Stanford Exploration Project, Report 100, April 20, 1999, pages 293–314

Robust and stable velocity analysis using the Huber function

Antoine Guitton and William W. Symes[1]

keywords: *inversion, velocity, noise, nonlinear*

ABSTRACT

The *Huber function* is one of several robust error measures which interpolates between smooth (l^2) treatment of small residuals and robust (l^1) treatment of large residuals. Since the Huber function is differentiable, it may be minimized reliably with a standard gradient-based optimizer. Tests with a linear inverse problem for velocity analysis, using both synthetic and field data, suggest that (1) the Huber function gives far more robust model estimates than does least squares, (2) its minimization using a standard quasi-Newton method is comparable in computational cost to least squares estimation using conjugate gradient iteration, and (3) the result of Huber data fitting is stable over a wide range of choices for $l^2 \rightarrow l^1$ threshold and total number of quasi-Newton steps.

INTRODUCTION

Robust error measures such as the l^1 norm have found a number of uses in geophysics. As measures of data misfit, they show considerably less sensitivity to large measurement errors than does the mean square (l^2) measure. Since geophysical inverse problems are generally ill-posed, relatively noise insensitive misfit measures can yield far more stable estimates of Earth parameters than does the mean square measure (Claerbout and Muir, 1973; Taylor et al., 1979; Chapman and Barrodale, 1983; Scales and Gersztenkorn, 1988; Scales et al., 1988). This insensitivity to large noise has a statistical interpretation: robust measures are related to long-tailed density functions in the same way that the mean square is related to the (short-tailed) Gaussian (Tarantola, 1987).

A simple choice of robust measure is the l^1 norm: denoting the residual (misfit) components by r_i, $i = 1, ..., N$, l^1 norm of the residual vector is $\sum_{i=1}^{N} |r_i|$. This function is not smooth: it is singular where any residual component vanishes. As a result, numerical minimization is difficult. Various approaches based for example on a linear programming viewpoint (Barrodale and Roberts, 1980) or iterative smoothing (Scales et al., 1988), have been used with success but require considerable tuning. Moreover, the singularity implies

[1]**email:** antoine@sep.stanford.edu, symes@caam.rice.edu

that small residuals are "taken as seriously" as large residuals, which may not be appropriate in all circumstances.

These drawbacks of the l^1 norm have led to various proposals which combine robust treatment of large residuals with Gaussian treatment of small residuals. In the work reported here, we use a hybrid l^1-l^2 error measure proposed by Huber (Huber, 1973):

$$M_\epsilon(r) = \begin{cases} \frac{r^2}{2\epsilon}, & 0 \le |r| \le \epsilon \\ |r| - \frac{\epsilon}{2}, & \epsilon < |r| \end{cases}$$

We will call $\sum_{i=1}^{N} M_\epsilon(r_i)$ the *Huber misfit function*, or Huber function for short (Figure 1). Note that the Huber function is smooth near zero residual, and weights small residuals by mean square. It is reasonable to suppose that the Huber function is easier to minimize than l^1 while still robust against large residuals.

This paper describes the application of the Huber misfit function to velocity analysis. Estimation of RMS velocity (or slowness) can be posed as a linear inverse problem through the *velocity transform* described in the next section. Definition of the misfit *via* the Huber function (or any other robust error measure) results in a nonlinear optimization problem for the velocity model. This nonlinearity would seem to compare unfavorably with the least squares (l^2) treatment of the same problem, which leads to a linear system (the normal equation) and so can be solved by efficient iterative methods such as conjugate gradient. We show that use of an appropriate nonlinear optimization method gives a Huber-based solution with comparable efficiency to that of conjugate gradient least squares solution. Thus the noise rejection properties of the Huber misfit function come at no appreciable premium in computational effort. In the work reported here we have used a version of the Limited

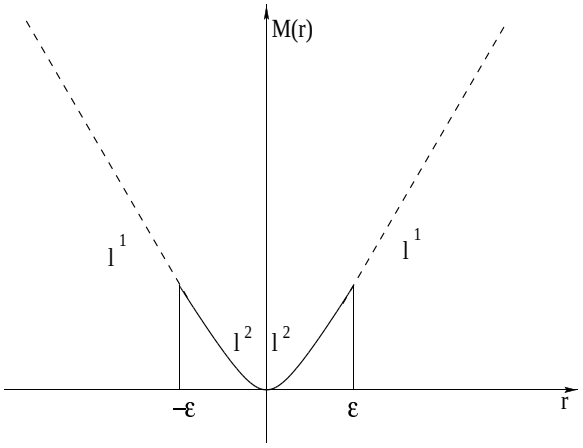

Figure 1: Error measure proposed by Huber (Huber, 1973). The upper part above ϵ is the l^1 norm while the lower part is the l^2 norm. antoine1-huber [NR]

Memory BFGS algorithm (Nocedal, 1980) as implemented in the Hilbert Class Library (Gockenbach et al., 1999). Other nonlinear iterative optimizers could be used; we have solved the same examples with nonlinear conjugate gradient methods (Fletcher, 1980) and obtained comparable results. We note that specially adapted Huber minimizers have been suggested (Ekblom and Madsen, 1989). One of our questions in beginning this work was whether a standard quasi-Newton method, as opposed to a special solver, would perform satisfactorily in Huber estimation.

The second section of the paper explains the velocity transform and formulates a linear inverse problem for velocity analysis. The third and fourth sections present synthetic and field data examples.

APPLICATION TO VELOCITY ESTIMATION

The *velocity domain* representation of seismic data is an alternative to the standard CMP presentation. Transformation of CMP data into the velocity domain (producing a velocity *model* or *panel* of the data) exhibits clearly the moveout inherent in the data and therefore, forms a convenient basis for velocity analysis as a linear inverse problem. The velocity transform $\mathbf{A}$ from the model space (velocity domain) into the data space (CMP gathers) stretches the velocities back in the offset plane (superposition of hyperbolas) whereas the adjoint operation ($\mathbf{A}'$) squeezes the data (summation over hyperbolas):

$$\mathbf{A} = \mathbf{HS},$$

with

$$\mathbf{S}m(t,x) = \sum_s \frac{t}{\tau} w(s,x,\tau) m(\tau,s) \Big|_{\tau=\sqrt{t^2-s^2x^2}}$$

$$\mathbf{A}' = \mathbf{S}'\mathbf{H}',$$

with

$$\mathbf{S}'d(\tau,s) = \sum_x w(s,x,\tau) d(t,x) \Big|_{t=\sqrt{\tau^2+s^2x^2}}$$

where $w(s,x,\tau)$ is a weighting function, $\mathbf{H}$ is a filter that we define later. $\mathbf{A}$ is related to the velocity stack as defined by Taner and Koehler (1969).

The problem is: given a CMP gather can we find a velocity panel which synthesizes it *via* $\mathbf{A}$? In equations, given data $\mathbf{d}$, we want to solve for model $\mathbf{m}$:

$$\mathbf{Am} \approx \mathbf{d}.$$

A simple way to solve this problem is to find a model $\mathbf{m}$ that minimizes the *mean square* misfit

$$(\mathbf{Am}-\mathbf{d})'(\mathbf{Am}-\mathbf{d}).$$

This optimization problem is equivalent to the linear system ("normal equations")

$$\mathbf{A}'\mathbf{Am} = \mathbf{A}'\mathbf{d}.$$

This system is easy to solve if $\mathbf{A}'\mathbf{A} \approx \mathbf{I}$, i.e if $\mathbf{A}$ is close to unitary: then $\mathbf{m} = \mathbf{A}'\mathbf{d}$. In general, $\mathbf{A}$ is far from an unitary operator for many reasons. However, the choice of a weighting function compensates to some extent for geometrical spreading and other effects (Claerbout and Black, 1997):

$$w(s,x,\tau) = \frac{1}{\sqrt{(\tau^2+s^2x^2)^{1/2}}} \frac{\tau}{\sqrt{\tau^2+s^2x^2}} \sqrt{xs}.$$

The summation in the velocity space boosts low frequencies. Claerbout and Black (1997) suggest that a good choice of filter **H** is a half derivative operator ($\sqrt{i\omega}$). These choices for **H** and $w(s, x, \tau)$ bring **A** closer to being an unitary operator.

Since the data is noisy, the modeling operator is not unitary and the numbers of equations and unknowns may be large, an iterative data-fitting approach seems reasonable:

$$\min_{\mathbf{m}}(E(\mathbf{Am} - \mathbf{d}))$$

where **m** is the model, **d**, the data we want to fit, **A** the modeling operator, and E a misfit measurement function we have to choose. We have already presented one possibility, namely that E is the l^2 norm (least squares inversion). A convenient iterative method in this case is to solve the normal equation using conjugate gradient iteration. We refer to this approach as "CG" or "l^2". An alternative approach is to take for E the Huber function introduced in the first section. With this Huber misfit measure, the velocity transform inverse problem is no longer equivalent to a linear system. We choose to solve it using a general-purpose nonlinear optimizer, as mentioned above, rather than one of several special-purpose methods invented for this type of problem. We refer to this approach as"Huber" or the "Huber solver".

