Stanford Exploration Project, Report 77, November 17, 1997, pages 261–274

Steep-dip deconvolution

Jon F. Claerbout[1]

keywords: *deconvolution, velocity, one-dimensional, noise, gain*

ABSTRACT

I design two-dimensional filters for deconvolving field profiles. In the spatial direction, filter coefficients are constrained to lie within a triangle defined by water velocity. (The filters were computed and applied within windows and outputs are then patched together.) I selected six Yilmaz-Cumro field profiles with low velocity noises for demonstration testing. All results were good to excellent.

INTRODUCTION

Low-velocity noises on shot records are often not fully suppressed by stacking because the noises are spatial aliased. Routine field arrays are not perfect and the noise is often extremely strong. Chapter 8 (Missing Data) in my textbook, (Claerbout, 1992) shows that two-dimensional filters have some "magical" properties regarding aliasing. With this and (f, x)-decon in mind I set out to design some data-adaptive anti-noise filters. It turned out to be much easier than I expected and on the first try I got good results on all all six field profiles tested. I have not yet tweaked the many adjustable parameters. As you inspect these deconvolved profiles from different areas of the world with different recording methods, land and marine, think about how the stacks should be improved by the deconvolution.

THEORY AS RELATED TO EARLIER WORK

Normally an autoregression filter predicts a value at a point using values at earlier points. In practice a gap may also be set between the predicted value and the earlier values. What is not normally done is to supplement the regression by nearby signals. That is what I did here. I allowed the prediction of a signal to include nearby signals at earlier times. The times I allowed in the regression are inside a triangle of velocity less than about the water velocity. The new information allowed in the prediction

[1]**email:** not available

is extremely valuable for water velocity events because looking along a trajectory at water velocity slope, water noise is transformed to zero-frequency and hence is easily predicted.

The overall process proceeds independently in each of many overlapping windows. At the end, filter outputs are pieced together as described in (Claerbout, 1992b). In each window there are two-computational phases, first filter design, and then filter application. Application is simply two dimensional filtering. Filter design restricts nonzero filter coefficients to a triangular (noise) region. As with usual deconvolutions, one filter coefficient is constrained to unity while others are either constrained to zero or they are adjusted to minimize the power out. The coefficients of the filter are easily determined by the conjugate-gradient methodology described in (Claerbout, 1992).

BASIC ASPECTS OF TWO-DIMENSIONAL FILTERING

Consider the two-dimensional filter

```
    +1
-1   0 -1
    +1
```

This filter applied to a field profile with 4 milliseconds time sampling and 6 meter trace spacing should perfectly extinguish 1.5 km/s water-velocity noises. Likewise, the filter

```
    +1
     0
     0
-1   0 -1
     0
     0
    +1
```

should perfectly extinguish water noise when the trace spacing is 18 meters. Such noise is, of course, spatially aliased for all temporal frequencies above 1/3 of Nyquist, but that does not matter. The filter extinguishes them perfectly anyway. Inevitably, the filter cannot both extinguish the noise and leave the signal untouched where the alias of one is equal the other. So you expect signal to be altered where it matches aliased noise. This simple filter does worse than that. On horizontal layers, for example, signal wavelets become filtered by $(1, 0, 0, -2, 0, 0, 1)$. If the noise is overwhelming, this signal distortion is a small price to pay for eliminating it. If the noise is tiny, however, the distortion is unforgivable. In the real world, data-adaptive deconvolution is often a good compromise.

The two-dimensional deconvolutions filters I use look like this:

```
x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x
.  x  x  x  x  x  x  x  .
.  x  x  x  x  x  x  x  .
.  x  x  x  x  x  x  x  .
.  .  x  x  x  x  x  .  .
.  .  x  x  x  x  x  .  .
.  .  x  x  x  x  x  .  .
.  .  .  x  x  x  .  .  .
.  .  .  x  x  x  .  .  .
.  .  .  x  x  x  .  .  .
.  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .
.  .  .  .  1  .  .  .  .
```

where each "." denotes a zero and each "x" denotes a (different) adjustable filter coefficient that is chosen to minimize the power out.

