/K analogue design, Journal of the Canadian Society of Exploration Geophysicists, **21**, No. 1, p54-63.

Witkin A.P., 1984, Scale space filtering: a new approach to multi-sscale description, in Image Understanding, Norwood N.J., Ablex Publishing Corporation.

Appendix

The pages that follow contain the similarity matrices and ten best matching functions after ranking by MLC. The ranking of the chosen function is given in the captions of the 'b' figures for each pair of logs. If the top 10 functions are ranked also by the SIMP norm, the chosen interpretation is then ranked as number 1. The chosen functions have been used to generate the structural cross section of Figure 3.15.

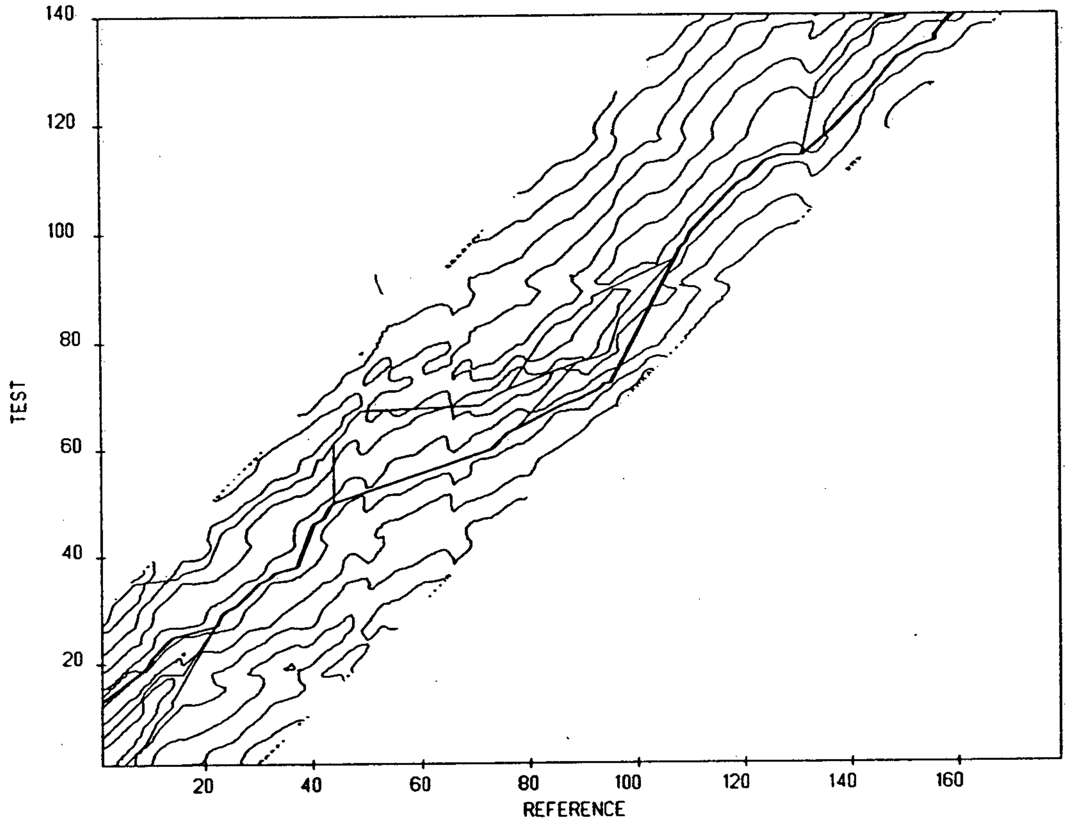

Figure A1.a Similarity matrix for reference = log 3 and test = log 2, and the best 10 functions after ranking by MLC. The top ranking function is shown as a thickened line.