The next two parts of this paper compare the performance of the CG algorithm to Huber in the velocity analysis application.

SYNTHETIC DATA TESTS

To compare the Huber function with the least squares measure, we generate a synthetic CMP gather (Figure 2) that we perturb by introducing: (1) missing traces, (2) a low velocity aliased plane wave, and (3) some sparsely distributed spiky noisy events. These data sets constitute the input for the iterative schemes ($\epsilon = 0.01$ for each result). The panels display the model space on the left (after 20 iterations), and the data space on the right. The bottom right panels show the modeling of the last velocity result. All these results (Figures 4, 6, 8) prove the following: outcome of the Huber solver is insensitive to spiky events, like a pure l^1 norm misfit function. The outcome of the missing data problem was probably less predictable, but again, Huber copes more easily with the inconsistency introduced in the data.

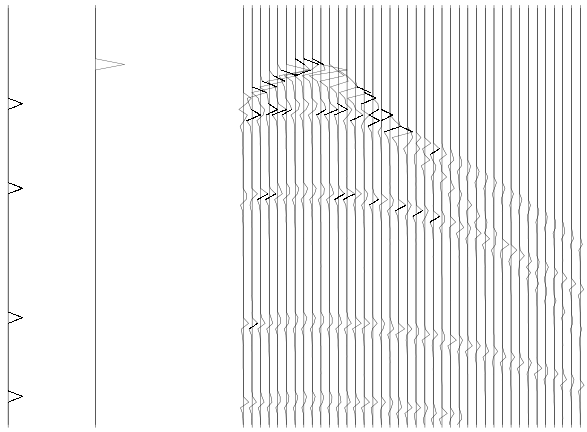

Figure 2: Left: ideal velocity panel. Right: data model. antoine1-datamodel [CR]

FIELD DATA EXAMPLES

In this section we compare the CG to the Huber function for two CMP gathers. We divide this section in three: first, we show the inverted velocity panels and modeled data obtained for both the CG and Huber solver. The second and third section address the stability problems and computational efficiency.

Velocity analysis on field data

Figure 9 displays different interesting features: the first CMP (*A*) shows low velocity events (ground roll or guided waves) and time shifts near offset 2km while the second example (*B*) shows bad traces with high amplitudes. The same clip has been applied to each figure. Note that the offsets are probably wrong since we have reflected waves traveling at about 10 km/s for Input *A*! The velocities are computed after 5 iterations only, and $\epsilon = 0.001$ for both gathers. Figure10 shows the velocity analysis using Huber and CG on a first data set. It appears that the CG result has artifacts at times above 1.5 sec. We see some horizontal stripes that make reliable interpretation difficult. In contrast, the Huber result displays a focused velocity corridor. Some low frequency events do appear in the upper right part of the panel but they do not interfere with the main fairway. It is interesting to notice that Huber separates the low-frequency low-velocity noise from the signal whereas the l^2 measure spreads it along the velocity axis. If we now model those results back into the data-space, we obtain Figure11. We notice that CG does not do a good job in estimating the upper part. Furthermore, some high frequency noise appears. The Huber result however looks close to the original data. In particular, the upper part is well estimated and no harmful artifacts are visible. The anomalous high amplitude first trace does not affect the final results either.

Let's now look at the results for Input *B*. Figure12 show the strength of the Huber solver compared to CG. The CG velocity panel shows horizontal stripes in the velocity scan making any reliable picking quite impossible. The Huber velocity panel displays a focused bended corridor with low noise amplitude. If we now model those results back into the data-space, we get Figure13. Those sections support the same observations as previous, and

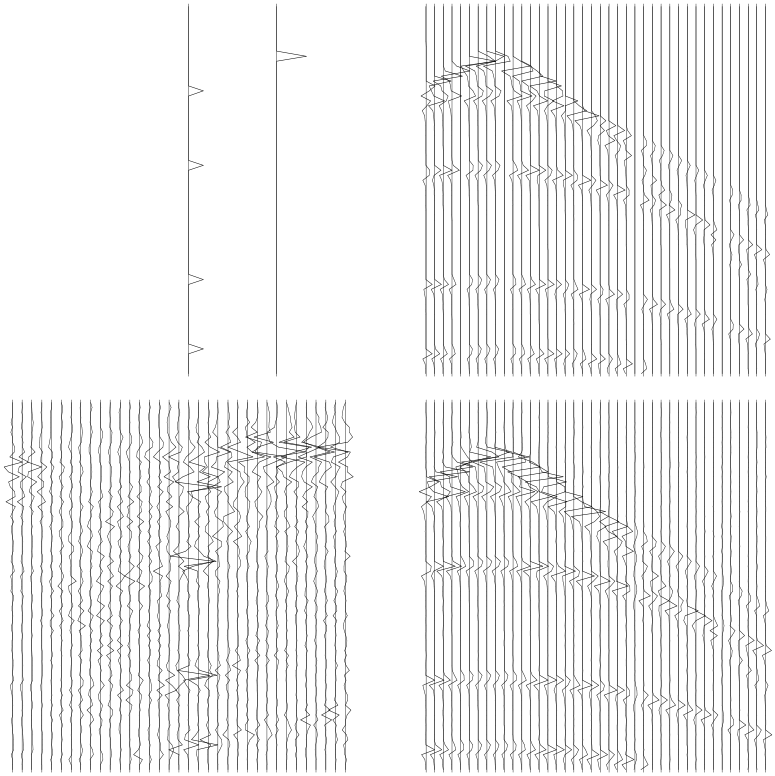

Figure 3: CG result with missing data. antoine1-vel-miss2g [CR]

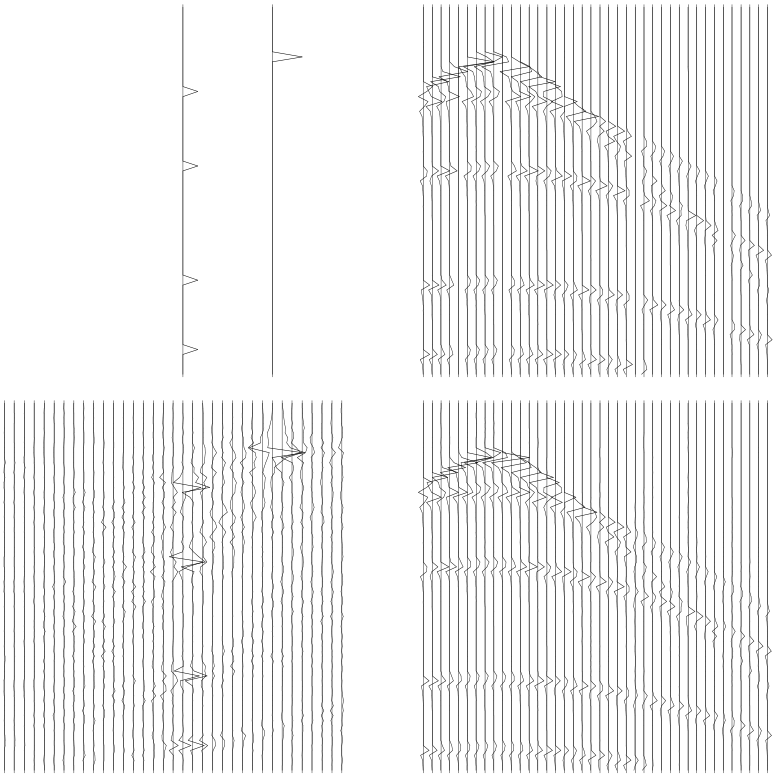

Figure 4: Huber result with missing data. antoine1-vel-miss3h [CR]

