# Constrained automatic moveout picking from semblances

William S. Harlan

September 2001

## Introduction

I first implemented this method of automatic picking in 1987-88 while working at the Osservatorio Geofisico Sperimentale in Trieste Italy and at Stanford University as a visiting scholar. The implementation was an obvious application and simplification of the work of John Toldi [1] [2] which I had closely followed for several years. That implementation was in a computer language I now try to forget.

Over the years, I applied that algorithm to many varieties of data, including noisy data with weak reflections, with large lateral velocity anomalies, and with residual moveouts after prestack time or depth imaging. I've rarely resorted to hand-picking tools except as a Q.C. tool for this algorithm.

Recently I reimplemented this method to take advantage of a newer C++ Gauss-Newton optimization algorithm, with fewer restrictions on the number of spatial dimensions. But the numerical properties are exactly the same.

## The objective function

First one must perform a stacking velocity analysis to construct a hypercube of stacking semblances $S(m, \mathbf{x})$ as a function of a moveout parameter $m$ and dimensions $\mathbf{x}$ including zero-offset time and a spatial X and Y.

The moveout parameter should be chosen so that resolution is approximately constant for large or small moveouts. I prefer to use the squared reciprocal of a conventional stacking velocity. That moveout parameter also conveniently includes flat or negative moveouts.

The goal of this algorithm is to find a smooth surface $m(\mathbf{x})$ that gives a single-valued moveout as a function of all coordinates. The surface has a limited number of degrees of freedom to make the surface stiff. The best surface maximizes the integral of the semblance over all coordinates.

$$\max_{m(\mathbf{x})} \int S[m(\mathbf{x}), \mathbf{x}] d\mathbf{x} \tag{1}$$

This is essentially the same objective function as used by John Toldi, but without his tomographic constraints on moveout. I assume that stiffness and hard bounds alone will be sufficient to keep the moveouts reasonable. Zhang Lin [3] similarly dropped Toldi's physical constraints, but made more extensive changes to the objective function and optimization. I make no significant changes to Toldi's objective function. I use a very conventional Gauss-Newton optimization, with Toldi's iterative smoothing and relaxation to ensure convergence.

## Motivation

Stiffness prevents the moveout surface from jumping easily from primary reflections to a multiple and back again. Hard bounds may be necessary in areas where multiples and other coherent noise are more common than primaries. A hard constraint is implemented by simply muting that part of the semblance that falls outside an allowed range. Ranges of moveouts can be specified relative to reference moveouts, or independently.

The moveout surface is stiff over all spatial dimensions, including time. This allows more redundancy over more dimensions than available to a human interpreter. A surface integral includes the contribution of many weak reflections rather than just the few strong reflections visible to the eye on a contour plot.

Weak lateral velocity anomalies can cause stacking "velocities" to swing wildly by 50% or more as inner and outer offsets are delayed relative to each other. A human interpreter will reject such swings as unphysical, although the characteristic signature of such anomalies can be seen clearly when neighboring ensembles are included. An automatic picker sees more data and finds the swings are essential for consistency.

Some automatic pickers attempt first to identify peaks and fit them second. Unfortunately all such peaks influence the final result, even when they are inconsistent with neighbors. An integral, on the other hand, does not increase the penalty of moving farther from a multiple once the peak has been found inconsistent.

## Optimization

A Gauss-Newton algorithm finds model parameters that best fit some data after a non-linear transform. This particular objective function does not look like an inversion at first glance: we have an objective function and a model, but no data. Let our data be a single value that represents the maximum normalized integral of semblances.

A Gauss Newton algorithm requires three transforms.

First we need a full non-linear calculation of the data from the model. Forward integrate semblance $S(m, \mathbf{x})$ over moveouts $m$ and positions $\mathbf{x}$:

$$d = \int S[m(\mathbf{x}), \mathbf{x}] d\mathbf{x}. \tag{2}$$

Next, we need a linearized perturbation of the data $\Delta d$ for a perturbation of the model $\Delta m$ – i.e. a gradient:

$$\Delta d = \int \tfrac{\partial}{\partial m} S[m(\mathbf{x}), \mathbf{x}]\, \Delta m(\mathbf{x}) d\mathbf{x}. \tag{3}$$

Finally, we need the adjoint of the linearized forward transform:

$$\Delta m(\mathbf{x}) = \tfrac{\partial}{\partial m} S[m(\mathbf{x}), \mathbf{x}]\, \Delta d. \tag{4}$$

Set data $d$ to the maximum possible normalized sum and plug into your favorite Gauss-Newton solver.

Actually, the conventional solver requires one modification. In early iterations, the moveout surface is far from the optimum peaks. The surface needs to be close enough to a peak for the gradient to push the solution in the correct direction. Fortunately, John Toldi had a clever solution. In early iterations the surface should be extremely stiff with perhaps only a few basis functions over the range of each dimension. The semblance cube should be heavily smoothed in the direction of the moveout parameter — up to half the allowed range of values in the first iteration. The locations of the peaks are less accurate but the surface always finds itself on a broadened flank with the gradient pointing in the correct direction. After this small number of degrees of freedom has been allowed to converge, then later iterations can reduce the smoothing and increase the number of basis functions. In the final iteration, the semblance volume is not smoothed at all, and the surface is as flexible as the user allows.

## References

[1] J. L. Toldi. *Velocity analysis without picking.* PhD thesis, Stanford University, 1985.

[2] J. L. Toldi. Velocity analysis without picking. *Geophysics*, 54(2):191–199, 1989.

[3] L. Zhang. Automatic picking and its applications. *Stanford Exploration Project Report, http://sepwww.stanford.edu/research/reports/*, 70:275–292, 1991.