A TECHNIQUE FOR COMPUTING INTERVAL VELOCITIES FROM COMMON MIDPOINT GATHERS

S. M. Deregowski

Introduction

In an article entitled "Snell Waves" (SEP-15), Jon Claerbout showed that if a linear moveout of p sec/m were applied to a common midpoint gather, then any pair of horizontal tangencies would yield an estimate of the interval velocity according to

$$v^2 = \frac{1}{p(m + p)}$$

where m is the slope of the line joining tangent points (Figure 1).

The technique suggested here picks such tangent points and then uses them to construct an interval velocity model by means of a least square fit for the slope m. The tangency picking method has the advantages of being relatively fast and of not requiring that any assumptions be made about an initial velocity model. Unlike the inherently longer and more sophisticated wave equation techniques suggested by Gonzalez (SEP-15), the tangency picking method used here is non-linear. The non-linearity, however, lies in the property that the response is amplitude-independent: so that weak events can score equally with strong ones.

Discussion

The method starts with a common midpoint gather and applies a scan of linear moveouts. For each value of linear moveout p, the gather will typically have the appearance of Figure 2 with horizontal tangencies

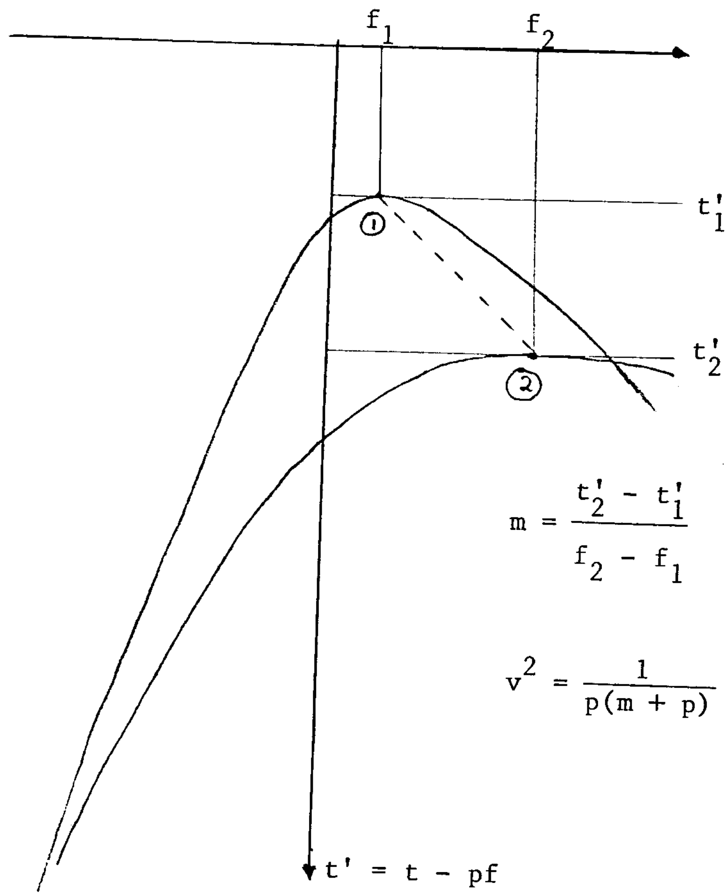


FIGURE 1.--Basic technique for interval velocity estimation from a linearly moved out common midpoint gather.

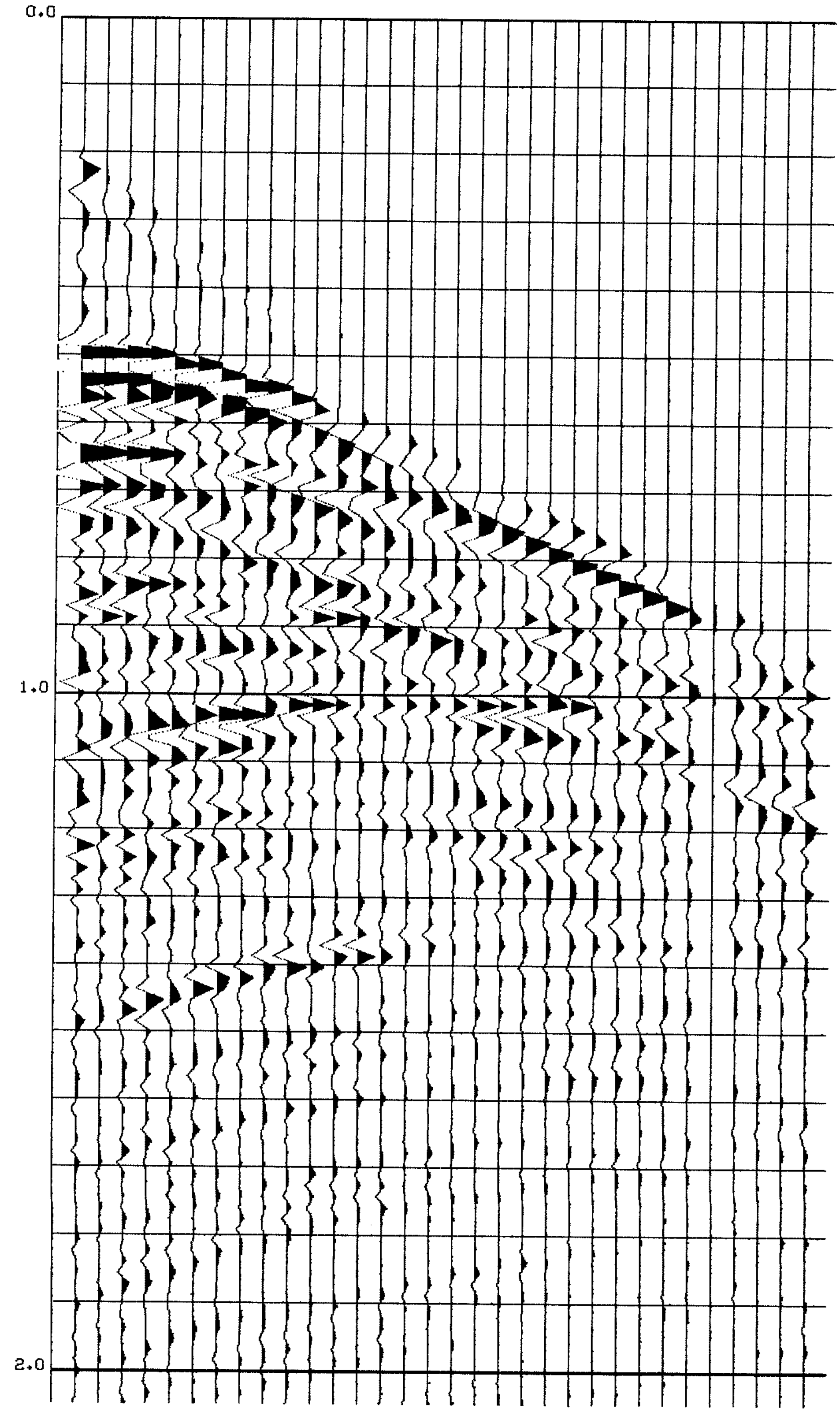


FIGURE 2.--Linearly moved out gather.

confused by the interaction of many events. (Figure 2 is actually a common shot gather, but it will serve for purposes of illustration.) The horizontal tangent points are now picked by means of a simple, velocity- and amplitude-independent three-point operator, which transforms the gather into the form shown in Figure 3. Here each isolated peak corresponds to the position of horizontal tangent, and because of the amplitude-independent non-linearity, tangent points corresponding to "invisible" random noise have also been picked. The next stage is to filter out these noise picks by introducing a weak amplitude dependence and also demanding that any picks fall within a region of coherent energy. After this stage the data takes the appearance of Figure 4, where the noise terms have been filtered out. The picked amplitudes have also been generally readjusted, and they now depend not only on tangency but also upon signal coherency. Within any particular time-gate, the high offset strong amplitude peaks can be associated reasonably safely with primary events (high velocities). The magnitude of each peak lies between zero and unity and itself corresponds to a product of three components, each of which scores between zero and unity. These are the *goodness of tangency* (shown in Figure 3) together with the *data coherency* and *data strength*. The idealized data strength function would take a value of unity everywhere the data was non-zero, and would become zero only when the signal strength fell to zero. In our case, however, the threshold is set to a fixed (small) fraction of some measure of the mean data amplitude. Figure 5 shows the form taken by the product of the data strength/data coherency functions. The "coherency" filtered tangencies shown in Figure 4 were obtained by forming the corresponding product with the raw tangencies (Figure 3).

