Smooth dynamic warping

Dave Hale and Stefan Compton
Center for Wave Phenomena, Colorado School of Mines, Golden CO 80401, USA

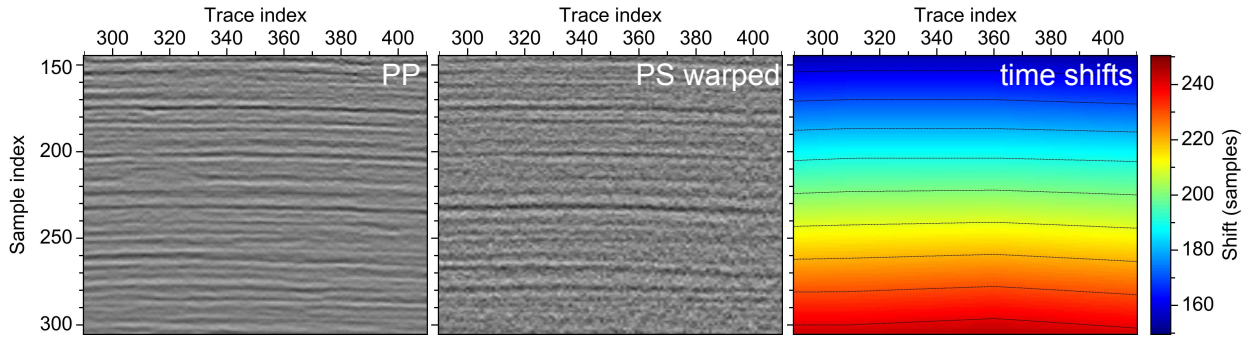

Figure 1. Subsets of a PP image and a corresponding PS image warped using time shifts estimated automatically by smooth dynamic warping. Contour lines (at 10-sample intervals) highlight the smoothness of the estimated time shifts.

ABSTRACT
Dynamic time warping is a simple classic method for aligning two sampled functions of time. The sequence of time shifts computed by this method are a globally optimal solution to a non-linear optimization problem with linear inequality constraints that may be directly related to subsurface properties. In applications to seismic traces, the time shifts may increase or decrease rapidly in time (and space). However, when related to integrals of subsurface parameters, such variations in time shifts are often smooth. A new method for smooth dynamic warping exploits this inherent smoothness to increase the accuracy of time shift estimates, especially where differences between seismic traces to be aligned are not limited to time shifts. The new method requires only a few simple modifications to the classic method, and can be easily extended to multidimensional image warping as well. In an application to registration of PP and PS images, we used smooth dynamic warping to compute times shifts that vary smoothly in both time and space, and are significantly more accurate than those computed using the classic method.

Key words: seismic image dynamic warping

1 INTRODUCTION

Dynamic image warping is a method for shifting features in one image to align them with corresponding features in another image. In applications to seismic images, the shifts are often in time or depth, and the correspondence is rarely exact.

Figure 1 illustrates one example, the registration of PP and PS seismic images. Here we have warped a PS image to align reflections with those in a PP image. The time shifts, while increasing rapidly with time, vary smoothly in time and space. We computed these time shifts using a new method for *smooth dynamic warping* proposed in this paper.

Clearly, differences between the PP and PS images in Figure 1 are not limited to time shifts. Other differences include noise, which is most apparent in the PS image, as well as differences in reflection amplitudes and waveforms related to differences in PP and PS reflection coefficients.

The PP and warped PS images shown in Figure 1 are subsets of the PP and (not yet warped) PS images shown in Figure 2. The images are displayed with different vertical scales so that reflections in the PP image are visually and approximately aligned with corresponding reflections in the PS image. The time shifts shown in Figure 1 are a subset of the time shifts we computed from the entire PP and PS images shown in Figures 2a and 2c.

Again, differences in noise and reflection amplitudes and waveforms are apparent in the PP and PS images. We compensated somewhat for amplitude differences by applying a time-variable gain to each trace in the PP and PS images. This gain normalizes amplitudes within seamlessly overlapping windows so that the rms amplitude within each window is one. Because the windows approximate a Gaussian with half-width 100 samples, the applied gain varies slowly with time.

Remaining differences complicate our estimation of time shifts. Yet we often estimate time differences, as in this example, because (1) they are related to subsurface properties, such as the ratio of P-wave and S-wave velocities V_P/V_S, and (2) after compensating for differences in reflection times, we can more readily analyze differences in reflection waveforms. So it is important that we estimate time shifts using methods that are accurate in the presence of noise and other such differences.

Several authors, including Gaiser (1996), Fomel et al. (2003), Nickel and Sonneland (2004), Fomel et al. (2005), and Liang and Hale (2012) have described methods for registration of PP and PS images and the corresponding estimation of V_P/V_S ratios. For this and similar problems, the dynamic warping method (Sakoe and Chiba, 1978; Anderson and Gaby, 1983; Hale, 2013) has two important advantages. First, dynamic warping honors specified bounds on time shifts and on the rate at which time shifts may vary with time. Second, using the computational method of dynamic programming, dynamic warping finds a globally optimal solution to an error minimization problem that may have many local minima.

The first advantage is important because the constraints are often related to geophysical parameters. For example, because $V_P \geq V_S$, we know that PS reflection time is never less than the corresponding PP reflection time, and that the differences between PS and PP reflection times can never decrease with time.

The second advantage improves the accuracy of dynamic warping in the presence of noise and reflection waveform differences. Such differences, which are unrelated to time shifts, imply that we often want a locally sub-optimal but globally optimal image registration.

In this paper we propose an improved method for *smooth dynamic warping* that enables control of a trade-off between accuracy and resolution of estimated *time strains*, changes in time shifts with time. The trade-off is important because estimates of time strain may correspond directly to subsurface parameters, such as V_P/V_S ratios. In addition, this new warping method improves robustness in the presence of noise and other differences, while requiring significantly less computer memory in applications to 2D and 3D images.