You can easily imagine variations on this shape, such as diamond instead of triangle. I did not have time to experiment with the various shapes that suggest themselves.

WHICH COEFFICIENTS ARE REALLY NEEDED?

Steep-dip decon is a heavy consumer of computer time. Many small optimizations could be done, but more importantly, I feel there are some deeper issues that warrant further investigation. The first question is how many filter coefficients should there be and where should they be? We would like to keep the number of nonzero filter coefficients to a minimum because it would speed the computation, but more importantly I fear the filter output might be defective in some insidious way (perhaps missing primaries) when too many filter coefficients are used. Perhaps if 1-D decon were done sequentially with steep-dip decon the number of free parameters (and hence the amount of computer time) could be dropped even further. I looked at some of the filters and they scatter wildly with the Nyquist frequency (particularly those coefficients on the trace with the "1" constraint). This suggests using a damping term on the filter coefficients after which perhaps the magnitude of a filter coefficient will be a better measure of whether it is really helpful. Also, it would, of course, be fun to get some complete data sets (rather than a single shot profile) to see the difference in the final stack.

FIELD DATA TESTS

I scanned the forty Yilmaz and Cumro shot profiles for strong low-velocity noises and I selected six examples. To each I applied an AGC that is a slow function of time and space (triangle smoothing windows with triangle half widths of 200 time points and 4 channels). Since my process simultaneously does both low velocity rejection and deconvolution I prepared more traditional 1-D deconvolutions for comparison. This is done in windows of 250 time points and 25 channels, the same filter being used for each of the 25 channels in the window. In practice, of course, considerably more thought would be given to optimal window sizes as a function of the regional nature of the data. The windows were overlapped by about 50%. The same windows are used on the steep-dip deconvolution.

Unexpectedly, results in Figures 1-6 show that the conventional 1-D deconvolution also suppresses low velocity noises. An explanation can be that these noises are often either low frequency or quasimonochromatic.

Filters are never allowed off the edges of their application windows. Thus on all the deconvolutions at early times you see zero valued output of duration equal the filter length. Likewise, on the steep-dip deconvolutions, you see some missing output traces along the sides. The number of missing traces matches the halfwidth of the filter. These missing outputs may be unacceptable in a production environment. Fundamentally, my process cannot work ideally along the edges, but with a little extra coding, nonzero but nonideal outputs could be produced there. More drastically, you can imagine spatially one-sided filters like

```
x  x  x  x  x
x  x  x  x  x
.  x  x  x  x
.  x  x  x  x
.  .  x  x  x
.  .  x  x  x
.  .  .  x  x
.  .  .  x  x
.  .  .  .  .
.  .  .  .  .
.  .  .  .  1
```

being applied on one side of the shot with the opposite orientation being applied on the other side. Such filters would not have missing outputs at wide offset, but would have an inner offset problem. The advantage of a double sided filter is that it should be able to extinguish backscattered noises as well as direct noises. We see this effect on Figure 4, but unfortunately, there are no interesting signals beneath the backscattered noises. I did not test the spatially one-sided filters, though it is simply a matter of changing a parameter in the existing code.