The interval velocities are now found, in principle, by selecting the primary events in Figure 4 and joining them with straight lines, finding the slopes m, and applying the formula

$$v^2 = \frac{1}{p(m + p)}$$

where p is the linear moveout applied to the original data. The procedure is then repeated for different linear moveouts and the results averaged

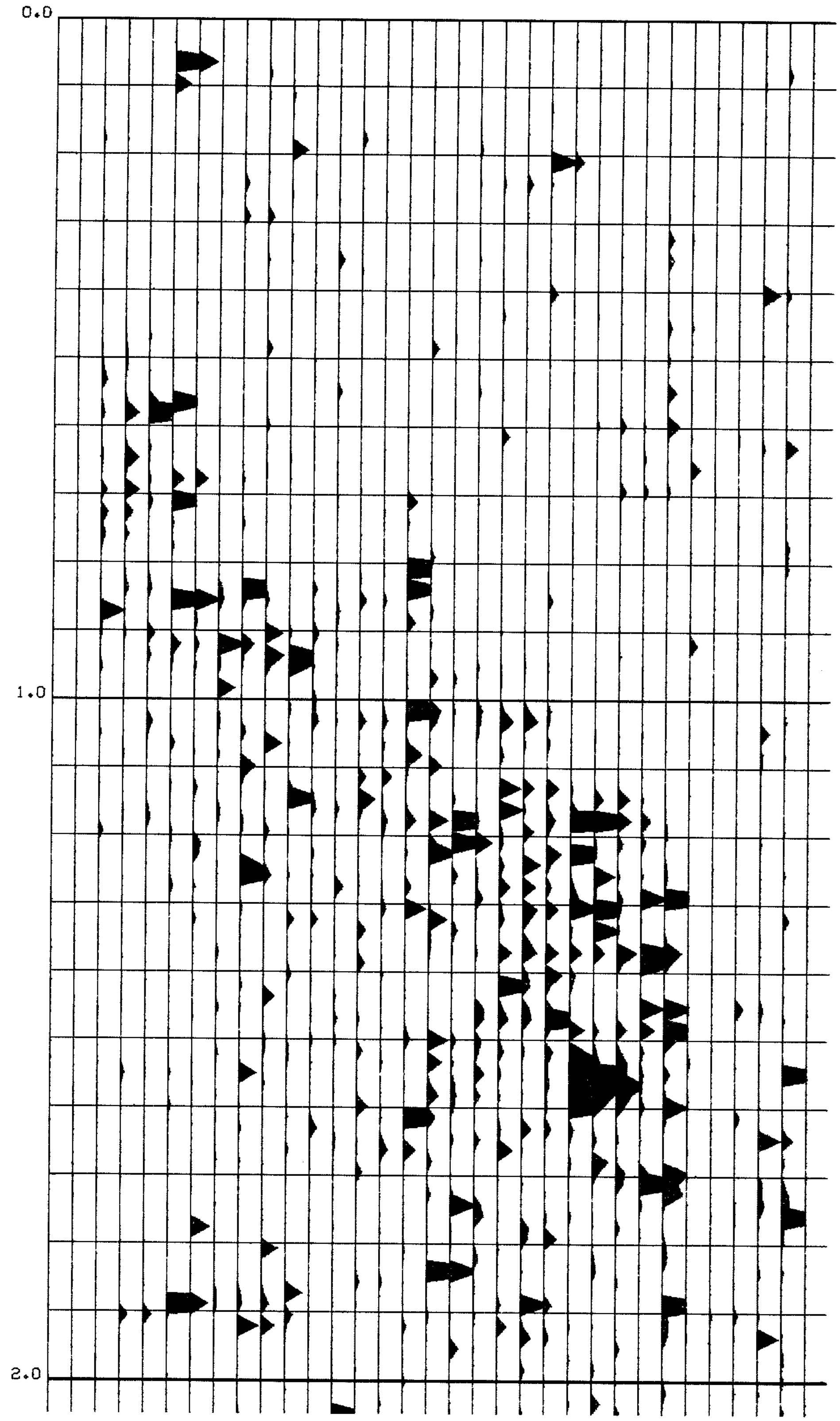


FIGURE 3.--Raw *goodness of tangency.*

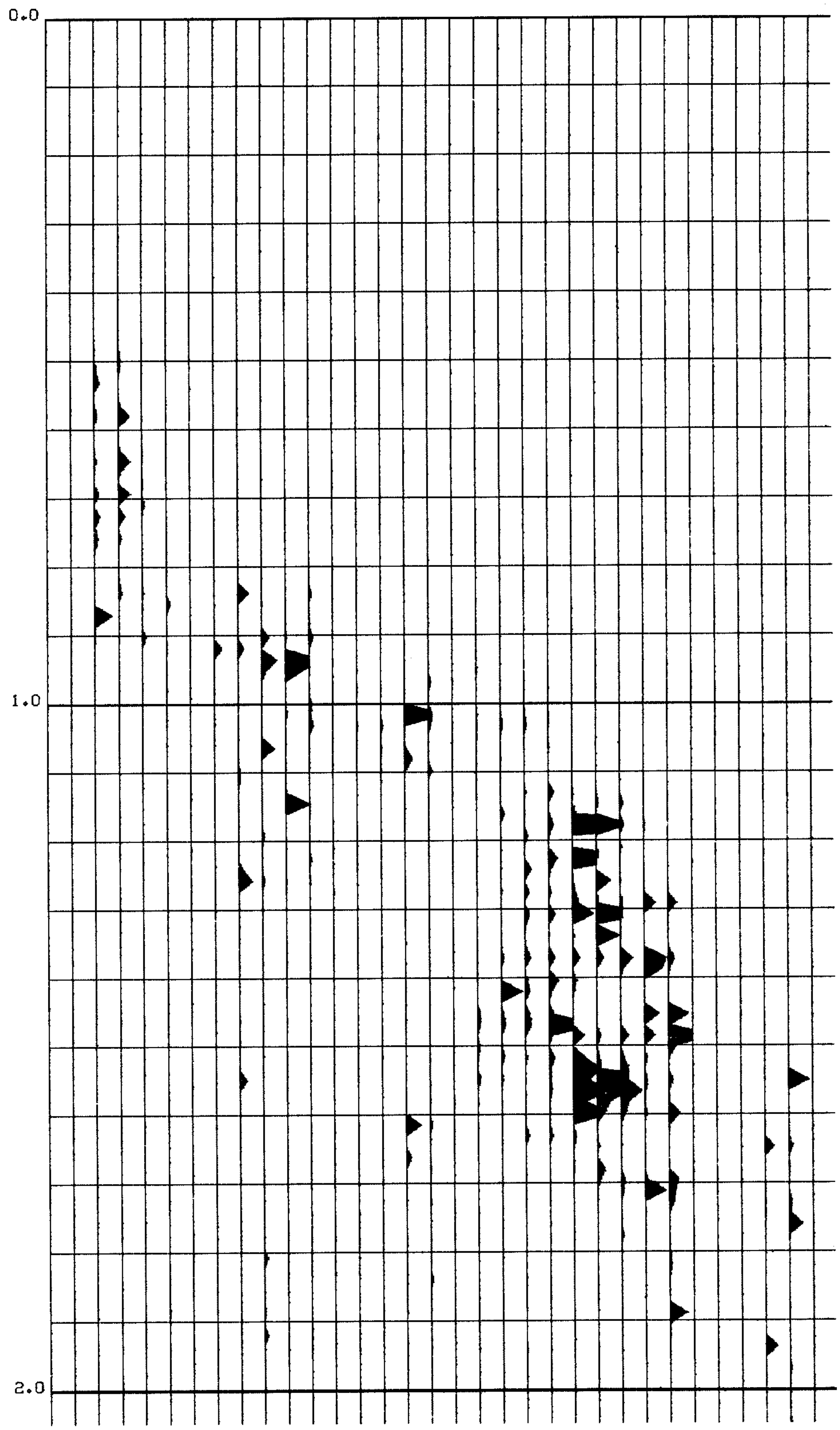


FIGURE 4.--Goodness of tangency after *coherency filtering*.