We first describe a basic dynamic time warping algorithm that is equivalent to one developed by Sakoe and Chiba (1978), but is most similar to one developed by Hale (2013). We then describe our improved algorithm, and extend the improvement to image warping, in which we compute shifts that are smooth in all sampled image dimensions (time and space), as illustrated in Figure 1. Throughout this paper we use the registration of PP and PS images shown in Figure 2 as just one example of the application of our improved method for smooth dynamic warping. The improvements extend to other applications as well.

2 DYNAMIC TIME WARPING

The problem in dynamic warping is to find a sequence of time shifts $u[0:n_i-1] \equiv \{u[0], u[1], \ldots, u[n_i-1]\}$ that aligns two sequences (time series) f and g so that

$$f[i] \approx g[i+u[i]], \quad i = 0, 1, \ldots, n_i - 1 \tag{1}$$

More precisely, dynamic time warping computes time shifts

$$u[0:n_i-1] \equiv \underset{l[0:n_i-1]}{\arg\min} \sum_{i=0}^{n_i-1} e[i, l[i]] \tag{2}$$

subject to constraints

$$u_l \leq u[i] \leq u_u, \quad r_l \leq u[i] - u[i-1] \leq r_u, \tag{3}$$

where

$$e[i,l] \equiv (f[i] - g[i+l])^2. \tag{4}$$

In the inequality constraints 3, u_l and u_u are lower and upper bounds on shifts $u[i]$, and r_l and r_u are lower and upper bounds on time strain, the rate at which time shifts change with time sample index i. As noted above, these lower and upper bounds may sometimes be related to bounds on geophysical parameters.

The 2D array $e[i,l]$ contains *alignment errors* computed for all sample indices i and lag indices l. These alignment errors could be computed in alternative ways. For example, we could use an absolute value of differences instead of the square of differences in equation 4; we used the latter to obtain all of the results shown in this paper.

The bounds $[u_l, u_u]$ on shifts $u[i]$ imply that we must compute $e[i,l]$ only for lags l that satisfy $u_l \leq l \leq u_u$. For simplicity in array indexing, we instead use the bounds $0 \leq l \leq n_l - 1$, where $n_l = 1 + u_u - u_l$ is the number of lags l for which we compute $e[i,l]$. In practice, if $u_l \neq 0$, then we simply add u_l to l on the right-hand side of equation 4 as we compute alignment

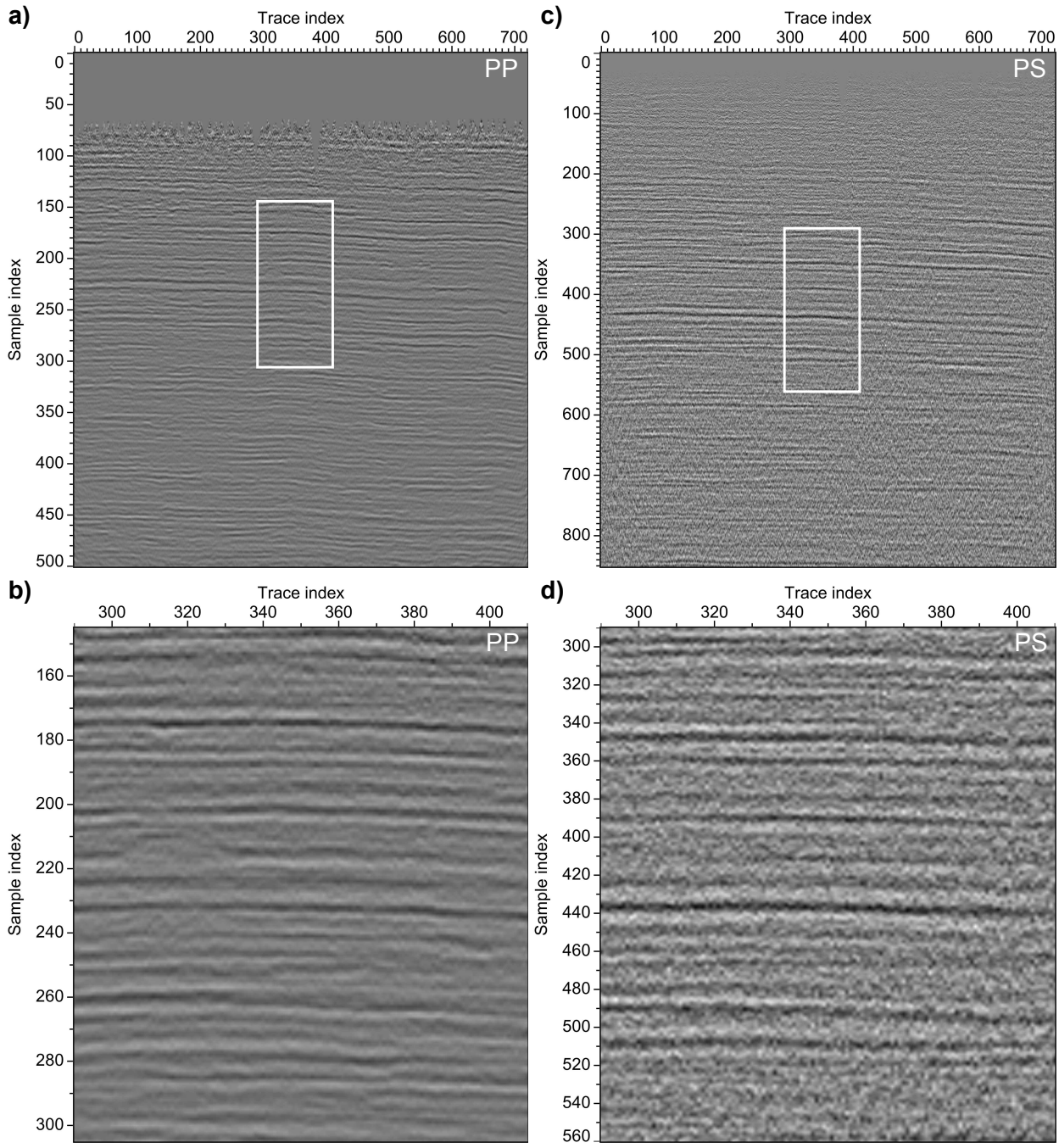

Figure 2. PP (a) and PS (c) images to be aligned, with subsets (b and d) corresponding to the white rectangles. The vertical scales for the PS images are different from those for the corresponding PP images, and were chosen to approximately align reflections in the two images.

errors. If we let f denote one trace from the PP image in Figure 2a and g denote a corresponding trace from the PS image in Figure 2c, then the lower bound on shift is in fact $u_l = 0$ because, as noted above, PS reflections never appear before corresponding PP reflections.

