Imaging using compressive sensing

Robert G. Clapp

ABSTRACT

Constructing 3-D angle gathers is quickly becoming the computational bottleneck in Reverse Time Migration (RTM). Both the expansion in volume size (120-900x) and the cost of cross-correlating wavefields greatly increase the cost of RTM. Both the storage and computational cost can be greatly reduced by reformulating the imaging process as a compressive sensing problem. Preliminary results show that this approach holds promise but finding an acceptable L-1 inversion approach is still problematic.

INTRODUCTION

Reverse Time Migration is quickly becoming the standard high-end imaging technique. The last few years has seen numerous papers on how to speed up the finite-difference kernel on various platforms (Micikevicius, 2009; Nguyen et al., 2010; Nemeth et al., 2008; Clapp et al., 2010). These techniques was sufficient when RTM was simply used to produce a final image.

When RTM is used to construct angle gathers for velocity or rock property analysis, the finite difference kernel becomes a secondary concern. The construction of angle gathers, particularly 3-D angle gathers, through sub-surface offset correlation (Sava and Fomel, 2006) or time-shift gathers (Sava and Fomel, 2003) becomes the dominant cost. Some have proposed reducing the cost of 3-D angle gathers by constructing angle gathers along only a few azimuths. While these techniques are significantly less costly than full 3-D angle gathers, they are still expensive and not ideal. Compressive sensing (Donoho, 2006) offers a potential solution to this computation and storage problem. In compressive sensing, a random sub-set of the desired measurements are made. An inversion problem is then set up to estimate in an ℓ_1, or preferably ℓ_0, sense, a sparse basis function that fully characterizes the desired signal. For compressive sensing to work, a signal must be highly compressible. For compressive sensing to be worthwhile, the cost of inverting for the basis function must be significantly less than the cost of acquiring the full signal.

In this paper, I show how angle gather construction fits the criteria for compressive sensing. I demonstrate how angle gathers are highly compressible in the multi-dimensional wavelet domain. Further, I demonstrate how the cost of constructing a sub-set of the sub-surface offsets and then performing an ℓ_1 inversion is significantly less expensive.

COMPRESSIVE SENSING

Compressive sensing is a statistical technique whose start is usually traced back to Donoho (2006), but whose start could be placed as early as the basic pursuit work of Mallat and Zhang (1993). A compressive sensing problem at its heart is a special case of a missing data problem. In geophysics, we often think of a missing data problem as solving for a model $\mathbf{m}$ given some data $\mathbf{d}$ which exist in the same vector space. We have a masking operator $\mathbf{M}$ (1 where the data is known, 0 elsewhere). We add in some knowledge of the covariance of the model through a regularization operator $\mathbf{A}$. We then estimate the best model from the following system of equations in a ℓ_2 sense,

$$\begin{aligned} \mathbf{0} \approx \mathbf{r_d} &= \ell_\mathbf{2}(\mathbf{d} - \mathbf{Mm}) \\ \mathbf{0} \approx \mathbf{r_m} &= \ell_\mathbf{2}(\mathbf{Am}), \end{aligned} \tag{1}$$

where $\mathbf{r_d}$ and $\mathbf{r_m}$ are the result of taking the ℓ_2 norm of the first and second equations. The success of this approach relies on the accuracy of $\mathbf{A}$ to describe the covariance of the model.

Compressive sensing approaches the problem from a different perspective. It starts from the notion that there exists a basis function that $\mathbf{d}$ can be transformed into through the linear operator $\mathbf{L}'$ in which very few non-zero elements are needed to represent the signal. The compressive sensing approach is then to set up the missing data problem in two phases. First, estimate the elements of the sparse basis function $\mathbf{m}$ through,

$$\mathbf{0} \approx \mathbf{r} = \ell_\mathbf{1}(\mathbf{d} - \mathbf{MLm}), \tag{2}$$

where we are now estimating $\mathbf{m}$ in the $\ell_\mathbf{1}$ sense. We can then apply $\mathbf{L}$ to recover the full model. Compressive sensing makes the claim that if the signal can be represented in this basis function by n points, if you collect bm *randomly sampled* points, where b is typically 4-5, significantly less than what the Nyquist-Shannon criteria would suggest.

Figure 1 shows an example of this technique applied to a 2-D missing data problem. In this case, we are trying to recover the image seen in Figure 1(a). We start from the data points seen in Figure 1(b), and recover the image seen in Figure 1(c). In this case, the image was subsampled by a factor of 8 and nearly perfect recovery was achieved.

Reconstruction using compressive sensing techniques is expensive. ℓ_1 solvers are significantly more expensive than their ℓ_2 counterparts, which in themselves can represent a significant cost. As a result, only certain classes of problems benefit from compressive sensing techniques. For compressive sensing to be useful, the cost of acquiring the full dataset must be significant. In addition, the signal must be highly compressible. The following two sections address both criterion.