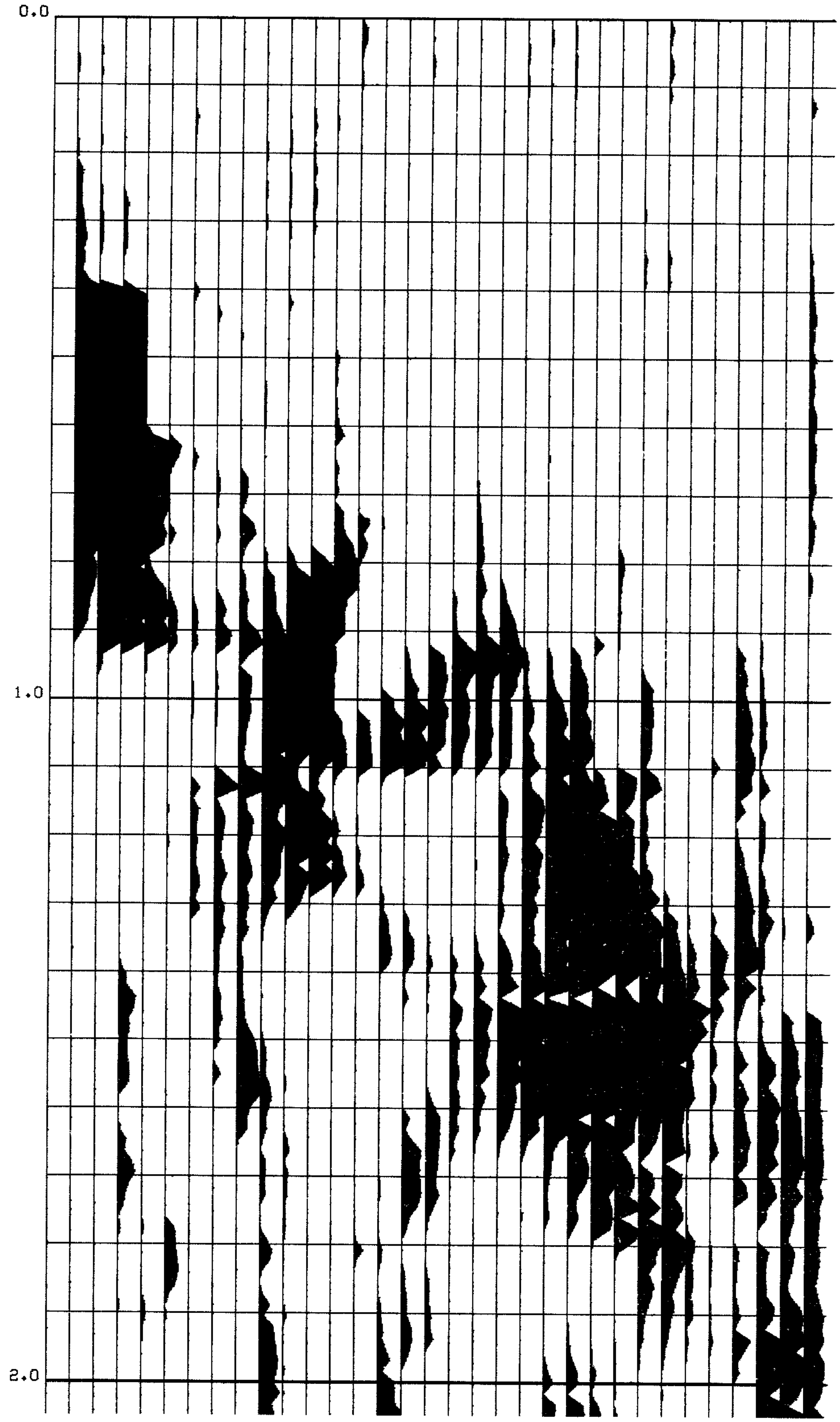


FIGURE 5.--Data coherency (incorporating data strength).

to provide the final set of interval velocities at that particular midpoint. If this method is semi-automated, displays will be required to help identify miss-picks. One possibility is to convert the interval velocity model, defined as a function of midpoint, into a pseudo-seismic section for "eye-balling" and possible editing using $[\gamma_i = (v_{i+1} - v_i)/(v_{i+1} + v_i)]$. This also holds for the velocities obtained as a function of linear moveout.

The actual method for finding interval velocities so far has been the performance of a weighted least squares fit for the slope m over various time gates. In this method, all the energy above a specified threshold falling with a specified time-gate is picked from the "filtered" goodness of tangency (Figure 4) to provide triplets of the form (t'_i, f_i, a_i), where

t'_i = two-way traveltime (with linear moveout)

f_i = offset; a_i = amplitude of pick

A weighted least square fit is now performed with the weight set proportional to the square of the product of amplitude and offset. The square of the offset in used in the definition for the weight in order to shift emphasis to picks associated with primary rather than multiple events. The selection of time-gates for the least square filtering procedure can itself be carried out automatically via the data coherency (Figure 4), where each localized dark patch defines a cluster of tangency points, and the least square procedure simply fits straight lines between such clusters. A more sophisticated method might use full inversion theory applied to a forward model that predicts the positions of primary peaks and multiples from an interval velocity model with a specified number of intervals. (This number of intervals would then be equated with the number of clusters.) But this has not been attempted.

Algorithms and equations

The algorithms employed will now be described in detail and illustrated by means of the synthetic example shown in Figure 6, which corresponds to three diffractor points within a constant velocity medium