It is important to recognize that the sequence of shifts u defined by equation 2 is the globally optimal solution to a non-linear least-squares problem with linear inequality constraints 3. In practice, the oscillatory nature of seismograms represented by the sequences f and

g causes this minimization problem to have many local minima. As noted above, the ability to find the globally optimal solution to this problem is a key advantage of the dynamic warping method, when compared with alternative methods.

Given a 2D array of alignment errors $e[i,l]$ and bounds on strain r_l and r_u, we compute the optimal shifts $u[i]$ using a dynamic programming algorithm. This algorithm has four phases, which are highlighted in the pseudocode for Algorithm 1.

Algorithm 1 Find shifts $u[i]$

```
 1: procedure FindShifts(r_l, r_u, e, u)
 2:     for l = 0 to n_l − 1                                  ▷ initialize
 3:         d[0, l] = e[0, l]
 4:     for i = 1 to n_i − 1                                  ▷ accumulate
 5:         for l = 0 to n_l − 1
 6:             d_l = ∞
 7:             q_l = max(⌈r_l⌉, l − n_l + 1)
 8:             q_u = min(⌊r_u⌋, l)
 9:             for q = q_l to q_u
10:                 d_q = d[i − 1, l − q] + e[i, l]
11:                 if d_q < d_l
12:                     d_l = d_q
13:                     m_l = q
14:             d[i, l] = d_l
15:             m[i, l] = m_l
16:     i = n_i − 1                                           ▷ minimize d
17:     d_i = ∞
18:     for l = 0 to n_l − 1
19:         if d[i, l] < d_i
20:             d_i = d[i, l]
21:             u[i] = l
22:     while i > 0                                           ▷ backtrack
23:         u[i − 1] = u[i] − m[i, u[i]]
24:         i = i − 1
```

Lines 2–3 initialize the first column, the one for sample index $i = 0$, of a 2D array of *accumulated errors* $d[i,l]$.

Lines 4–15 perform a non-linear accumulation of alignment errors $e[i,l]$, and store the accumulated errors in $d[i,l]$. The accumulation is recursive, in that we compute accumulated errors for sample index i from those for sample index $i-1$. Also, while accumulating, we record in a separate 2D array $m[i,l]$ the lags l for the accumulated errors $d[i-1,l]$ used to compute $d[i,l]$. We think of the array $m[i,l]$ as recording the *error minimizing moves* that are made while accumulating. These moves are constrained by the bounds $[r_l, r_u]$; $\lceil r_l \rceil$ in line 7 denotes the smallest integer not less than r_l, and $\lfloor r_u \rfloor$ in line 8 denotes the largest integer not greater than r_u. Accumulation, the most costly of the four phases, ends with the computation of the last column of accumulated errors $d[n_i - 1, l]$.

Lines 16–21 then search this last column for the lag l that minimizes $d[n_i - 1, l]$. This lag is the optimal shift $u[i]$ for sample index $i = n_i - 1$. The accumulated error $d[n_i - 1, u[n_i - 1]]$ is the minimized sum in equation 2.

Finally, lines 22–24 use the moves recorded in $m[i,l]$ to backtrack, computing the optimal shift $u[n_i - 2]$ from $u[n_i - 1]$, $u[n_i - 3]$ from $u[n_i - 2]$, and so on, until finally computing $u[0]$ from $u[1]$.

Our improved Algorithm 2 is similar, but differs from Algorithm 1 in two ways.

Algorithm 2 Find subsampled shifts $u_i[j] \equiv u[i[j]]$

```
 1: procedure FindShiftsI(r_l, r_u, e, i, u_i)
 2:     for l = 0 to n_l − 1                                  ▷ initialize
 3:         d[0, l] = e[0, l]
 4:     for j = 1 to n_j − 1                                  ▷ accumulate
 5:         h = i[j] − i[j − 1]
 6:         for l = 0 to n_l − 1
 7:             d_l = ∞
 8:             q_l = max (⌈hr_l⌉, l − n_l + 1)
 9:             q_u = min (⌊hr_u⌋, l)
10:             for q = q_l to q_u
11:                 d_q = d[j − 1, l − q]
12:                 for p = 0 to h − 1
13:                     d_q = d_q + e[i[j] − p, l − pq/h]
14:                 if d_q < d_l
15:                     d_l = d_q
16:                     m_l = q
17:             d[j, l] = d_l
18:             m[j, l] = m_l
19:     j = n_j − 1                                           ▷ minimize d
20:     d_j = ∞
21:     for l = 0 to n_l − 1
22:         if d[j, l] < d_j
23:             d_j = d[j, l]
24:             u_i[j] = l
25:     while j > 0                                           ▷ backtrack
26:         u_i[j − 1] = u_i[j] − m[j, u_i[j]]
27:         j = j − 1
```

The first difference is that Algorithm 2 computes shifts $u_i[j] \equiv u[i[j]]$ for only a subset of sample indices $i[0 : n_j - 1] \equiv \{i[0], i[1], \ldots, i[n_j - 1]\}$. Note that an array of indices i in this subset must be specified as input to the procedure in Algorithm 2. To compute shifts $u[i]$ for all sample indices i, we must later interpolate the subsampled shifts $u_i[j] \equiv u[i[j]]$ computed by Algorithm 2.