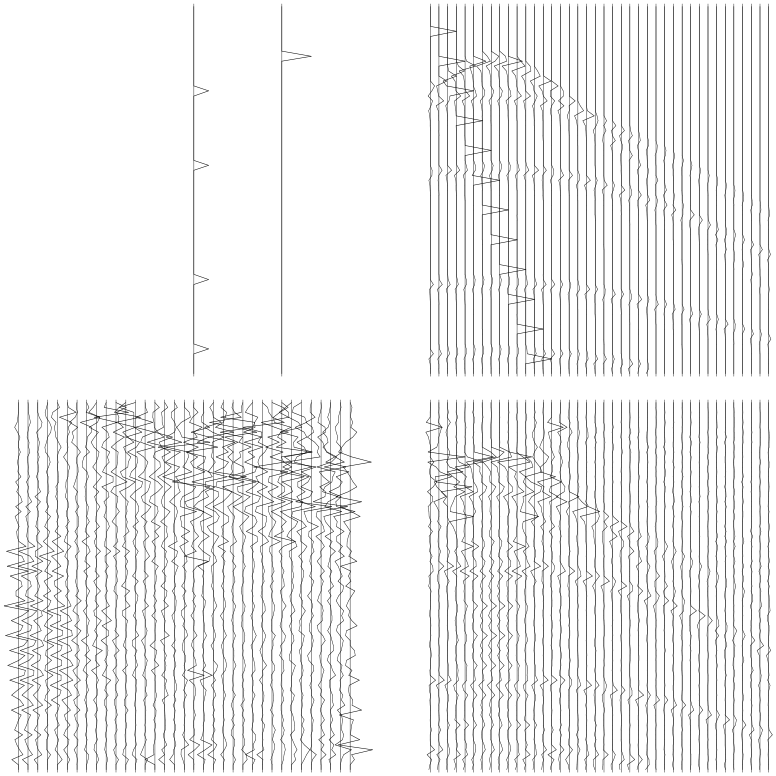

Figure 5: CG result with a slow plane wave. antoine1-vel-surf1g [CR]

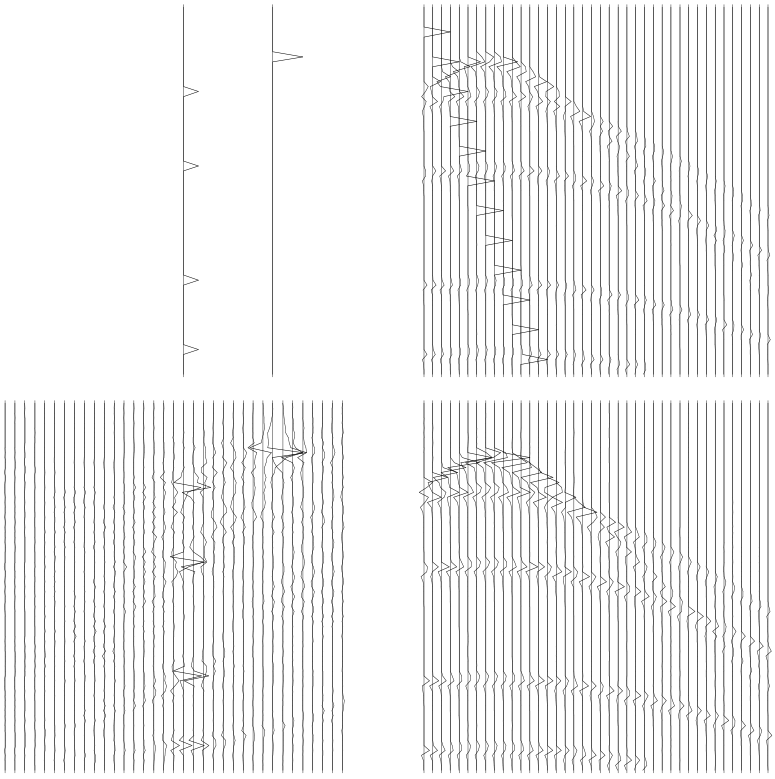

Figure 6: Huber result with a slow plane wave. antoine1-vel-surf2h [CR]

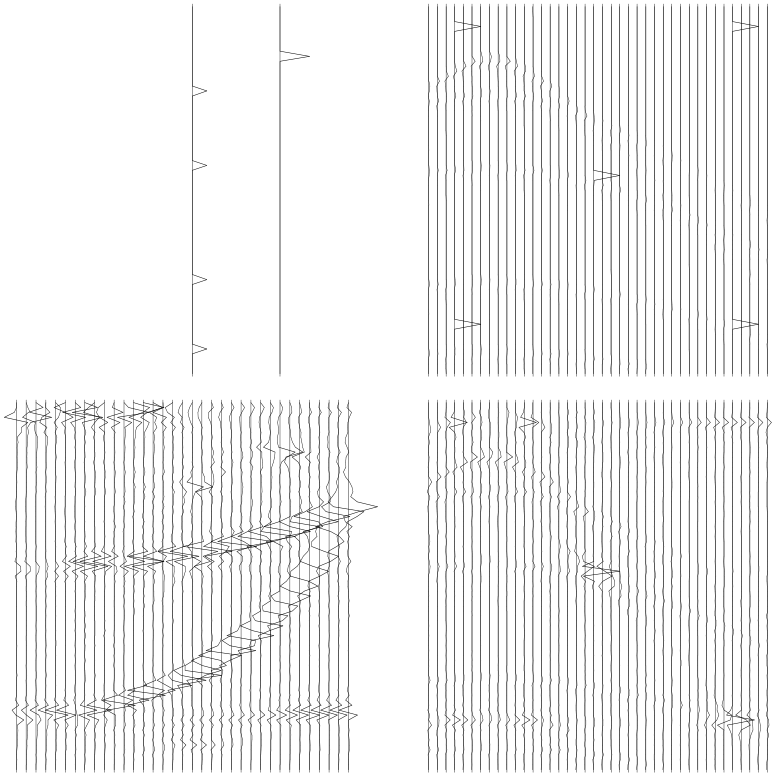

Figure 7: CG result with spiky events. antoine1-vel-spiky1g [CR]

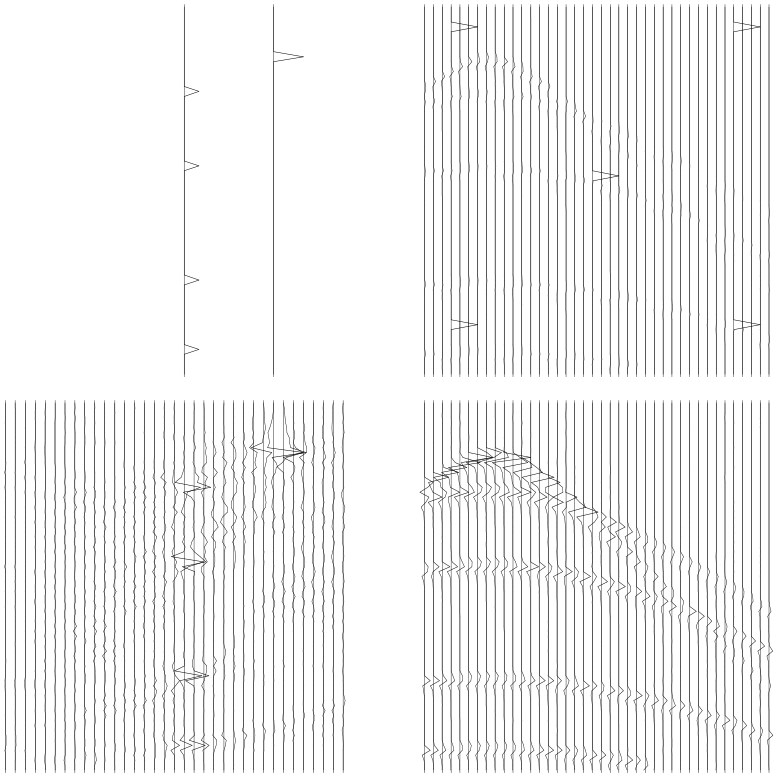

Figure 8: Huber result with spiky events. antoine1-vel-spiky2h [CR]