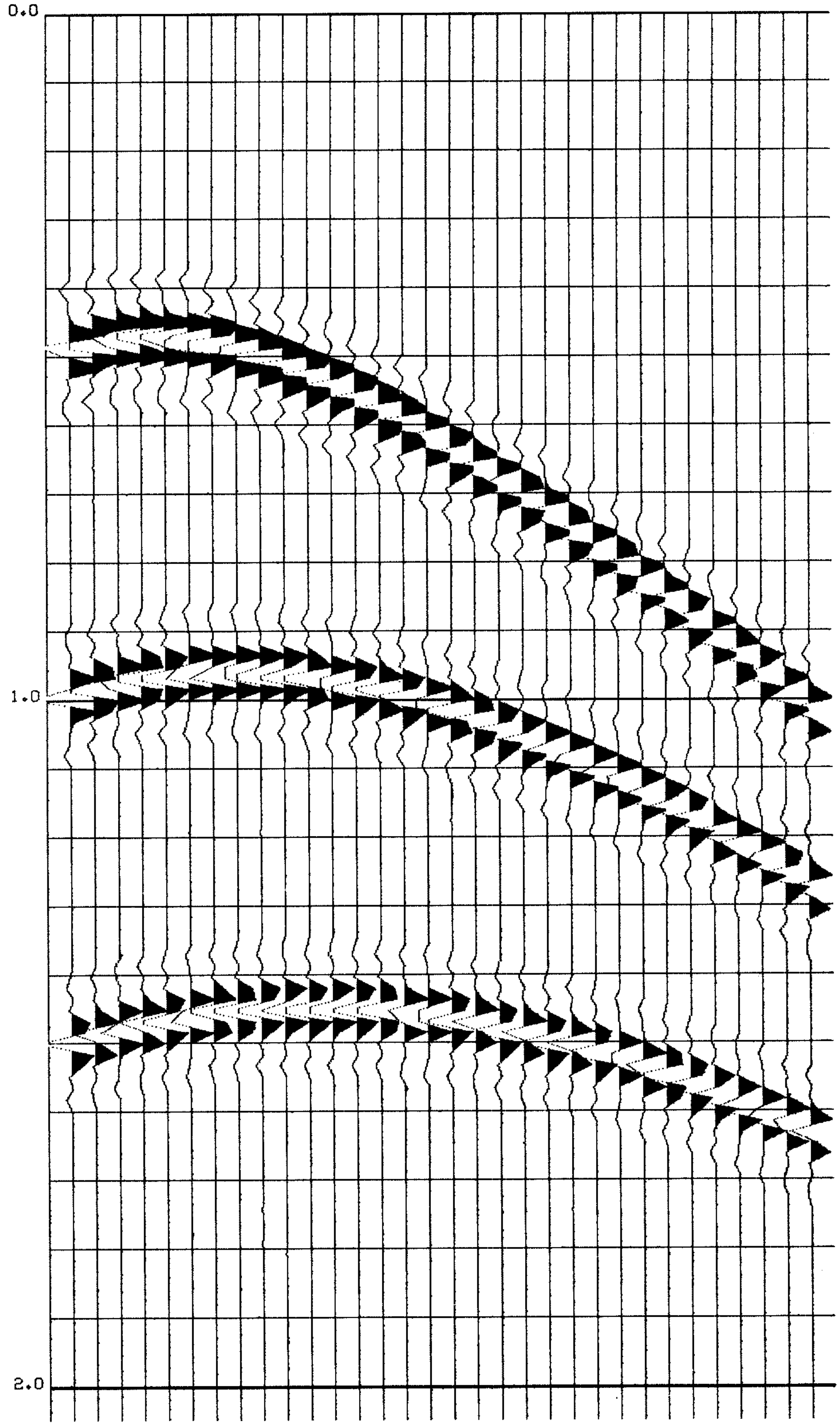


FIGURE 6.--Synthetic example: v = 2500 m/sec; ρ = 0.000125.

(2500 m/sec) and with a linear moveout of 1.25×10^{-4} sec/m. Figure 7 shows the same data but scaled up five hundredfold. This gives a better impression of what is actually seen by the amplitude-independent algorithm. We compute the *goodness of horizontal tangency* according to

$$G(i,j) = \frac{T(i,j)[T(i,j+1) + T(i,j-1)]}{\sqrt{[T^2(i,j) + T^2(i,j+1)][T^2(i,j-1) + T^2(i,j)] + \lambda\Sigma e^2}}$$

where $T(i,j)$ represents the amplitude of an event at sample i of trace j, and

$$\Sigma e^2 = [T(i,j-1) - T(i,j)]^2 + [T(i,j) - T(i,j+1)]^2 + [T(i,j-1) - T(i,j+1)]^2$$

whilst λ is an adjustable scalar parameter (typically, 200-500). The above measure is seen to compare any given point $T(i,j)$ on the i-th horizon with its two neighbors $T(i,j+1)$ and $T(i,j-1)$. As there is no mixing between horizons, we will drop the sample index i. The critical term in the above expression is found in the numerator:

$$T_j(T_{j+1} + T_{j-1})$$

This term can be thought of as "twice the magnitude of a given sample times the average value of its two neighbors," and is the basic measure of tangency used. It can also be interpreted as the dot (scalar) product between the pair of two-dimensional vectors:

$$(T_j, T_{j+1}) \cdot (T_{j-1}, T_j)$$

in which case the square root term in the denominator

$$\sqrt{(T_j^2 + T_{j+1}^2)(T_{j-1}^2 + T_j^2)}$$

is seen to be simply a normalization term designed so that the result

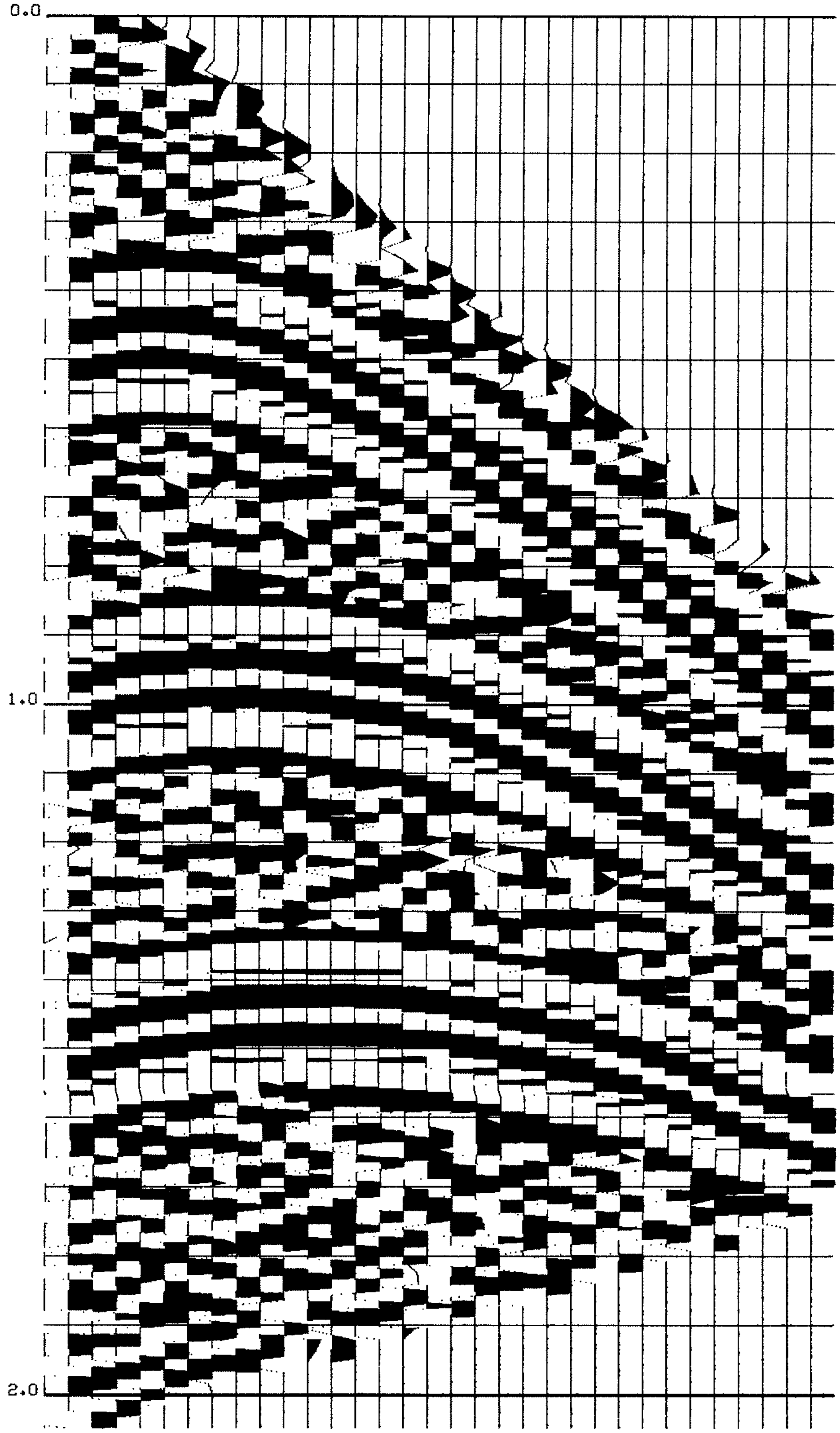


FIGURE 7.--Synthetic example scaled up (× 500).

becomes equivalent to the cosine of the angle between the vectors. Consequently, the maximum score for

$$G_j = \frac{T_j(T_{j+1} + T_{j-1})}{\sqrt{(T_j^2 + T_{j+1}^2)(T_{j-1}^2 + T_j^2)} + \lambda\Sigma e^2}$$

is unity and occurs when $T_{j-1} = T_j = T_{j+1}$ - that is, at a point of horizontal tangency.

The λ-term simply acts to reduce the score when T_{j-1}, T_j, T_{j+1} differ from each other, and naturally, the greater the value of λ the greater the reduction. Figure 8 was computed with $\lambda = 500$ whilst Figure 9 used $\lambda = 50$. The square root term in the definition of G_j might be regarded as a nuisance. If we replace T_j by T_{j-1} in the first bracket and T_j by T_{j+1} in the second, we obtain

$$G_j = \frac{T_j(T_{j+1} + T_{j-1})}{(T_{j+1}^2 + T_{j-1}^2) + \lambda\Sigma e^2}$$

which yields results virtually indistinguishable from those for the previous definition.