The second difference lies in the accumulation step, where lines 11–13 in Algorithm 2 have replaced line 10 in Algorithm 1. The additional inner loop **for** $p = 0$ **to** $h-1$ accumulates $h = i[j] - i[j-1]$ (line 5) alignment errors $e[i,l]$. Lines 4–18 compute and store accumulated errors $d[j,l]$ and moves $m[j,l]$ corresponding to the indices $i[j]$ in the subset $i[0 : n_j - 1]$.

Figure 3 illustrates how subsampling in Algorithm 2 can produce smoother sequences of shifts. This figure

shows a closeup view of shifts $u[i]$ computed from $e[i,l]$ obtained by averaging alignment errors for all traces in the PP and PS images shown in Figure 2. For this example, we chose strain limits $r_l = 0$ and $r_u = 2$, so that with $h = 1$ only three values of strain $u[i] - u[i-1]$ are possible: 0, 1, or 2. In other words, as we increase the sample index i, shift $u[i]$ must either remain the same or increase by one or two lags. This restriction explains the rough sequence of shifts for $h = 1$ shown in Figure 3a.

In contrast, for the same strain limits, subsampling with $h = 50$ yields 51 possible values of strain and the smooth sequence of shifts shown in Figure 3a. These shifts were obtained by interpolating subsampled shifts $u_i[j] \equiv u[i[j]]$ computed using Algorithm 2.

Figures 3b and 3c show computational stencils used for $h = 1$ and $h = 5$, again for strain limits $r_l = 0$ and $r_u = 2$. Lines 8 and 9 of Algorithm 2 determine which previously accumulated errors are accessed when computing the accumulated error $d[j,l]$ set in line 17. For $h = 1$ (Figure 3b), Algorithm 2 will access only three previously accumulated errors, $d[j-1,l]$, $d[j-1,l-1]$, and $d[j-1,l-2]$, as it computes $d[j,l]$.

For $h = 5$ (Figure 3c), Algorithm 2 will access 11 previously accumulated errors (corresponding to 11 different lags l), and for each of them it will sum $h = 5$ alignment errors $e[i,l]$, as it determines which change in shift minimizes accumulated error $d[j,l]$.

Note that the expression $l - pq/h$ in line 13 of Algorithm 2 may not have an integer value. For some sample indices i, the lines representing the computational stencil shown in Figure 3c lie between two integer lag indices l. This means that interpolation of alignment errors $e[i,l]$ for non-integer lags l is required in line 13. In practice, we find that linear interpolation or simply choosing the alignment error $e[i,l]$ for the nearest integer lag l is sufficient.

When the spacing between consecutive subsample indices $i[j]$ is $h = 1$ (so that $i[j] = j$ and $n_j = n_i$), then Algorithm 2 is equivalent to Algorithm 1. The ability in Algorithm 2 to sample shifts with larger subsampling intervals enables us to increase accuracy in estimated time strains (and shifts), by sacrificing temporal resolution of those same strains.

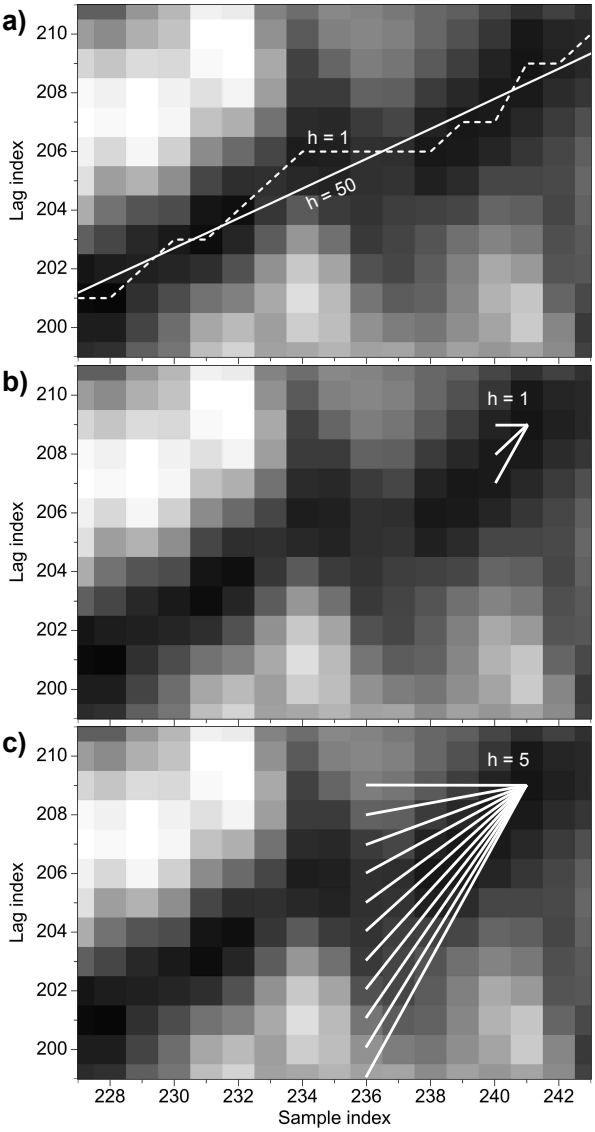

Figure 3. Closeup views of alignment errors $e[i,l]$ and shifts $u[i]$ (a) computed without subsampling ($h = 1$) and with subsampling by a factor of $h = 50$. For strain limits $0 \leq u[i] - u[i-1] \leq 2$, the computational stencil for $h = 1$ (b) can represent only two changes in lag; at each sample, lag must either remain constant or increase by one or two. In contrast, the stencil for $h = 5$ (c) represents 10 changes in lag, and a stencil for $h = 50$ (not shown) represents 100 changes in lag.

3 ACCURACY VERSUS RESOLUTION

Such a trade-off between accuracy and resolution is common in signal processing, and is unavoidable when estimating time shifts from two time series.