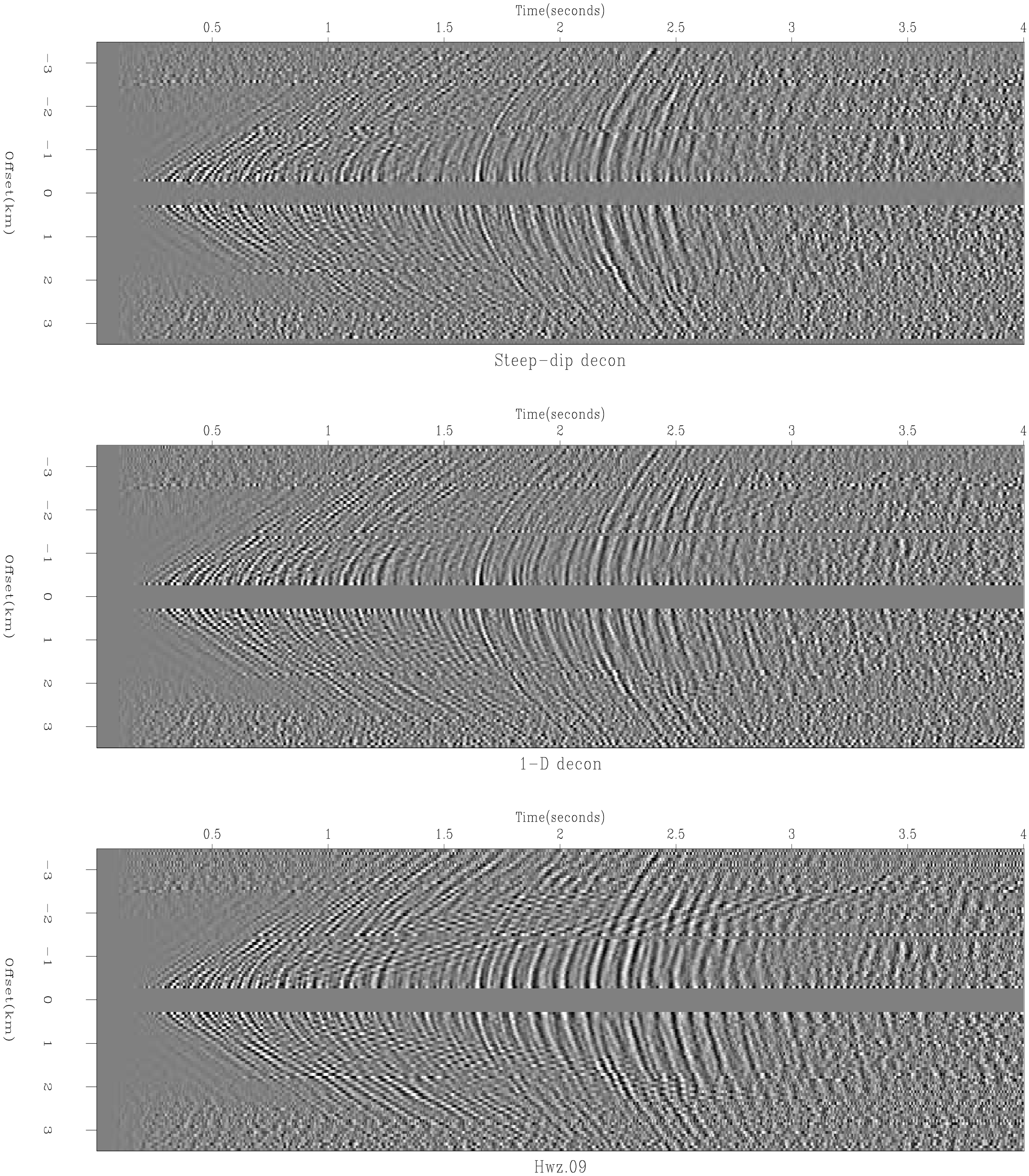

Figure 1: Bottom is an Alaska vibrator shot profile (Y&C #09) after AGC. Middle is gapped 1-D decon. Top is steep-dip decon. jon2-wz.09 [CR]

ARE FIELD ARRAYS REALLY NEEDED?

Field arrays cancel random noise but their main function, I believe, is to cancel low-velocity coherent noises, something we now see is handled effectively by steep-dip deconvolution. While I do not advocate abandoning field arrays, it is pleasing to notice that with the arrival of steep-dip deconvolution, we are no longer so dependent on field arrays and perhaps coherent noises can be controlled where field arrays are impractical, such as certain 3-D geometries.

REFERENCES

Claerbout, J. F., 1992a, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Scientific Publications.

Claerbout, J. F., 1992b, Nonstationarity and conjugacy: Utilities for data patch work: SEP–**73**, 391–400.