We will now move on to the definition of data coherency. The first step is to define a reference trace S by stacking the linearly moved out gather:

$$S(i) = \sum_j T(i,j) \; ; \quad i = \text{sample number}; \; j = \text{trace number}$$

and then computing the correlation factor over short windows of length $2\ell + 1$ samples:

$$\gamma^2(i,j) = \frac{\sum_{k=i-\ell}^{i+\ell} S(k)T_j(k)}{\sqrt{\sum |S(k)|^2 \sum |T_j(k)|^2}}$$

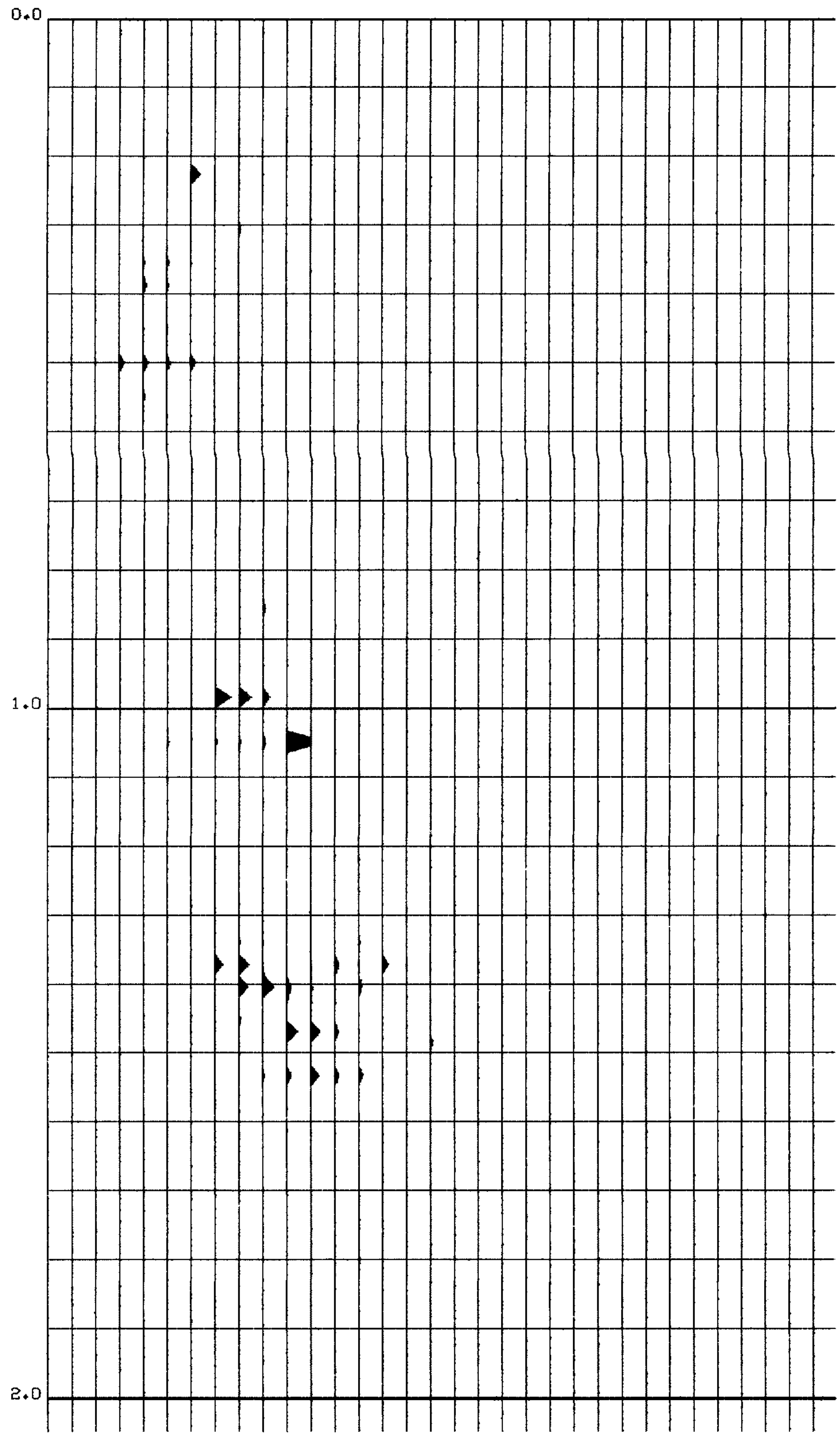


FIGURE 8.--Raw goodness of tangency (λ = 500).

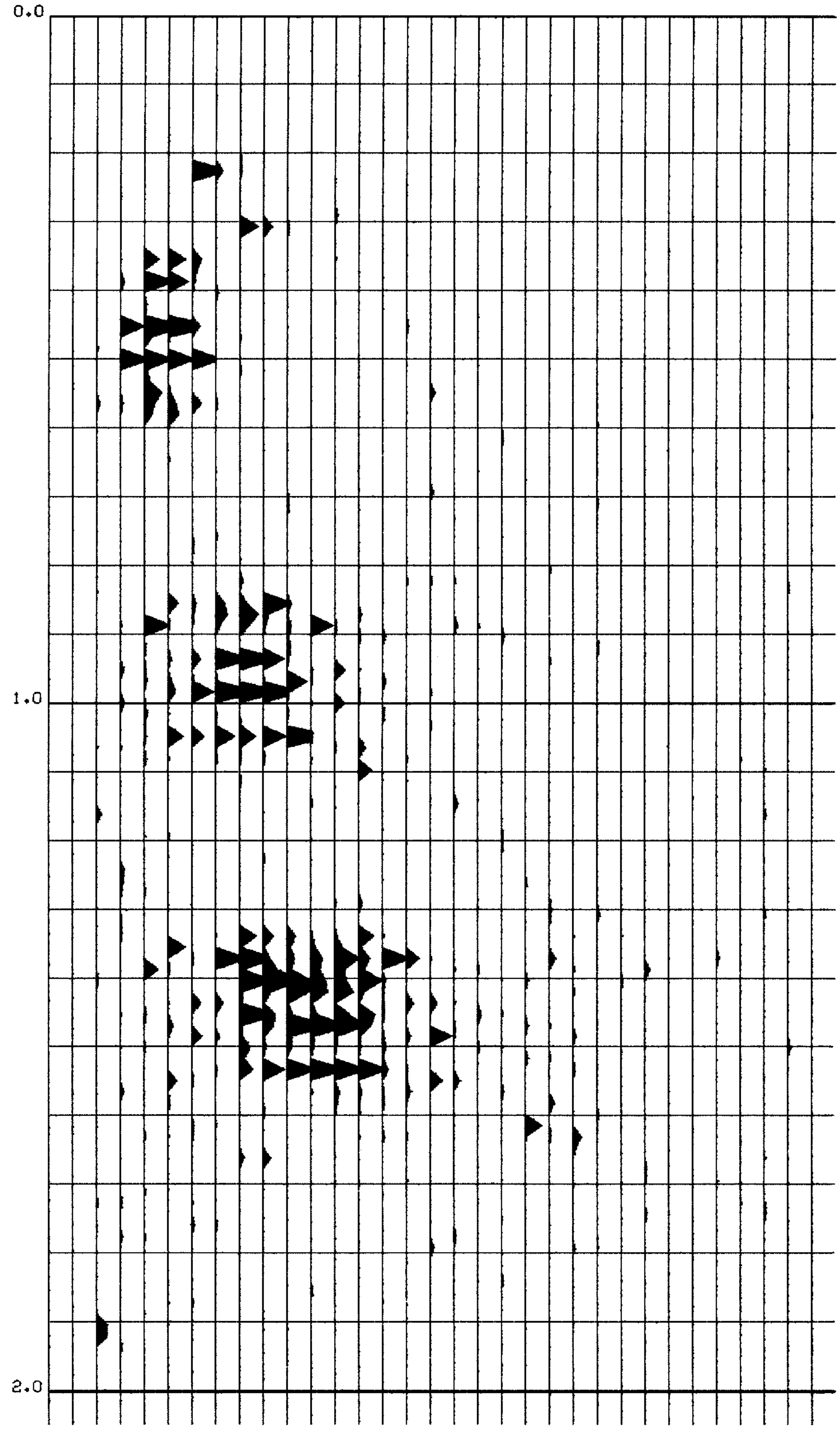


FIGURE 9.--Raw goodness of tangency (λ = 50).