Figure 4 illustrates this trade-off for shifts $u[i]$ estimated from the PP and PS images of Figure 2. In Figure 4a, the sequence of shifts $u[i]$ is the globally optimal solution to the optimization problem of equations 2–4. We computed these shifts from alignment errors $e[i,l]$ using Algorithm 2 for $h = 1$.

Figures 4b and 4c show smoother sequences of shifts $u[i]$ computed using the same algorithm for $h = 50$ and $h = 100$, respectively. With Algorithm 2, we first computed subsampled shifts $u[i[j]]$ at the highlighted points, and then interpolated those shifts using piecewise-cubic polynomials to obtain finely sampled shifts $u[i]$. We used polynomials that preserve monotonicity (Fritsch and Carlson, 1980) to ensure that $u[i] - u[i-1] \geq 0$, as required by the strain constraints for PP-PS image registration. A simpler alternative, one more consistent

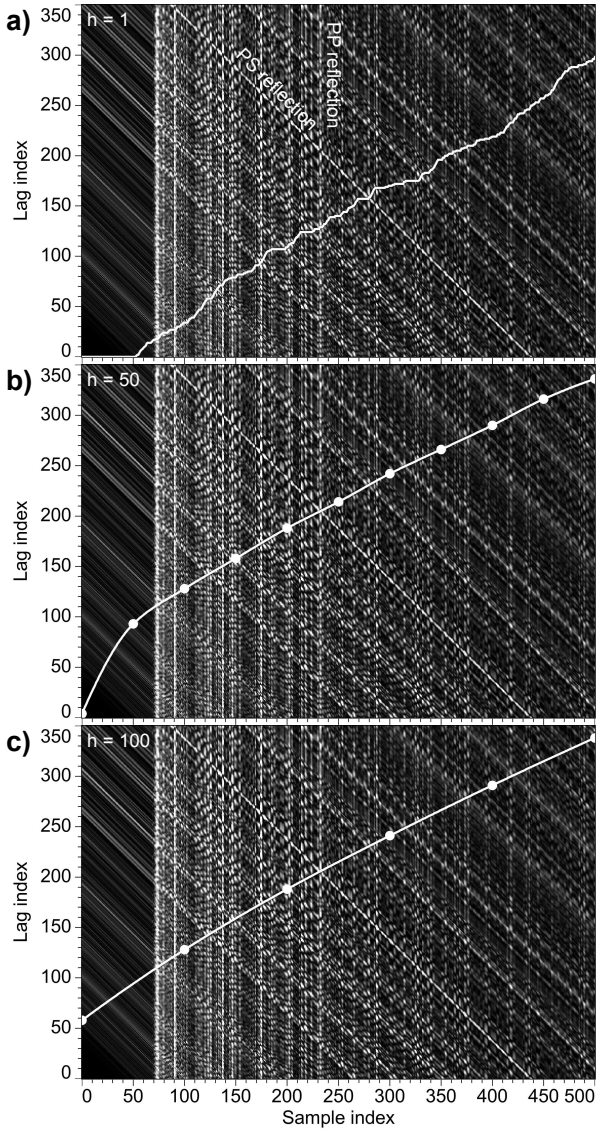

Figure 4. Alignment errors and shifts computed without subsampling (a) and with subsampling by factors of $h = 50$ (b) and $h = 100$ (c). Subsampled shifts are both smoother and more accurate, and cannot be computed by simply smoothing the shifts obtained without subsampling ($h = 1$). In the background images of alignment errors, white vertical features correspond to reflections in the PP image, and diagonal features correspond to (shifted) reflections in the PS image.

with the accumulation of alignment errors along linear paths like those shown in Figure 3c, would be piecewise-linear interpolation.

The sequence of shifts $u[i]$ obtained without subsampling for $h = 1$ exhibit much more detail than those obtained with subsampling for either $h = 50$ or $h = 100$. In other words, for $h = 1$, changes in time shifts with time are well-resolved. However, in this example the shifts obtained for $h = 1$ are useless. We show below that the shifts obtained for $h = 50$ are much more accurate. Moreover, the shifts computed for $h = 50$ cannot be obtained by simply smoothing the shifts computed for $h = 1$.

For $h = 100$ (Figure 4c) Algorithm 2 computes shifts for only six samples. Except near sample index $i = 0$, the interpolated shifts $u[i]$ well approximate those computed for $h = 50$.

To demonstrate that the shifts $u[i]$ computed for $h = 50$ are more accurate than those for $h = 1$, Figure 5 shows closeup views of results obtained for three different arrays of alignment errors $e[i, l]$. Figures 5a–c show averages of 1 (no averaging), 5, and 721 alignment errors computed from pairs of traces in the PP and PS images in Figure 2. Figures 5d–f show the same alignment errors with shifts computed for $h = 1$ and $h = 50$.

Figure 5a shows a closeup view of alignment errors $e[i, l]$ computed from only the middle pair of traces in the PP and PS images. Because differences between traces in the PP and PS images are not limited to time shifts, a unique path of minimum error is not apparent in this 2D array, and the shifts computed for $h = 1$ and $h = 50$ differ significantly, as shown in Figure 5d.

Recall that we can compute a 2D array of alignment errors $e[i, l]$ from each pair of corresponding traces in the PP and PS images. By averaging alignment errors computed for the five pairs of traces nearest the middle of these images, we obtain the results shown in Figures 5b and 5e. Again, the shifts computed for $h = 1$ and $h = 50$ differ significantly, but in this case the shifts for $h = 50$ are more accurate.

We know this because of the results shown in Figures 5c and 5f, which were obtained by averaging alignment errors for all 721 pairs of traces in the PP and PS images. In Figure 5c, the path of minimum error is obvious, and shifts computed using both $h = 1$ and $h = 50$ are well-aligned with this path, which we may assume corresponds to a lateral average of the correct relationship between PP and PS reflection times.