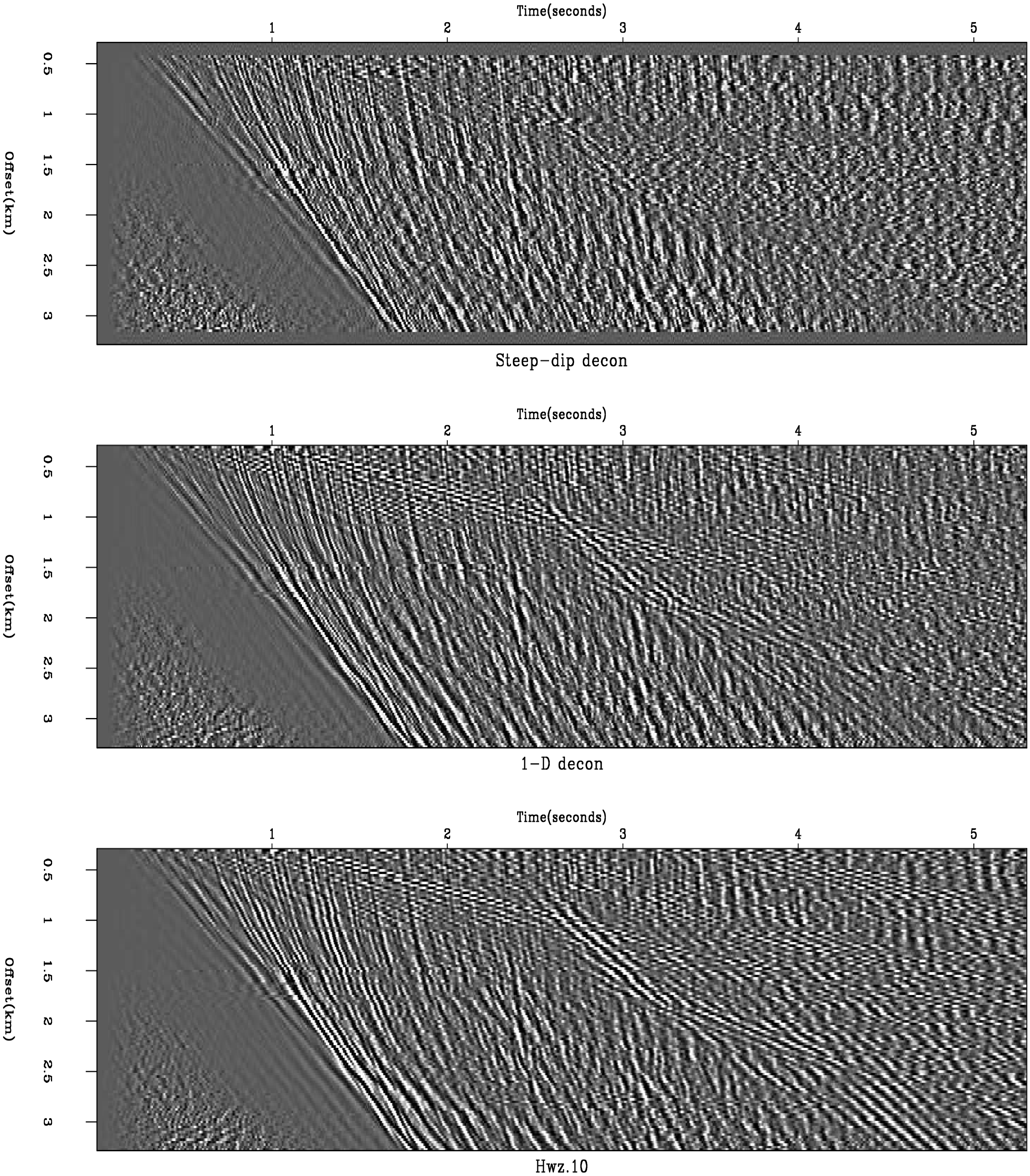

Figure 2: Bottom is a North African vibrator shot profile (Y&C #10) after AGC. Middle is gapped 1-D decon. Top is steep-dip decon. jon2-wz.10 [CR]