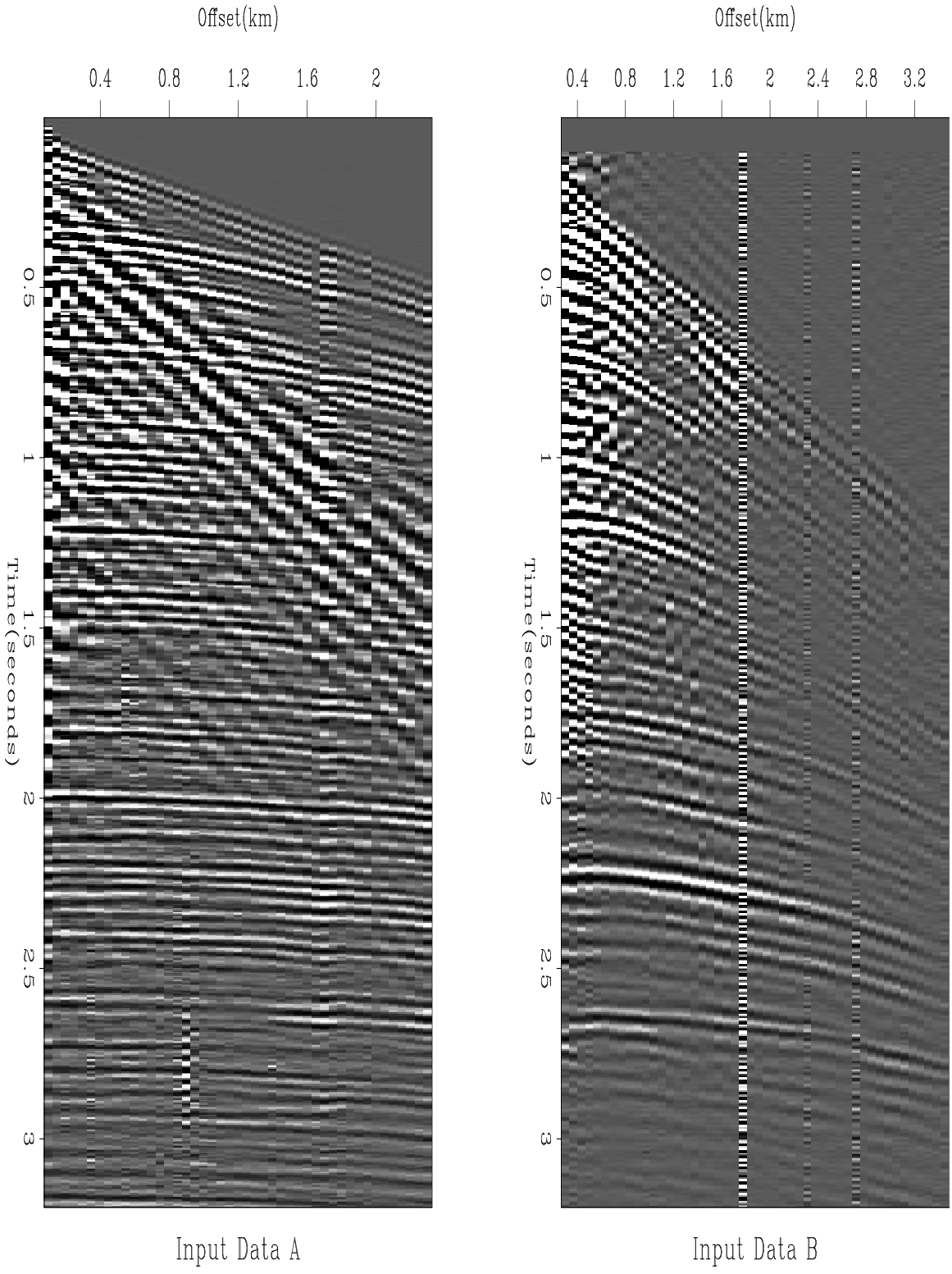

Figure 9: Input data- 2 CMP gathers. antoine1-datamodelreal [NR]

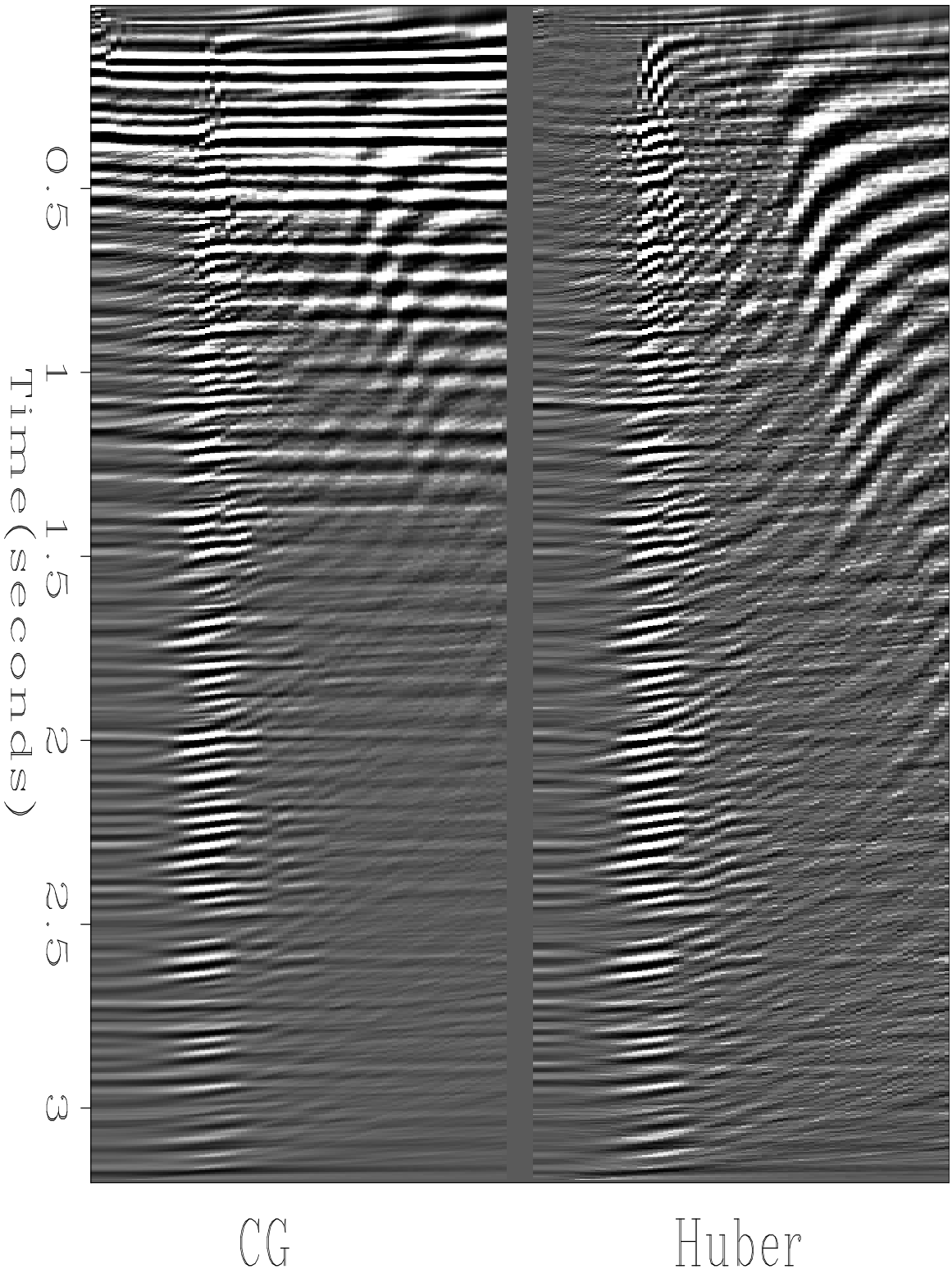

Figure 10: Velocity panel using CG and Huber solver: the left panel shows some stability problems above 1.5 sec. The Huber solver result shows a focused velocity corridor on the left and some artifacts on the right. Those artifacts are well separated from the fairway, however. antoine1-modelwz08 [CR]