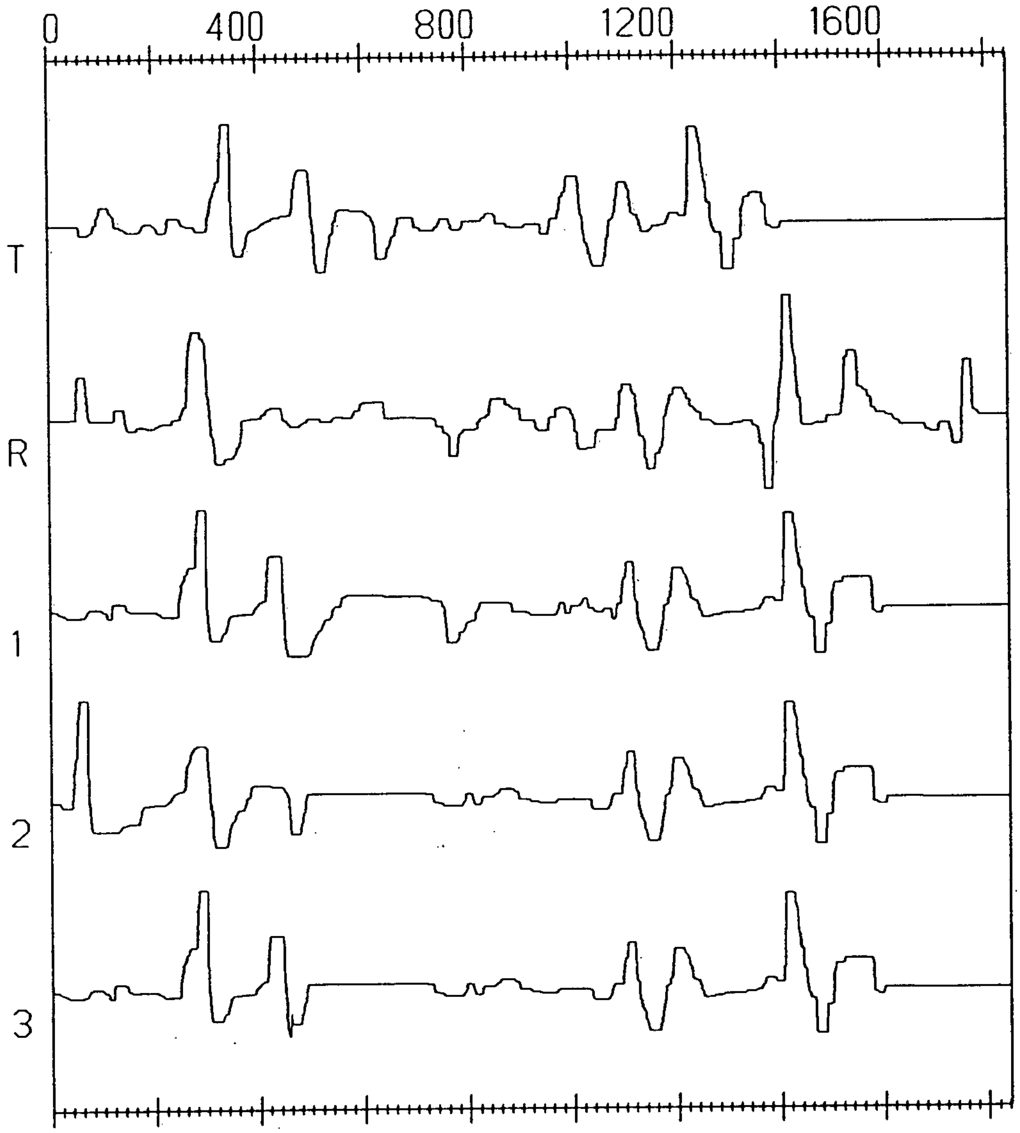

Figure A1.b The top three functions after MLC diversity applied to the test log. The number 2 ranking function was chosen.