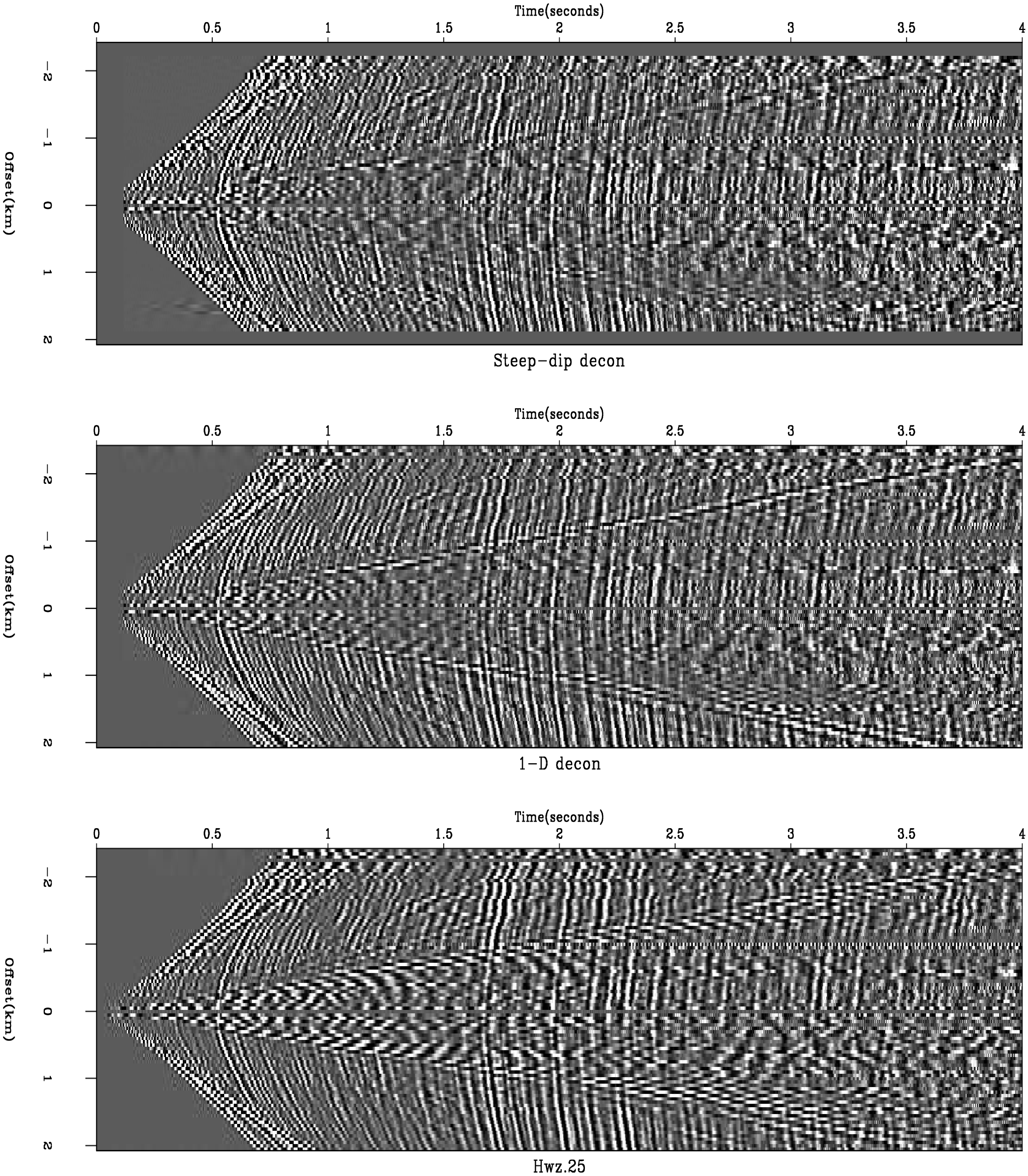

Figure 3: Bottom is an Alberta dynamite shot profile (Y&C #25) after AGC. Middle is gapped 1-D decon. Top is steep-dip decon. jon2-wz.25 [CR]