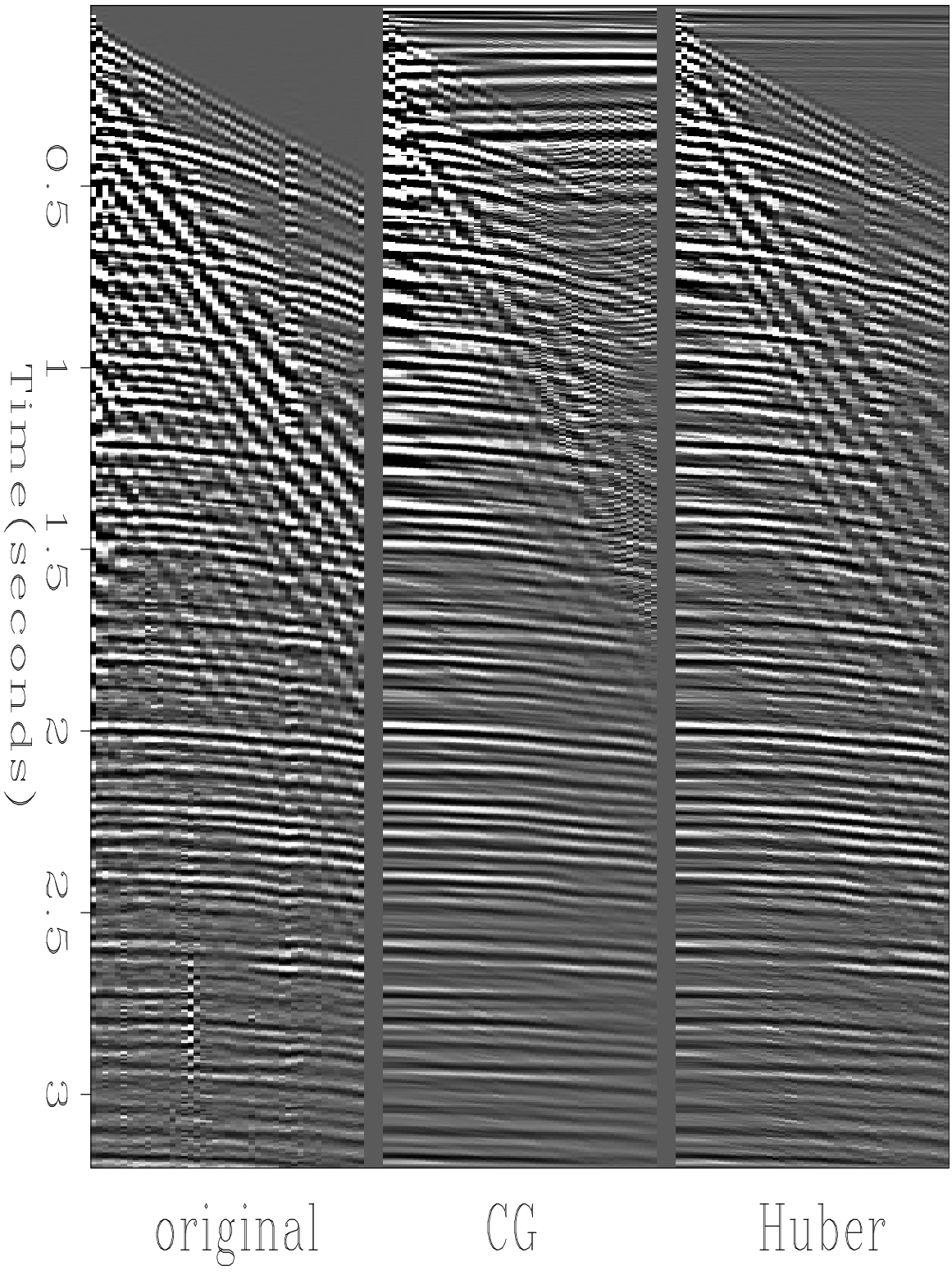

Figure 11: Data modeling after iteratively improved velocity model. antoine1-datawz08 [CR]

as expected, Huber allows to recover more accurately the original data. These results are quite encouraging and give us a flavor of what could be accomplished with the Huber norm. The same conclusion applies on the field data as on the synthetics: the Huber function is robust.

Is Huber a stable solver ?

For Huber to be a serious competitor to the l^2 norm and the CG, it needs to be stable with respect to the number of iterations and to the Huber threshold, ϵ. This is what we investigate in this section. Figure14 displays two velocity panels using 5 and 70 CG iterations on input *A*. The right panel shows one of the well known characteristics of the conjugate gradient: after a relative small number of iterations, the algorithm starts to invert noise in the data space, making the model space particularly messy and uninterpretable. In contrast, the Huber results are fairly stable (Figure15) since we can pick a reasonable velocity function without bothering with the noise. Some artifacts appear, however, which may be squelched by moving up or down the threshold. The next result is by far the most favorable to Huber (Input *B*). Figure16 shows two velocity functions after 5 and 70 iterations using conjugate gradient. The noise level is so strong after 70 iterations that we cannot distinguish any coherent feature. Once again, the Huber solver gives a better velocity panel (Figure 17). This is a major improvement to the performance of CG.

Testing the Huber response for different thresholds is another important issue. Remember that this threshold determines the border between the l^1 and l^2 norms. Ideally, we would like not to have to specify this parameter *a priori*, but rather have the algorithm adaptively estimates it from the data. It's interesting, however, to test the system's sensitivity to the threshold. Figure 18 shows a comparison for ϵ ranging from 0.001 to 1 for 20 iterations. The last panel is the CG result. As expected, Huber starts to behave like an l^2 norm for large ϵ. Nonetheless, for $\epsilon = 0.001$ to $\epsilon = 0.1$, the velocity panels are fairly comparable giving us a wide range of possibilities in the choice of the threshold.

Computational efficiency

Despite its general design intended for arbitrary optmization problem, the Limited Memory BFGS "off the shelf" code was successful in minimizing the Huber misfit function. The results achieved by the two algorithms are significantly different so that a direct comparison of cost is difficult. The Huber/BFGS combination tends to require about twice as much CPU time per iteration as the Least squares/CG.

CONCLUSION

Since geophysical inverse problems are often ill-posed due to the presence of inconsistent data, high amplitude anomalies and outliers, relative insensitivity to noise is a desirable characteristic of an inversion method. The Huber function is a compromise misfit measure

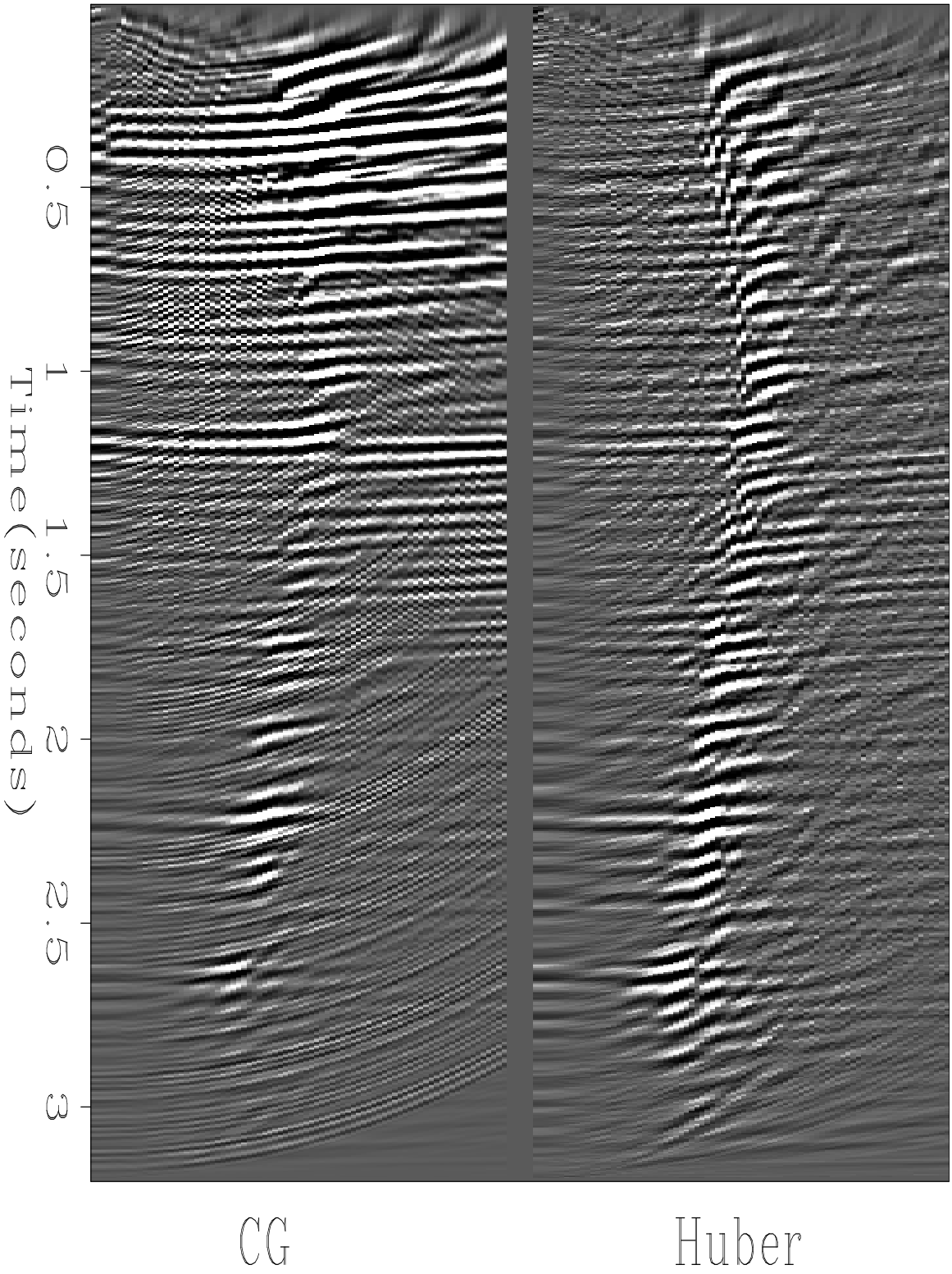

Figure 12: Velocity model using CG and Huber solver. antoine1-modelwz11 [CR]