Although the path of minimum error is clearly apparent in Figure 5c, other nearly parallel paths with small error are apparent as well. These other paths are caused by cycle skipping, subtracting one cycle of a reflection waveform in the PS image from a different cycle of the corresponding waveform in the PP image. Recall that one of the advantages of the dynamic warping algorithm is that it will always find the globally optimal and constrained path that minimizes the sum of alignment errors, and will never be trapped in nearby local minima caused by cycle skipping.

While more accurate, shifts $u[i]$ for $h = 1$ shown in Figure 5f are still not as smooth as those for $h = 50$. In fact, the alignment errors and shifts shown in Figure 3a are closeup views of those shown in Figure 5f. Again, the roughness of shifts for $h = 1$ is due to coarse sampling of only three values (0,1 or 2) of time strain

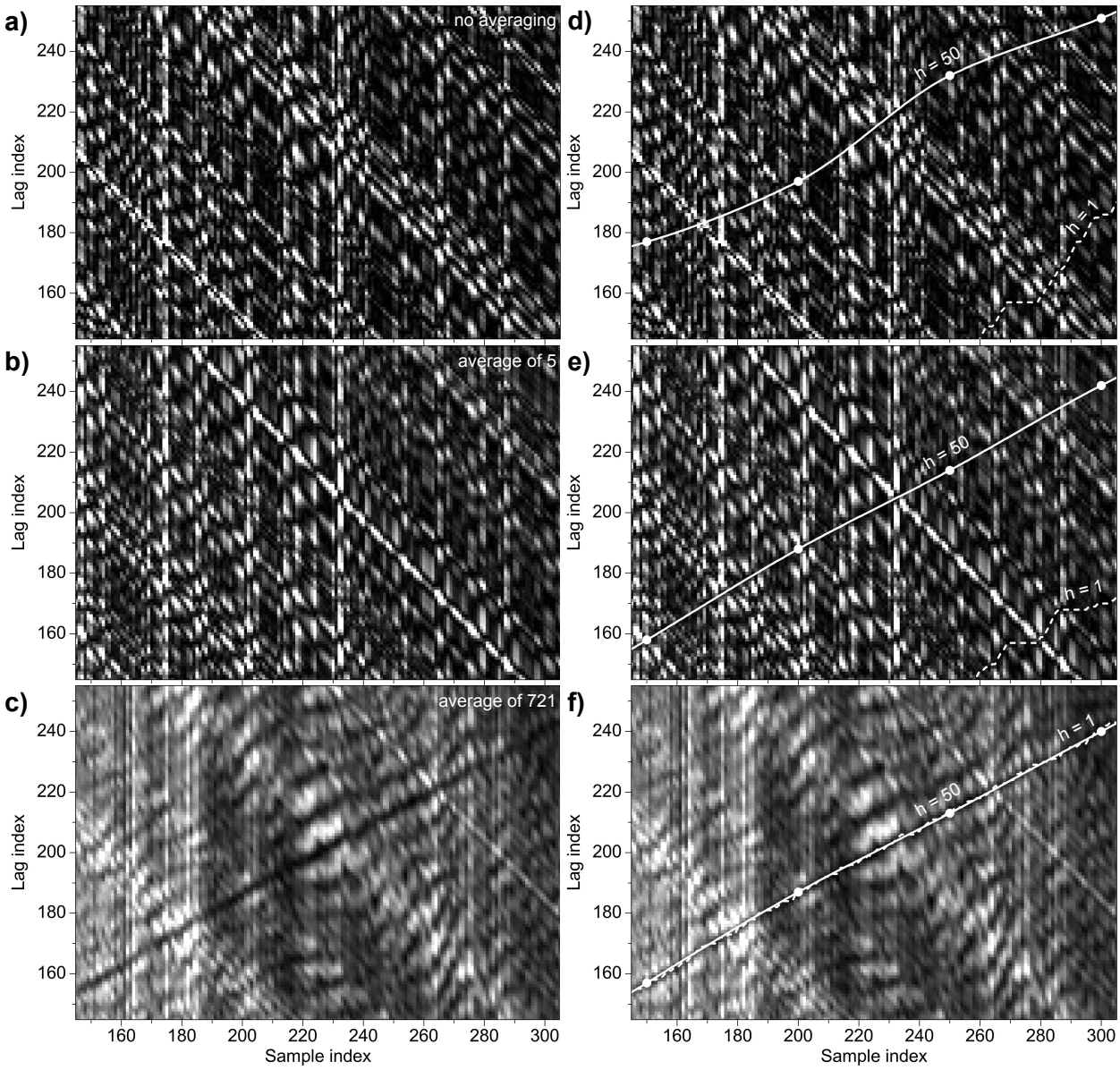

Figure 5. Closeup views of alignment errors $e[i, l]$ obtained from a single pair of PP and PS traces (a) and by averaging alignment errors computed for 5 (b) and all 721 (c) pairs of traces, displayed with (a, b and c) and without (d, e and f) shifts $u[i]$ computed for $h = 1$ and $h = 50$. In this example, the correct shifts (lags) are most apparent in (c), because lateral variation in shifts is small. With subsampling ($h = 50$) the correct shifts can be found by averaging alignment errors computed for only 5 pairs of traces (b and e).

$u[i]-u[i-1]$. The smoothness of shifts for $h = 50$ is due to a much finer sampling of 101 values of time strain. Such fine sampling may be useful in this application to PP-PS image registration, because a value of time strain corresponds directly to a value for

$$\frac{V_P}{V_S} = 1 + 2(u[i] - u[i-1]). \tag{5}$$

For sample indices i in the closeup view shown in Figure 3a, we observe that $V_P/V_S \approx 2$, but this value corresponds to none of the three time strains sampled for $h = 1$.

The results shown in Figure 5 demonstrate an improvement in accuracy that comes from computing subsampled shifts as in Algorithm 2. This improvement is due to the fact that, in this application, the correct shifts are smooth, because they are directly related to the integral of a subsurface property, V_P/V_S. This inherent smoothness implies that shifts $u[i]$ can be subsampled

without aliasing, and then later reconstructed with interpolation.