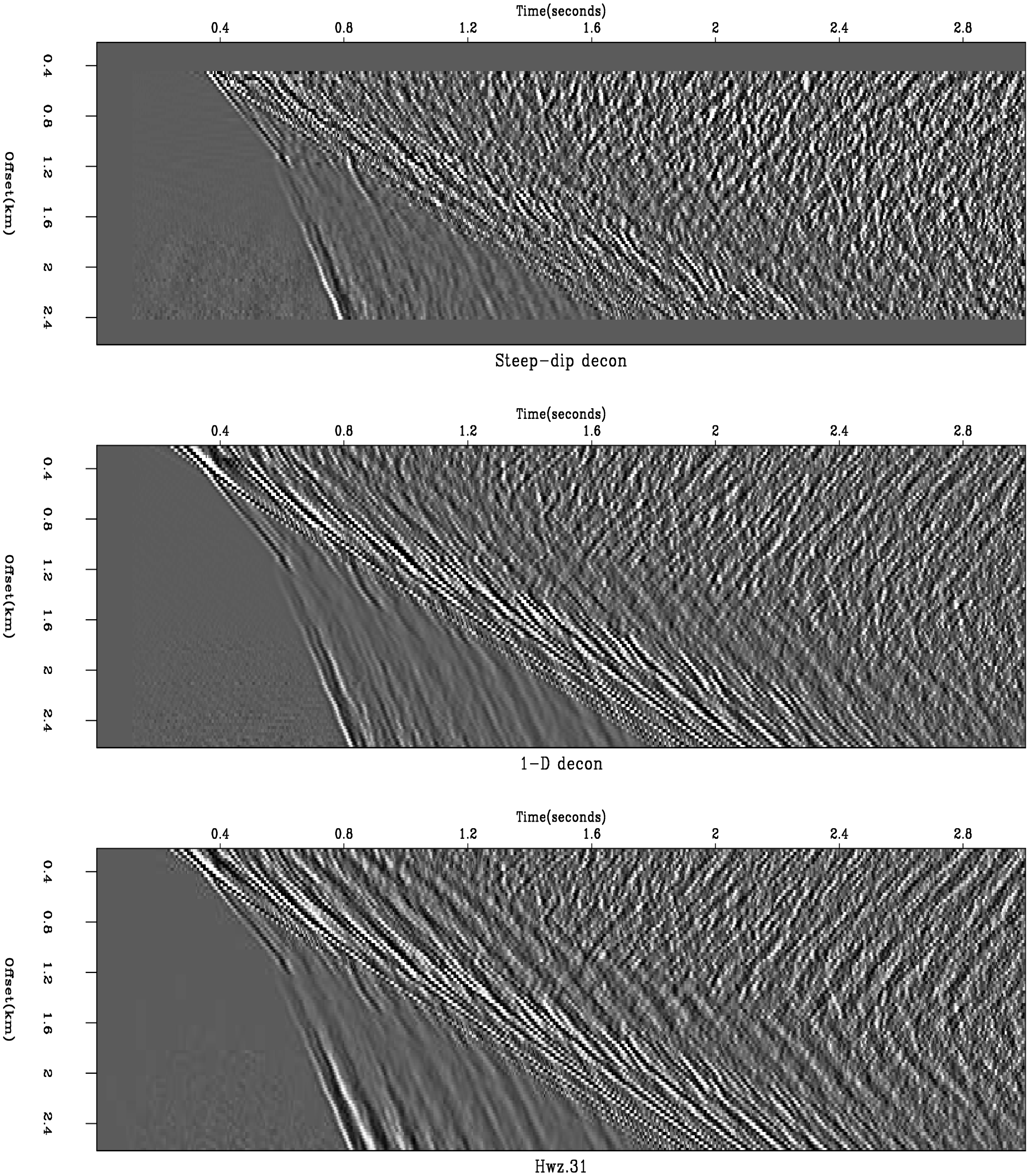

Figure 4: Bottom is a North Sea air gun shot profile (Y&C #30) after AGC. Middle is gapped 1-D decon. Top is steep-dip decon. $\boxed{\text{jon2-wz.31}}$ [CR]