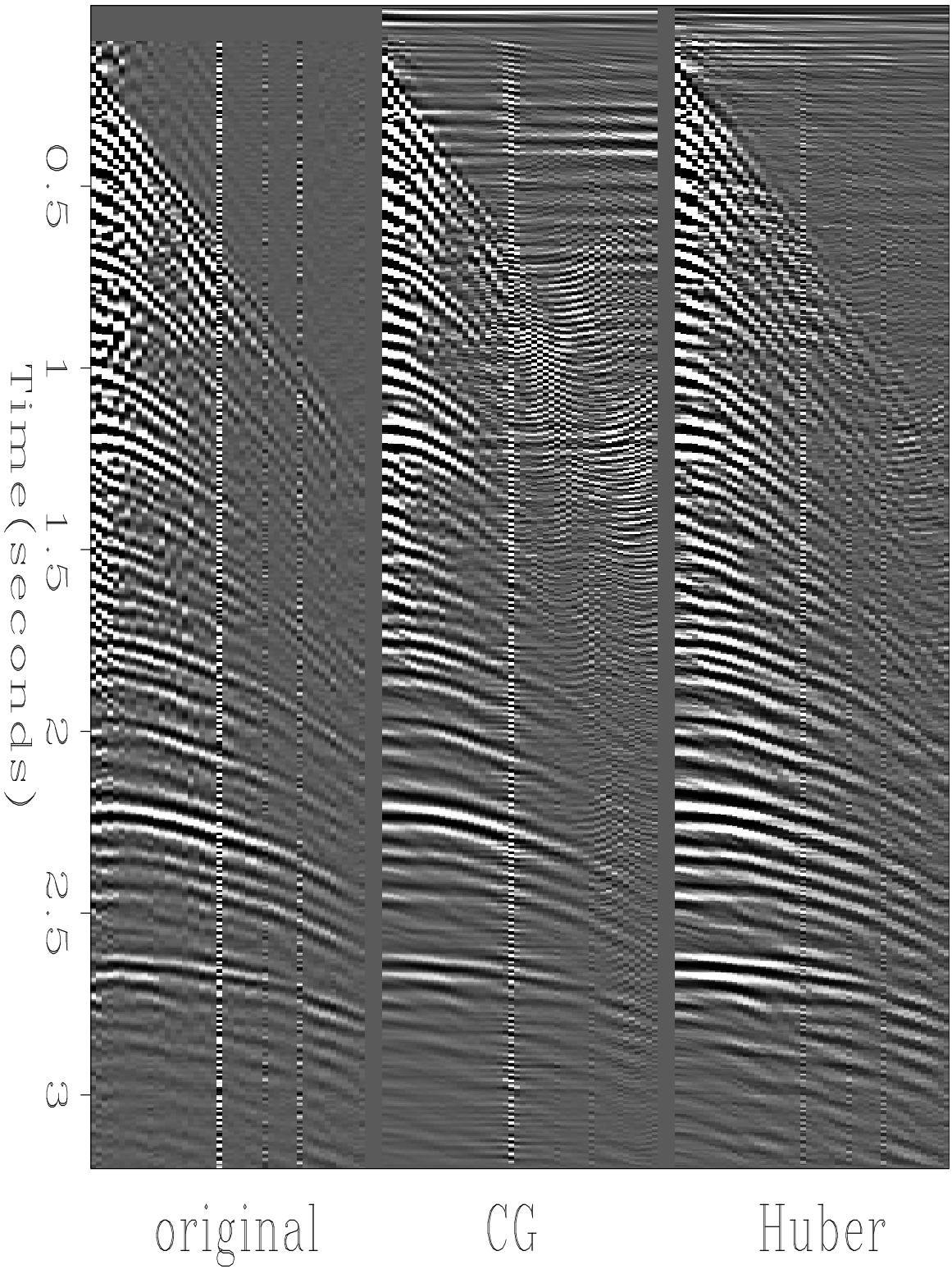

Figure 13: Data modeling after iteratively improved velocity model. antoine1-datawz11 [CR]

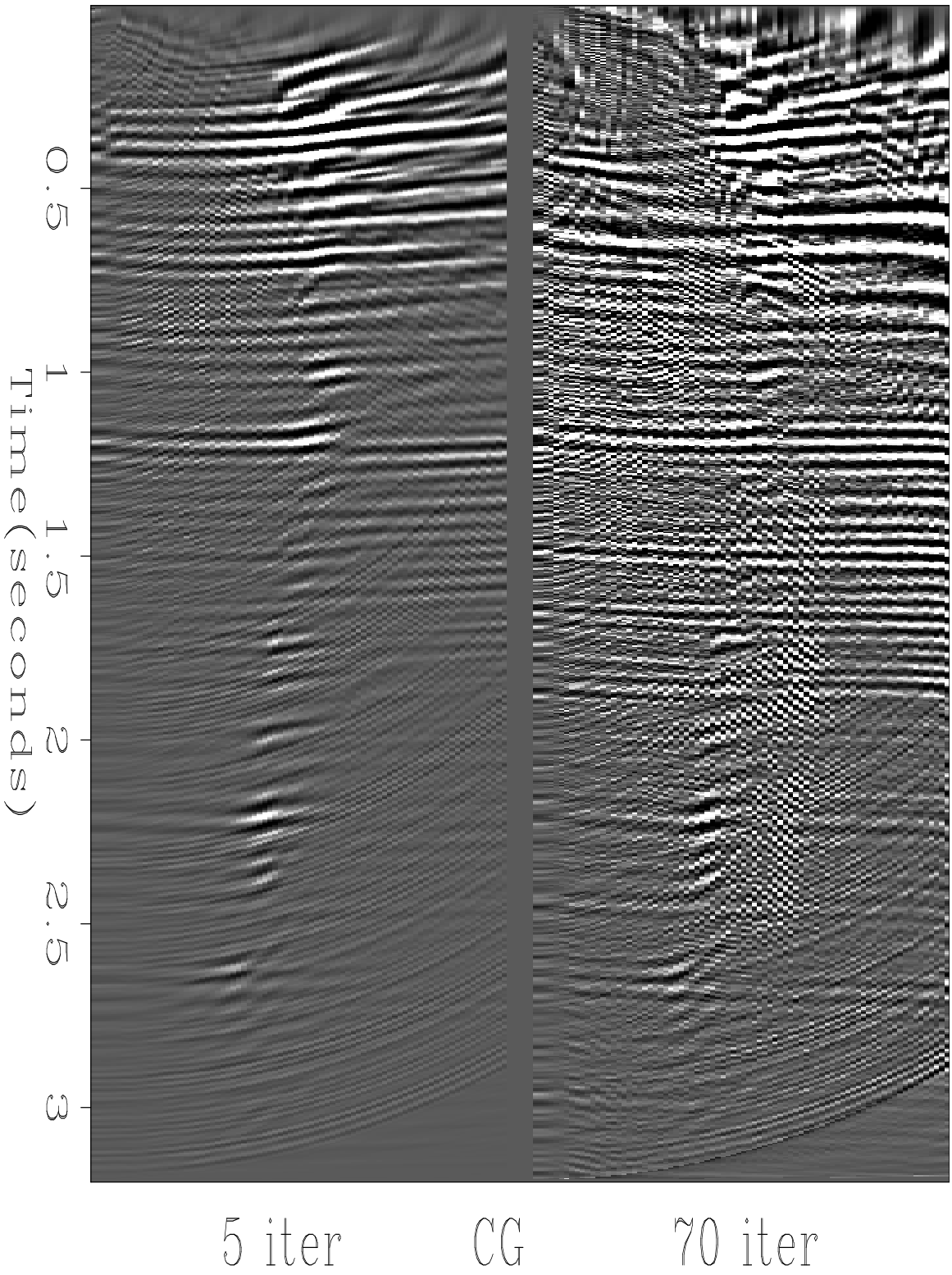

Figure 14: Input *B*: velocity panel using 5 and 70 CG iterations. antoine1-compmodel11g [NR]