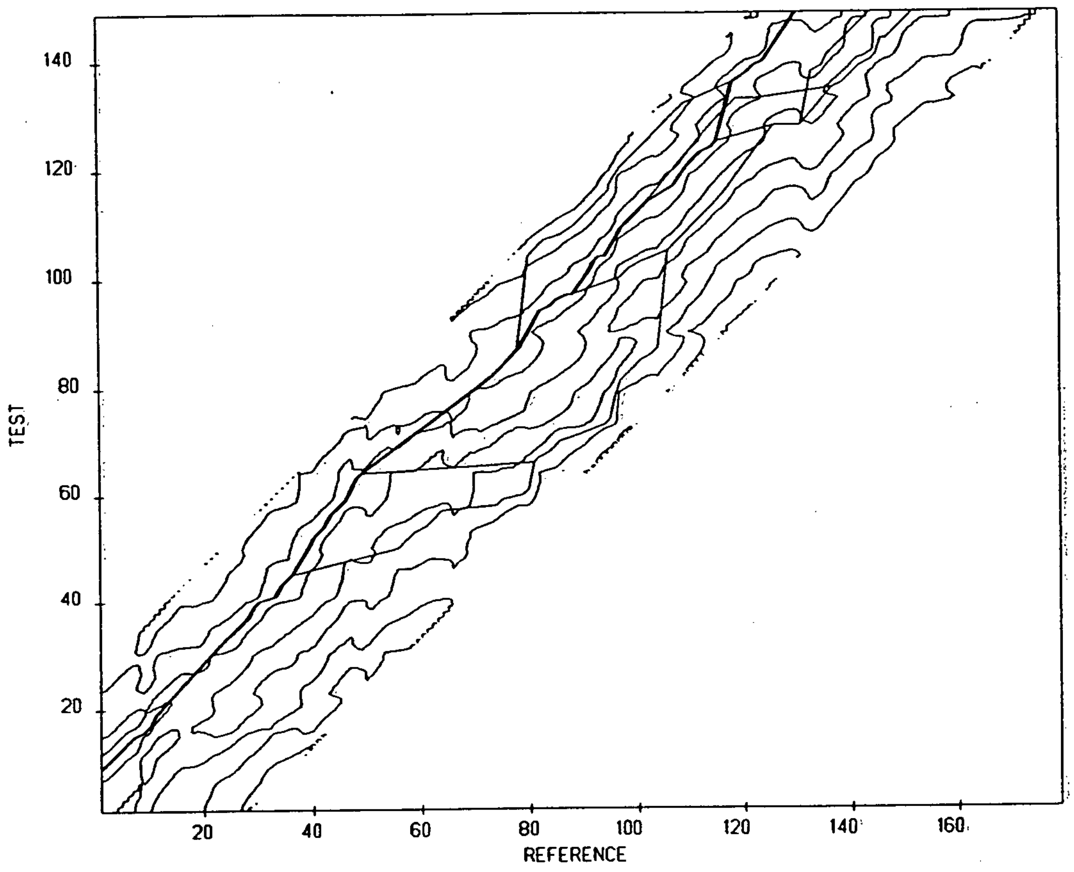

Figure A2.a Similarity matrix for reference = log 3 and test = log 4, and the best 10 percent of functions after ranking by MLC. The top ranking function is shown as a thickened line.