The above is once more seen to be the direct correspondent of the cosine of the angle between two vectors, as defined by the vector dot product. In the case of coherency function, the adjustable parameter is the window length ℓ. Figure 10 shows the coherency obtained using $\ell = 10$, whilst Figure 11 was produced with $\ell = 5$. These plots actually show $\gamma(i,j)$ defined by

$$\gamma(i,j) = \sqrt{\gamma^2(i,j)} \qquad \gamma^2_{ij} \geq 0$$

$$= 0 \qquad \gamma^2_{ij} \leq 0$$

Finally, we introduce a weak amplitude dependence to the coherency function by defining the data strength function

$$k(i,j) = \frac{|T(i,j)|}{|T(i,j)| + \mu . \text{ (maximum data amplitude)}}$$

to define (Figure 12):

$$\overline{\gamma}(i,j) = k(i,j) \ . \ \gamma(i,j)$$

This yields the third adjustable parameter μ. The direct product of the raw picks and coherency function is now formed to provide the final picks for the least squares fitting (Figure 13).

The coherency function $\overline{\gamma}(i,j)$ has another important role to play: the definition of windows for the least squares fit. For this purpose, we first perform a horizontal stack of $\overline{\gamma}(i,j)$ biased to large values. Ideally, the result of such a stack would correspond to a set of box functions with the peak values of the order of unity:

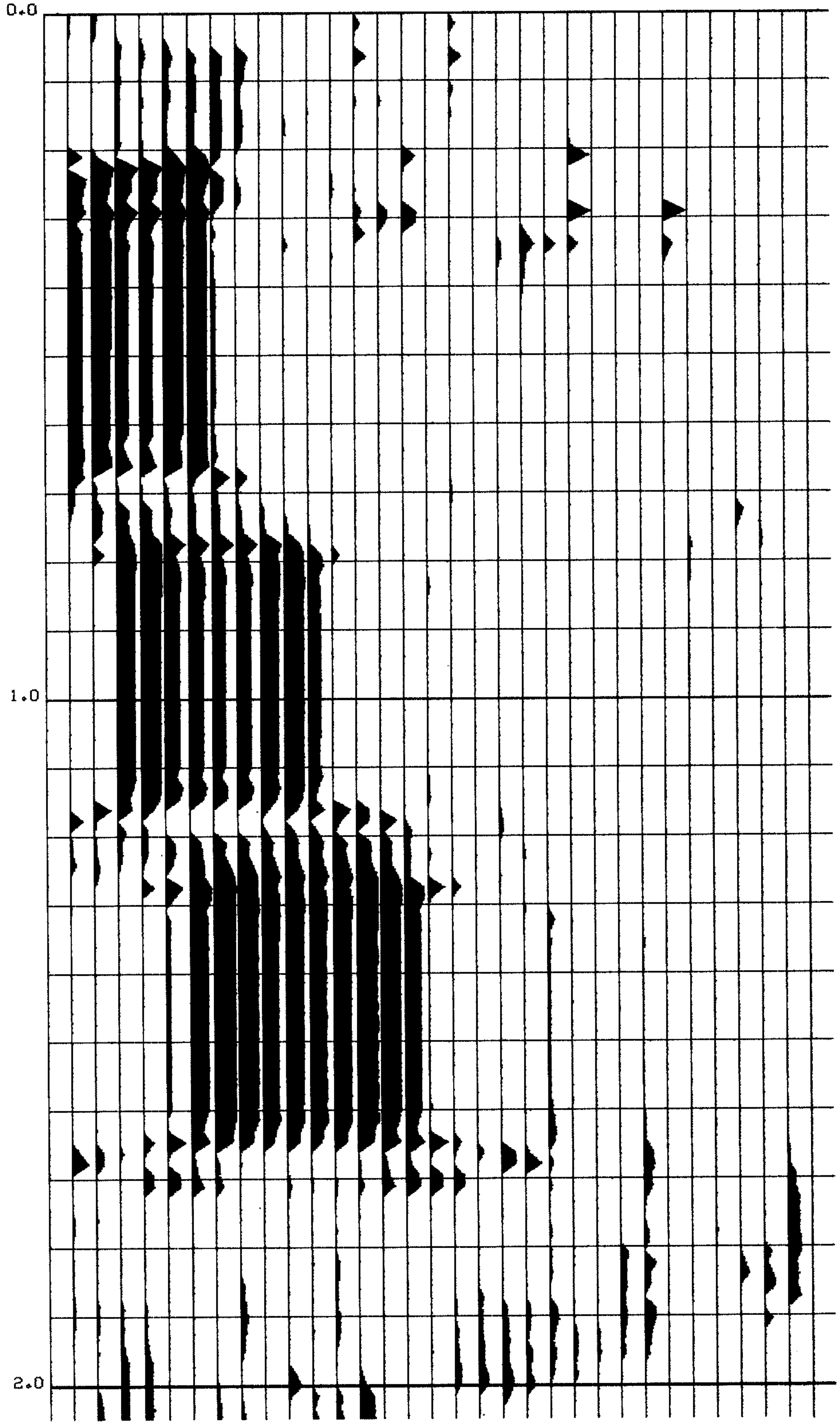


FIGURE 10.--Raw coherency function (ℓ = 10).