The results of Figure 5 also demonstrate the difficulty in estimating time shifts from time series that have differences unrelated to time shifts, such as noise and reflection waveform differences. Using just one pair of PP and PS traces (Figures 5a and 5d) it may be impossible to obtain an accurate estimate of the correct shifts $u[i]$. In this example, lateral averaging of alignment errors computed from many pairs of PP and PS traces helps to improve accuracy, but only because lateral variations in time shifts are small. However, while small, lateral variations can be significant, as shown below. By accounting for these variations we can further improve the accuracy of smooth dynamic warping.

4 SMOOTH IMAGE WARPING

We do this using the method for dynamic image warping proposed by Hale (2013). In that method, we first recognize that lines 4–14 of Algorithm 1 perform a recursive non-linear smoothing of alignment errors $e[i,l]$. We then repeatedly apply this smoothing filter — top-to-bottom, bottom-to-top, left-to-right, right-to-left, and so on, for all image dimensions — to alignment errors computed for all pairs of traces in two images. The result is a multi-dimensional array of alignment errors that have been smoothed both vertically and horizontally. Finally, we apply Algorithm 1 once more to the smoothed alignment errors to obtain shifts that vary both vertically and horizontally.

To extend this method to Algorithm 2, we recognize that lines 4–17 of this algorithm perform both a *smoothing and subsampling* of alignment errors. Accumulation of alignment errors e along linear paths like those shown in Figure 3c is equivalent to a smoothing filter, a form of anti-alias filter that enables accumulated alignment errors d to be subsampled. Let $\tilde{e}_f$ denote a 2D array of these forward-accumulated and subsampled errors, the values d stored in line 17 of Algorithm 2. We can apply the same filter in the opposite direction to obtain a different 2D array $\tilde{e}_r$ of reverse-accumulated and subsampled errors. Finally, we can construct symmetrically smoothed and subsampled alignment errors

$$\tilde{e}[j,l] = \tilde{e}_f[j,l] + \tilde{e}_r[j,l] - e[i[j],l]. \tag{6}$$

Each $\tilde{e}[j,l]$ is a sum along piecewise linear paths (Figure 3c) of alignment errors $e[i,l]$. In equation 6, $\tilde{e}_f[j,l]$ is the forward sum for sample indices 0 to $i[j]]$, $\tilde{e}_r[j,l]$ is the reverse sum for sample indices $i[j]$ to $n_i - 1$, and subtraction of $e[i[j],l]$ prevents this value from being included twice in the sum $\tilde{e}[j,l]$.

By repeating this smoothing and subsampling process for each array of alignment errors $e[i,l]$ (i.e., for all pairs of traces in two images), we obtain a collection of arrays of smoothed and subsampled alignment errors $\tilde{e}[j,l]$. In typical applications, the number of samples n_j after smoothing and subsampling will be much smaller (say, by a factor of $h = 50$) than the number of samples n_i input to this process. This means that we will typically be able to keep the entire collection of $\tilde{e}[j,l]$ in fast computer memory, as we apply the same smoothing and subsampling process along other (horizontal) image dimensions.

Finally, as in Hale (2013), we apply a simple dynamic warping (with no further subsampling) to the smoothed and subsampled alignment errors $\tilde{e}$ to obtain shifts u that are subsampled in all image dimensions. We then interpolate these coarsely sampled shifts to obtain smoothly varying shifts for every image sample.

We applied this method (using bilinear interpolation) for smooth dynamic image warping to the PP and PS images shown in Figure 2 to obtain the time shifts shown in Figure 1. We used these time- and laterally-varying time shifts to warp the PS image.

To illustrate the significance of the lateral variation, in Figure 6 we compare PS images after both 1D and 2D warping. We performed 1D warping using time shifts computed from the average of alignment errors of all 721 pairs of traces in the PP and PS images. These shifts were computed for $h = 50$ and are identical to those displayed in Figure 5f.

Figure 6a shows a subset of the PS image after 1D warping, and Figure 6b shows the sample-by-sample differences between this warped PS image and the PP image subset shown in Figure 2b. The images shown in Figures 6c and 6d show comparable results obtained for 2D warping.

While differences between the warped PS images in Figures 6a and 6c may be difficult to see, reflections in the latter appear at slightly earlier times near the middle of the image, consistent with the upward bend in the contours of constant time shift apparent in Figure 1.

This slight lateral variation in time shifts explains the differences apparent in Figures 6b and 6d. After 2D warping, significant differences remain between the warped PS image and the PP image, but the differences are weaker and are less spatially correlated than for 1D warping. These differences suggest that the time shifts computed with smooth dynamic 2D warping are more accurate than those for 1D warping.

5 CONCLUSION

With a simple modification to the dynamic time warping algorithm, we enable a trade-off between accuracy and resolution in estimates of time strains. Our improved dynamic warping algorithm computes smoothly varying shifts by smoothly interpolating shifts that have been coarsely subsampled. While decreasing our ability to resolve rapid variations in time strains, this coarse subsampling increases the accuracy with which smooth time shifts can be estimated.