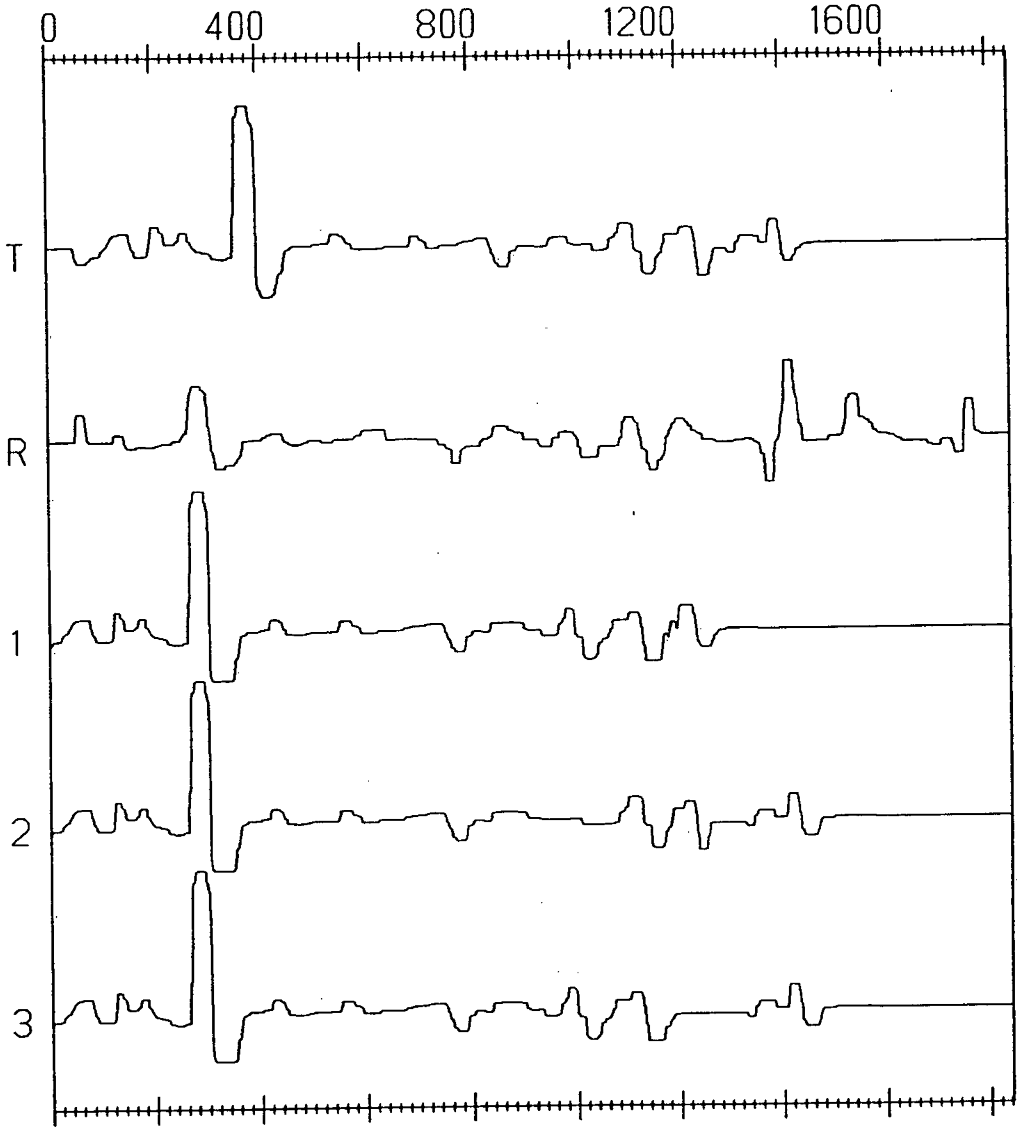

Figure A2.b The top three functions after MLC applied to the test log. The number 1 ranking function was chosen.

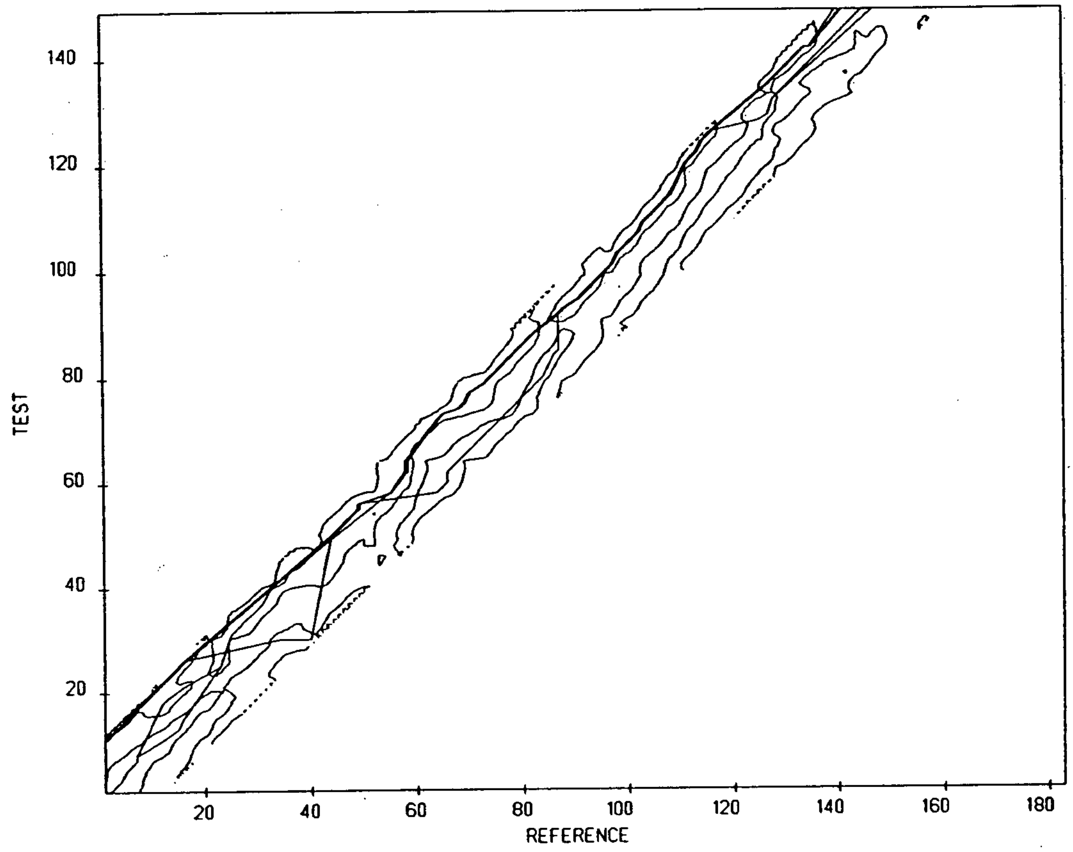

Figure A3.a Similarity matrix for reference = log 5 and test = log 4, and the best 10 percent of functions after ranking by MLC. The top ranking function is shown as a thickened line. The matrix bandwidth is much narrower due to the close proximity of the wells.