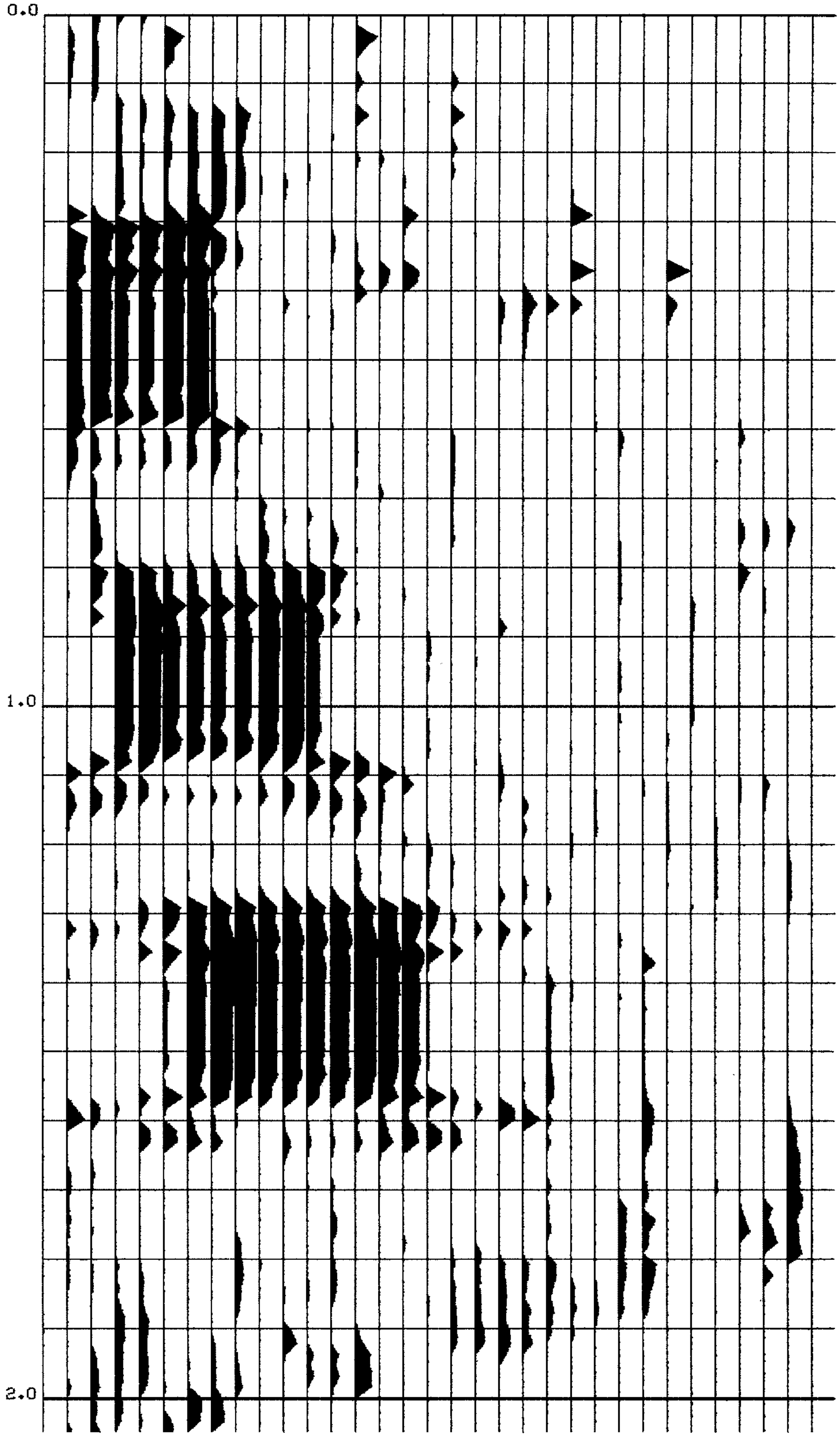


FIGURE 11.--Raw coherency function (ℓ = 5).

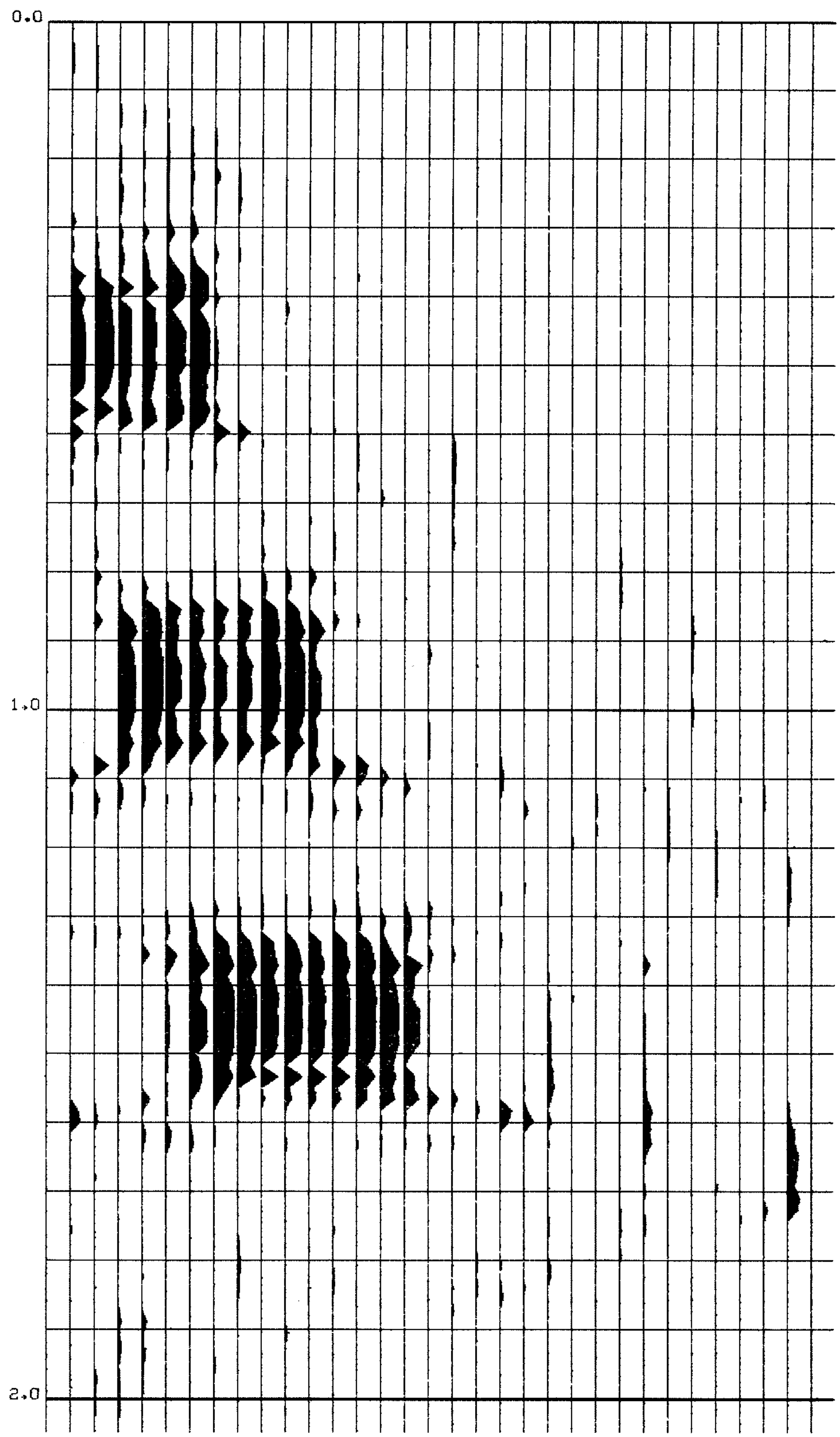


FIGURE 12.--$\overline{\gamma}(i,j)$: coherency function (ℓ = 5) incorporating data strength ($\mu = 10^{-2}$).