In registration of PP and PS images (and in other

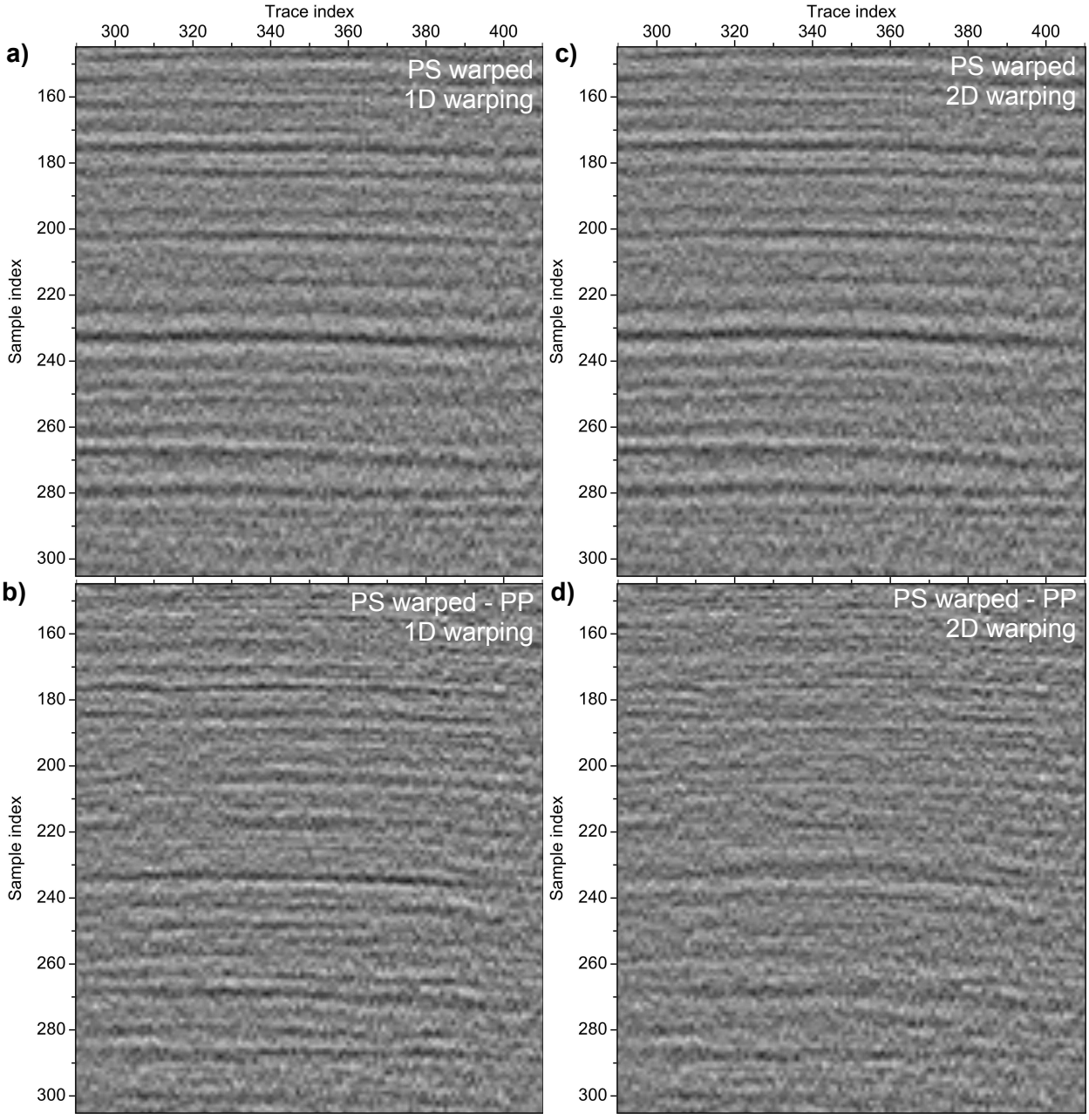

Figure 6. Subsets of the PS image after 1D (a) and 2D (c) warping. For 1D warping (a), shifts were subsampled vertically by a factor of $h = 50$. For 2D warping (c), shifts were subsampled by the same factor both vertically and horizontally. Corresponding sample-by-sample differences (b and d) indicate that 2D warping more accurately aligns the PS image with the PP image.

applications not discussed in this paper), smooth dynamic warping has enabled us to significantly improve the accuracy of time shift estimates, especially where differences between sequences or images to be aligned cannot be attributed entirely to time shifts. In a separate report, we use smooth dynamic warping to estimate V_P/V_S ratios that correspond directly to time strains computed using smooth dynamic warping of 3D images.

The only disadvantage we have found in smooth dynamic warping is an increase in computation cost that is proportional to h, the nominal subsampling interval. The computational complexity of simple dynamic warping via Algorithm 1 is $O(n_i \times n_l)$. The computational complexity of smooth dynamic warping via Algo-

rithm 2 is $O(n_i \times n_l \times h)$. In practice we have found this increase in cost to be worthwhile, especially for multidimensional image warping, in which subsampling of smoothed alignment errors quickly reduces computational cost as smoothing and subsampling are performed for each image dimension.

ACKNOWLEDGMENT

We thank sponsor companies of the Consortium Project on Seismic Inverse Methods for Complex Structures, whose support made this research possible. The seismic PP and PS images shown in Figure 2 were provided courtesy of Sinopec.

REFERENCES

Anderson, K., and J. Gaby, 1983, Dynamic waveform matching: Information Sciences, **31**, 221–242.

Fomel, S., M. Backus, M. DeAngelo, P. Murray, and B. Hardage, 2003, Multicomponent seismic data registration for subsurface characterization in the shallow Gulf of Mexico: Offshore Technology Conference, Expanded Abstracts, 15117.

Fomel, S., M. Backus, K. Fouad, B. Hardage, and G. Winters, 2005, A multistep approach to multicomponent seismic image registration with application to a West Texas carbonate reservoir study: 75th Annual International Meeting, SEG, Expanded Abstracts, 1018–1021.

Fritsch, F., and R. Carlson, 1980, Monotone piecewise cubic interpolation: SIAM Journal on Numerical Analysis, **17**, 238–246.

Gaiser, J., 1996, Multicomponent V_P/V_S correlation analysis: Geophysics, **61**, 1137–1149.

Hale, D., 2013, Dynamic warping of seismic images: Geophysics, **78**, S105–S115.

Liang, L., and D. Hale, 2012, Automatic registration of PP and PS images using dynamic image warping: CWP Report 724.

Nickel, M., and L. Sonneland, 2004, Automated PS to PP event registration and estimation of a high-resolution $V_P - V_S$ ratio volume: 74th Annual International Meeting, SEG, Expanded Abstracts, 869–872.

Sakoe, H., and S. Chiba, 1978, Dynamic programming algorithm optimization for spoken word recognition: IEEE Transactions on Acoustics, Speech, and Signal Processing, **26**, 43–49.