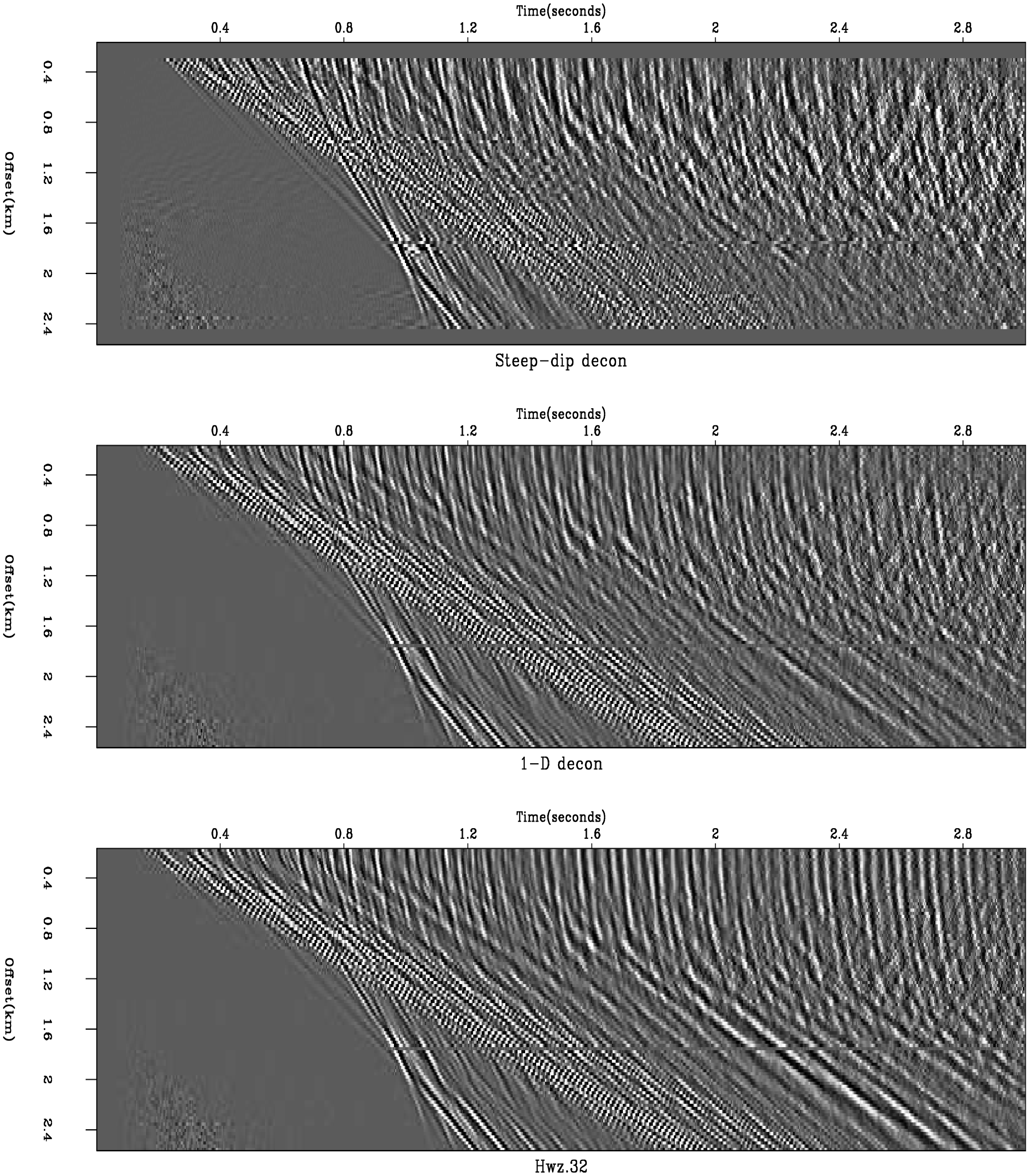

Figure 5: Bottom is a North Sea air gun shot profile (Y&C #31) after AGC. Middle is gapped 1-D decon. Top is steep-dip decon. jon2-wz.32 [CR]