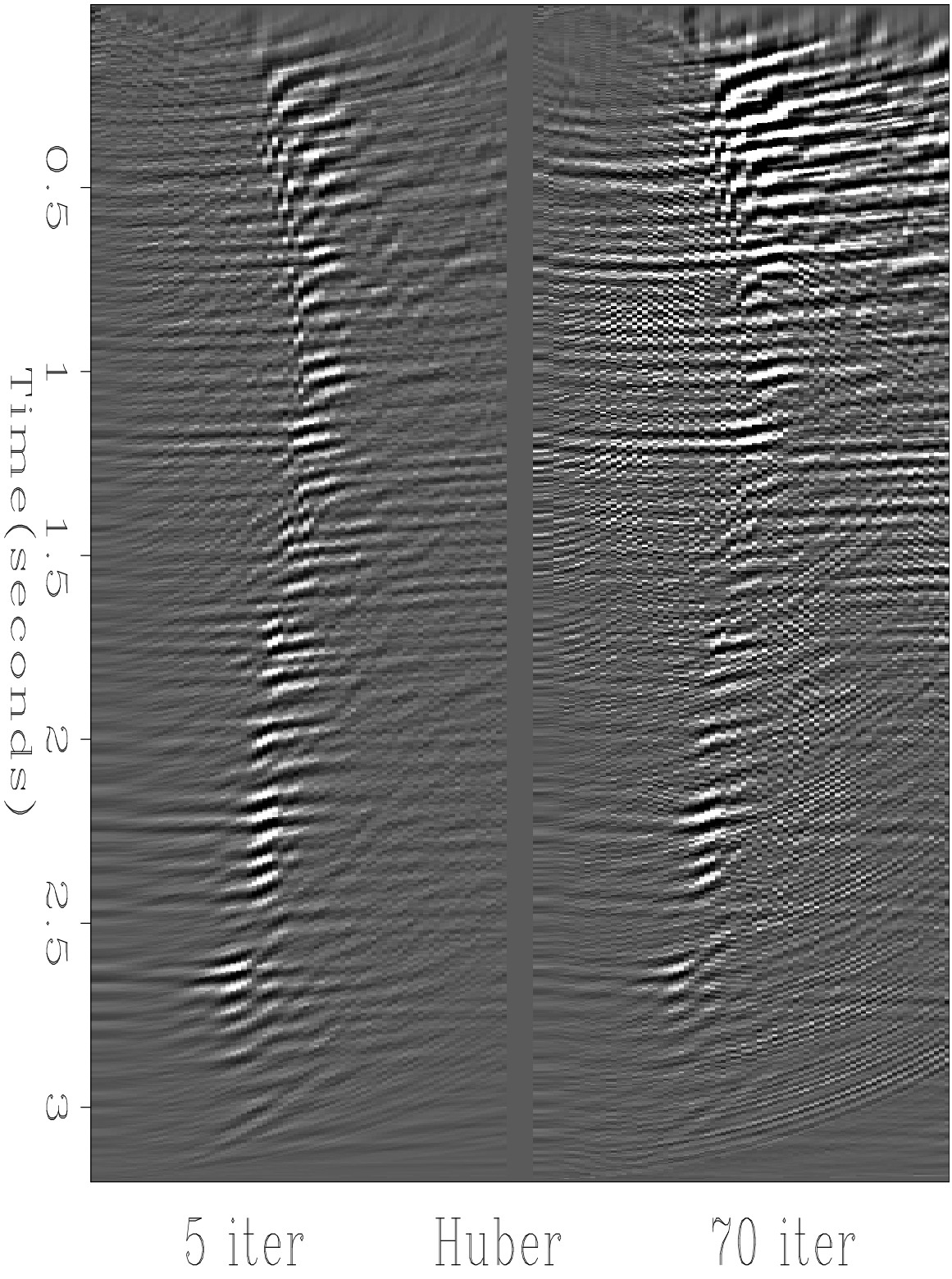

Figure 15: Input *B*: velocity panel using 5 and 70 Huber iterations. antoine1-compmodel11h [NR]

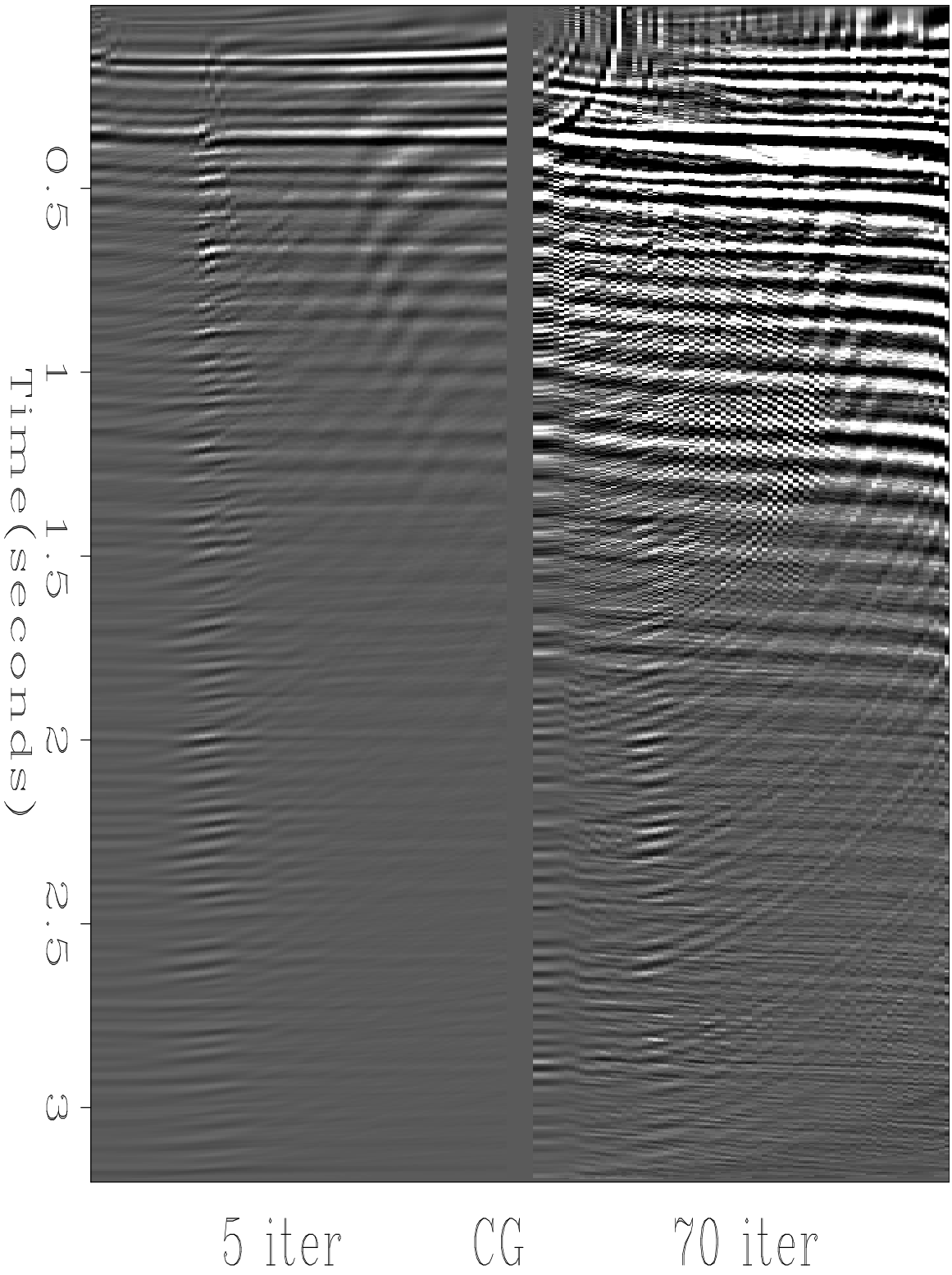

Figure 16: Input *A*: velocity panel using 5 and 70 CG iterations. antoine1-compmodel08g [NR]