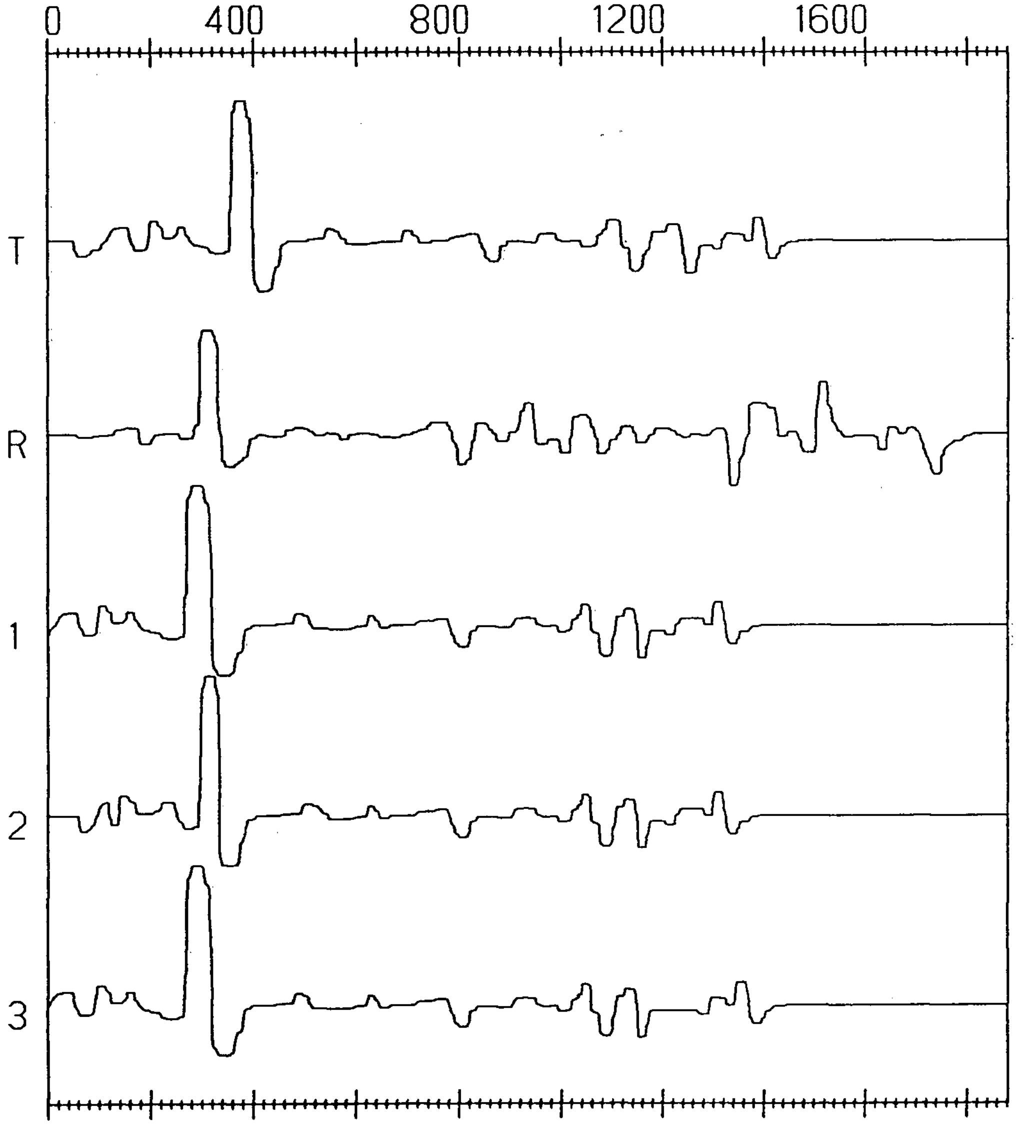

Figure A3.b The top three functions after MLC applied to the test log. The number 2 ranking function was chosen.