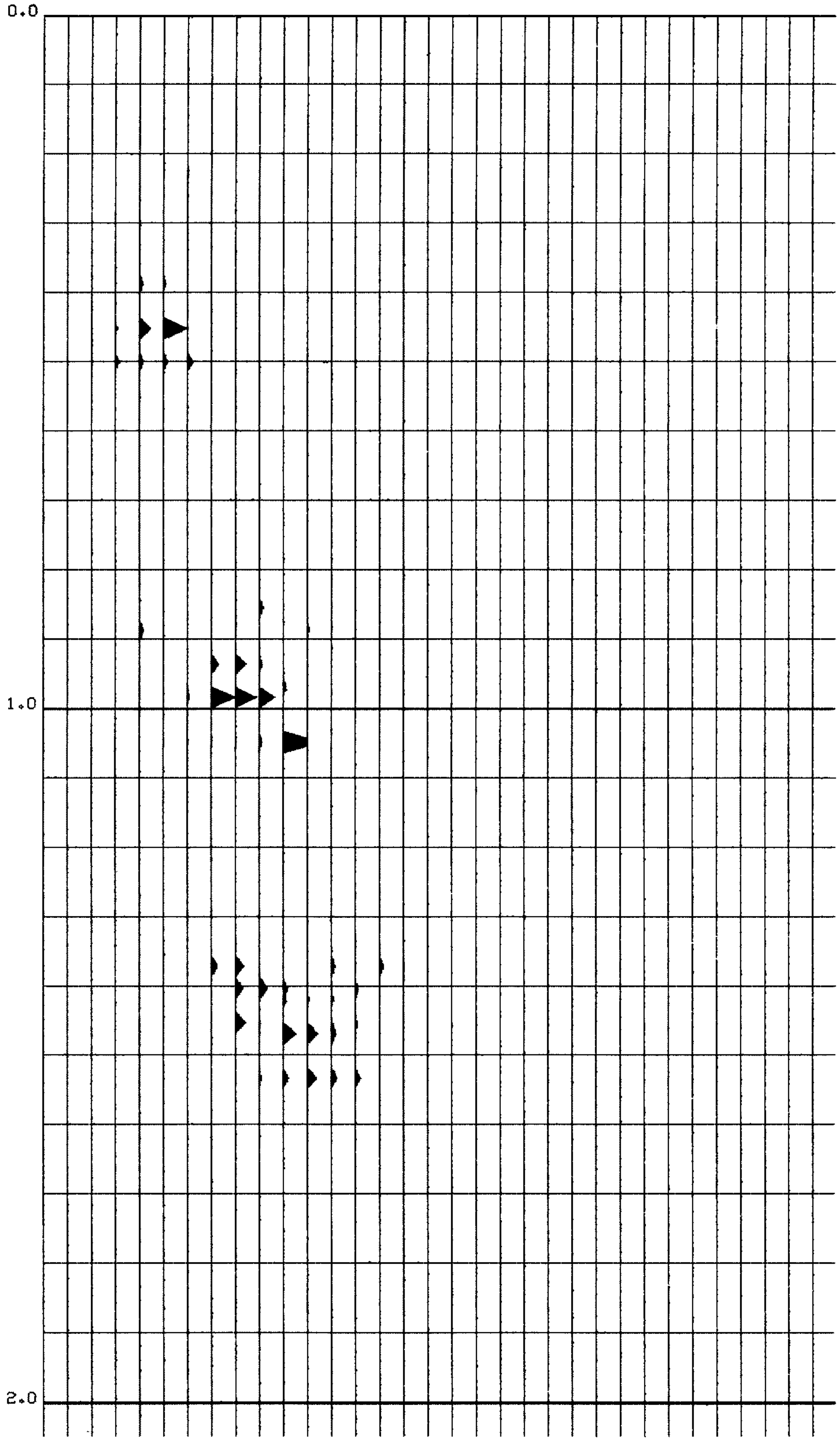
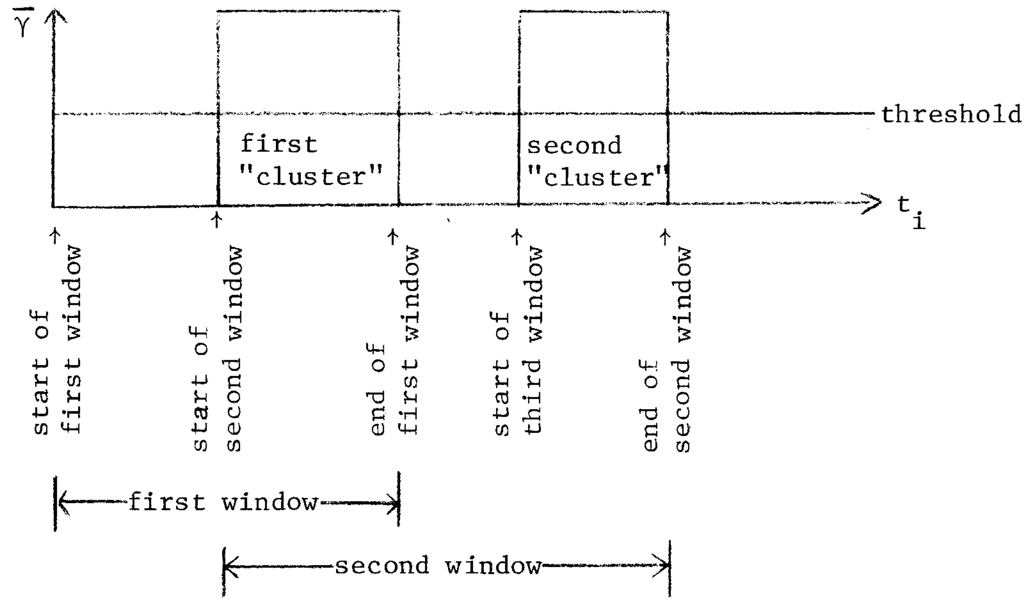


FIGURE 13.-- *Coherency filtered* goodness of tangency, i.e. incorporating *data strength* and *coherency* (ℓ = 5, ρ = 10^{-2}, λ = 200, dx = 100m, ρ = .000125 sec/m).



The onset of each box function defines the start of a new window, and its termination the end of a previous one. The picked values falling within each box function are said to define a cluster. A confidence weight on the velocity defined by fitting between clusters could now be set according to the maximum weights within the two clusters - that is, by the minimum of the two maxima. But prior to the least square fit, the maximum weight within each cluster is reset to be unity in order to ensure a meaningful slope.

The "windowing trace" could be defined by simply setting

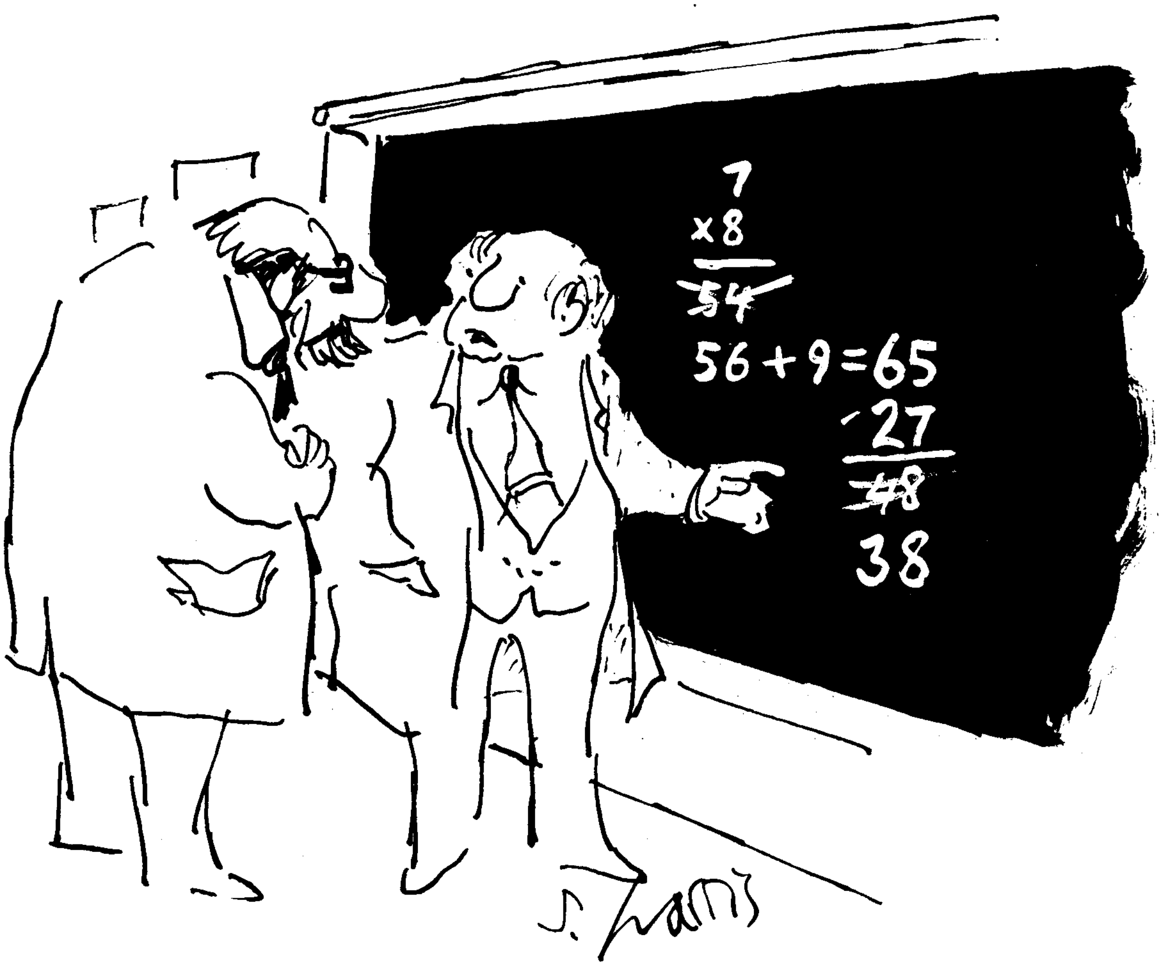
$$\bar{\gamma}_1 = \max_j(\bar{\gamma}_{ij})$$

but a more robust definition is

$$\bar{\gamma}_i = \frac{\sum_j (\bar{\gamma}^3_{ij})}{\sum_j (\gamma^2_{ij})}$$

In the above, we are simply performing a weighted average with the weight defined by the square of the sample amplitude.

In conclusion, the above technique describes a possible method for obtaining interval velocities. It is a method, however, that has yet to be proven (or otherwise!) by application to real datasets.



"Keep in mind that, like everyone else, I only use ten percent of my brain."