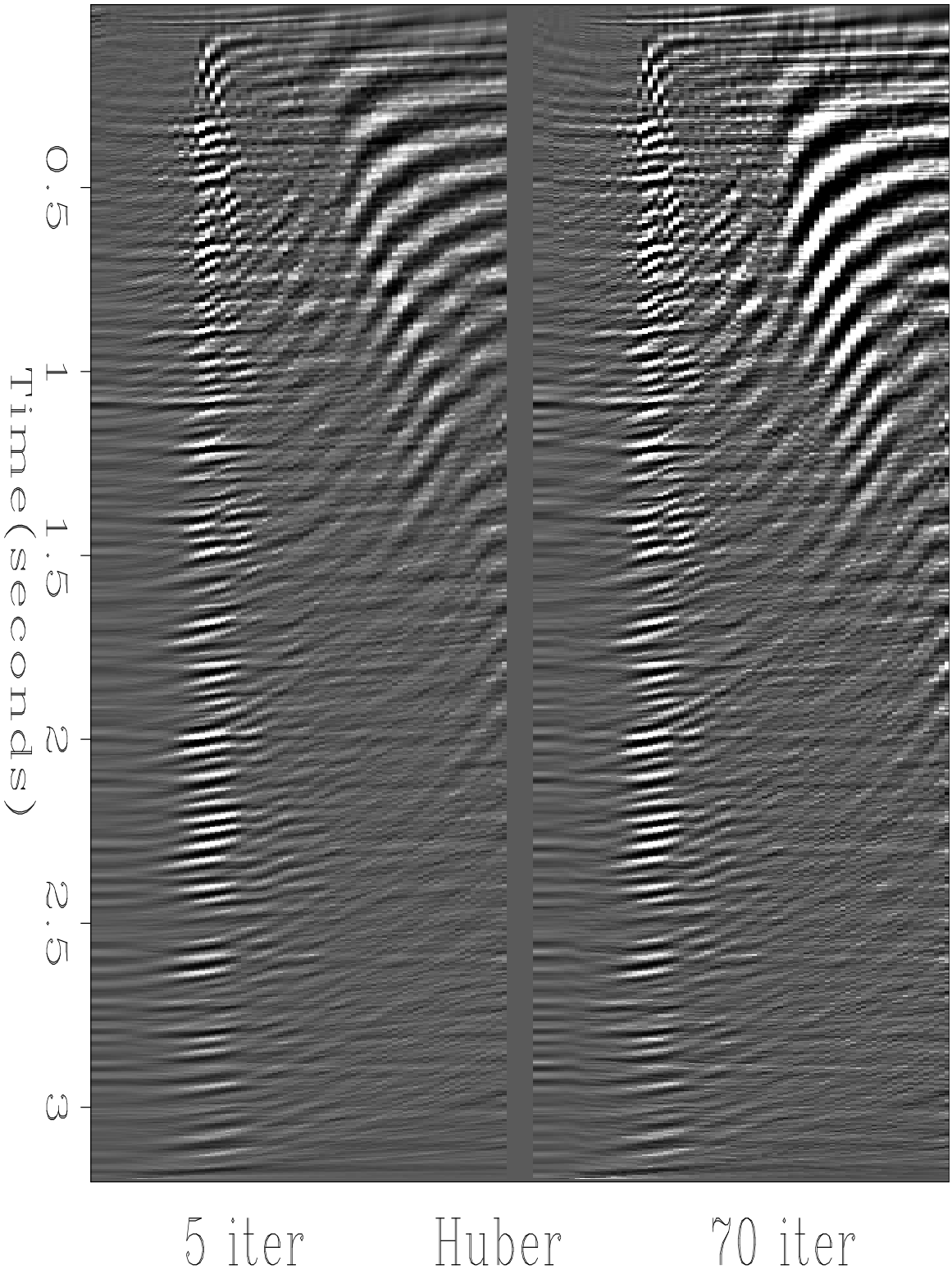

Figure 17: Input *A*: velocity panel using 5 and 70 Huber iterations. antoine1-compmodel08h [NR]

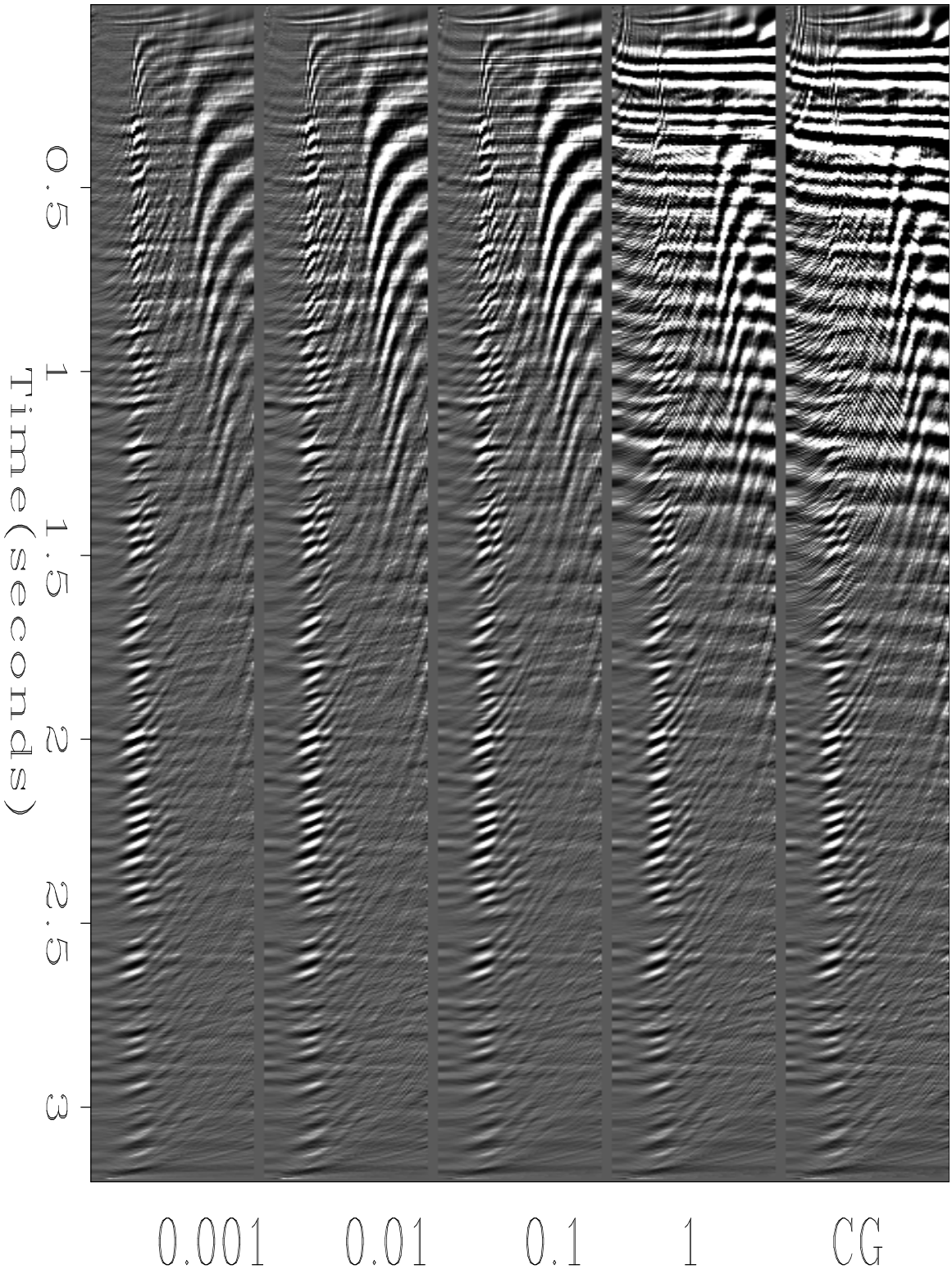

Figure 18: Input *A*: velocity panel for different Huber thresholds. The last panel is obtained using the CG. antoine1-treshcomp08 [NR]

between l^1 and l^2 norms, not only boasting robustness in the presence of noise and outlier effects like l^1 measures, but also smoothness for small residuals characteristic of l^2 measures. The transition between the two norms is governed by a free parameter, the Huber threshold ϵ.

The Huber solver is fairly stable with respect to two major choices: the number of iterations and ϵ. The most striking result arises when we increase the number of iterations: while the l^2 result explodes, the Huber result looks stable. In addition, we may choose a threshold within a large range without degrading the estimated velocity model (once ϵ is small enough). We did not apply any regularization on the least squares method: it would make l^2 less noise-sensitive but requires either a regularization weight or a noise level estimate and results are rather sensitive to these. The Huber function also requires an estimate for the parameter ϵ, but the results seem not to depend strongly on its choice. Furthermore, the Huber function gives better results than the l^2 when applied to velocity analysis showing its robustness to outlier effects. A data-dependent criterion for choosing the Huber threshold may prove fruitful, i.e., "treat X% of the data as Gaussian in the small residuals", where X is specified interactively by the end-user.

These results encourage the use of the Huber function whenever the data are contaminated with noise and, as a robust and stable measure, to replace the l^2 norm in many geophysical applications.

ACKNOWLEDGEMENTS

The authors are grateful to Louis Vaillant and Jon Claerbout for their help and fruitful suggestions.

REFERENCES

Barrodale, I., and Roberts, F. D. K., 1980, Algorithm 552 : Solution of the constrained l1 linear approximation problem: ACM Transactions on Mathematical Software, **6**, 231–235.

Chapman, N. R., and Barrodale, I., 1983, Deconvolution of marine seismic data using the l1 norm: Geophys. J. Roy. Astr. Soc., **72**, 93–100.

Claerbout, J. F., and Black, J. L., 1997, Basic earth imaging: Class notes, http://sepwww.stanford.edu/sep/prof/index.html.

Claerbout, J. F., and Muir, F., 1973, Robust modeling with erratic data: Geophysics, **38**, 820–844.

Ekblom, H., and Madsen, K., 1989, Algorithms for non-linear huber estimation: BIT, **29**, 60–76.

Fletcher, R., 1980, Practical methods of optimization, I:Unconstrained Optimization: Wiley & Sons, New York.

Gockenbach, M. S., Petro, M., and Symes, W. W., 1999, C++ classes for linking optimization with complex simulations: ACM Transactions on Mathematical Software, in press.

Huber, P. J., 1973, Robust regression: Asymptotics, conjectures, and Monte Carlo: Ann. Statist., **1**, 799–821.

Nocedal, J., 1980, Updating quasi-Newton matrices with limited storage: Mathematics of Computation, **95**, 339–353.

Scales, J. A., and Gersztenkorn, A., 1988, Robust methods in inverse theory: Inverse Problems, **4**, 1071–1091.

Scales, J. A., Gersztenkorn, A., and Treitel, S., 1988, Fast lp solution of large, sparse, linear systems: application to seismic travel time tomography: J. Comp. Phys., **75**, 314–333.

Taner, M., and Koehler, F., 1969, Velocity spectra: digital computer derivation and application of velocity functions: Geophysics, **34**, 859–881.

Tarantola, A., 1987, Inverse problem theory: Elsevier.

Taylor, H. L., Banks, S. C., and McCoy, J. F., 1979, Deconvolution with the l1 norm: Geophysics, **44**, 39–52.