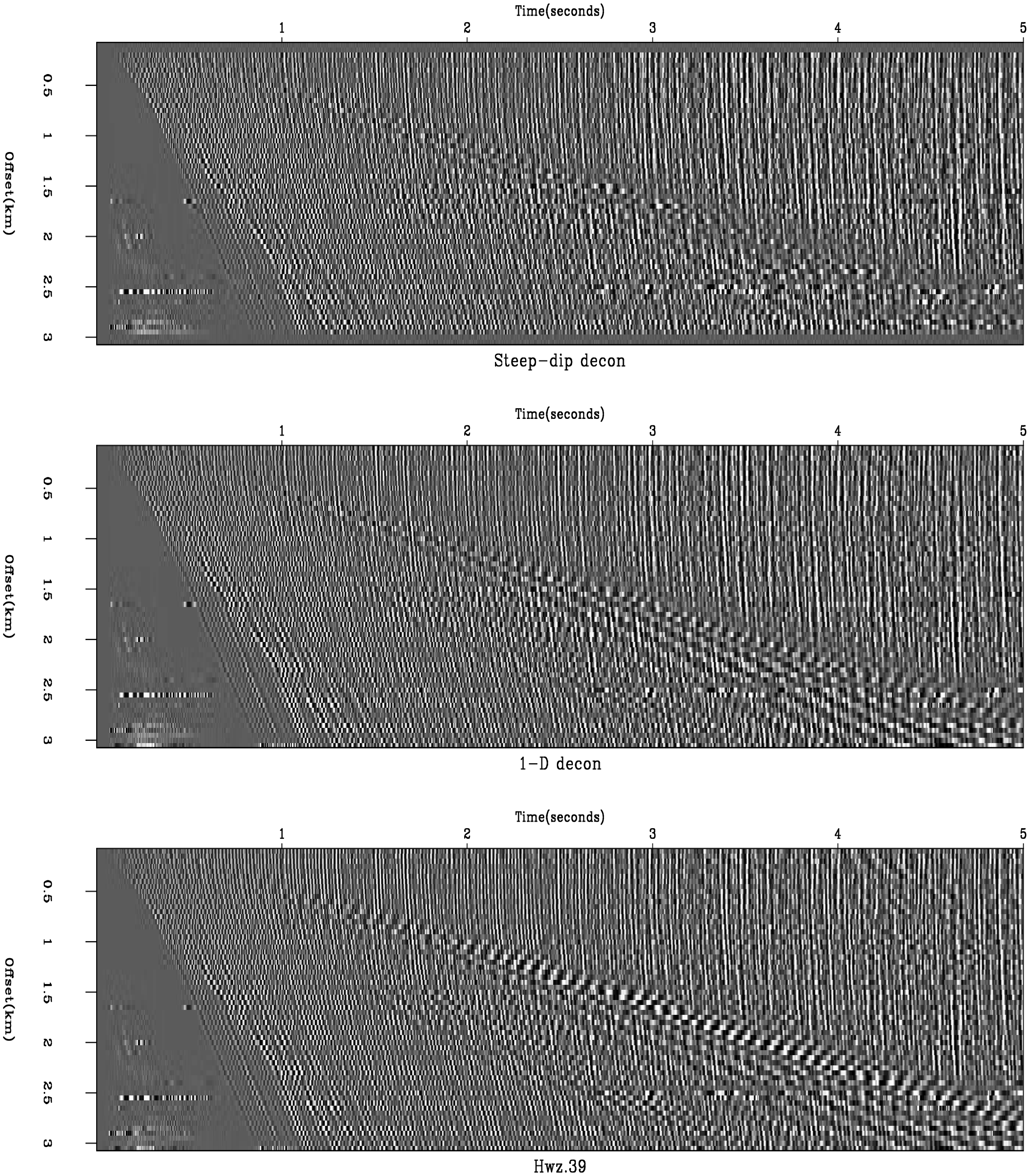

Figure 6: Bottom is a Middle east Geoflex shot profile (Y&C #39) after AGC. Middle is gapped 1-D decon. Top is steep-dip decon. jon2-wz.39 [CR]