(a) (b) (c)

Figure 1: The original image (a), the image sub-sampled by a factor of 8 (b), and the reconstructed image (c) using compressive sensing. Baraniuk et al. (2008). [**NR**]

IMAGING CONDITIONS COSTS IN RTM

There are many different imaging condition choices in RTM, that can have dramatically different cost profiles depending on the compute engines' memory structure and size, along with implementation method. All RTM imaging conditions start from a source $s(x, y, z, t)$ and receiver $g(x, y, z, t)$ wavefields. Generally some kind of correlations of these two wavefields is done to produce the RTM image. The data handling problem becomes challenging because the source wavefield is propagated forward in time while the receiver wavefield is propagated backward in time. As a result, one of the wavefields is stored to disk, or a checkpointing or random boundary condition is applied.

Until recently, a zero-time, zero-subsurface offset imaging condition was the most commonly implemented method. This amounts producing an image $I(x, y, z)$ by

$$I(x, y, z) = \sum_{\text{shot}} \sum_{t} s(x, y, z, t) g(x, y, z, t). \tag{3}$$

With this imaging condition, we normally store $I(x, y, z)$ at a relatively low cost memory hierarchy (it exists in RAM on a CPU, in global memory in the GPGPU, or attached RAM on a FPGA). For simplicity, let's assume that $g(x, y, z, t)$ is written to disk in a non-compressed form and then read backwards during the source propagation. Given that we often need to propagate six to eight time steps between imaging steps due to stability constraints, to avoid an IO bottleneck we need the cost of reading the receiver wavefield to be 1/6 the cost of our finite difference stencil. As of this writing, we are nearing a cross-over point where this will no longer be valid.

The relative cost of implementing the actual imaging condition is a little trickier. For this exercise, let's assume that we are implementing acoustic RTM on a GPGPU (the relatively simple memory model simplifies the calculation). Let's assume a naive second-order in time, 14th order in space approach. In this case, at each sample we need to read in the previous wavefield, the velocity, and store the updated wavefield.

In addition, we will access 43 values in the current wavefield, but because these are stored in shared memory they will have approximately $\frac{1}{10}$ latency. Therefore, the cost of propagating the wavefield eight times between imaging steps will be $nx * ny * nz * 8 * (3 + 43 * .1)$. The imaging step at each point will require the reading of the source and receiver wavefield from global memory, and the reading and writing of the image, a cost of $nx * ny * nz * 4$, or a magnitude less than the propagation cost.

Alternate imaging condition choices such as offset-gathers(Sava and Fomel, 2003), time-shift gathers(Sava and Fomel, 2006), and extended image gathers(Sava, 2007), dramatically alter this balance. For conciseness I will limit this discussion to sub-surface offset gathers, but similar limitations apply to all of the choices. To construct sub-surface offset gathers used to form angle gathers, the image, in its most general form, expands by one to three dimensions. In the extreme case, the image becomes $I(x, y, z, h_x, h_y, h_z)$, where h_x, h_y, and h_z represent lags in x, y, and z. The imaging condition then becomes

$$I(x, y, z, h_x, h_y, h_z) = \sum_{\text{shot}} \sum_{t} s(x + h_x, y + h_y, z + h_z) g(x - h_x, y - h_y, z - h_z). \quad (4)$$

In this formulation, the cost of propagating the wavefield and reading the receiver wavefield from disk is the same as the zero-time, zero-offset imaging condition. The IO requirements and computational cost of the imaging steps changes dramatically. The image is now 20 (in the case where we only evaluate h_x) to several thousand times larger. As a result, it must now be read and written to disk at every imaging step. If we assume that we are constructing 400 sub-surface offsets at each point, the computational cost increases dramatically. We now benefit from transferring the source and receiver wavefields to shared memory, reducing their cost by possibly up to 80%, but we still end up with a cost of $nx * ny * nz * 800 * (2 * .2 + 2)$, more than a magnitude more expensive than the propagation. However, that is not the real problem. We must read and write each image point at each image step, resulting in an increased IO requirement of 800x. As a result, only approaches that construct several different azimuths of 1-D angle gathers have been generally shown.

Sub-sampling sub-surface offsets offers benefits in both computation and storage. The reduction in offset calculation proportionally reduces the storage on large memory machines, potentially eliminating the need to go to disk. Total computation is reduced by sub-sampling, but given the required random nature of compressive sensing, cache misses will not be proportionally reduced.

COMPRESSIBILITY

There is significant literature on compressing seismic data. Relatively low compression ratios are achievable by compressing a trace. Significantly higher compression ratios are achieved by multi-dimensional approaches. Generally, the best results have used either multi-dimensional wavelets (Mallat, 1999) or its successor curvelets (Candes and Donoho, 1999). Villasenor et al. (1996) showed that compression ratios of 100:1

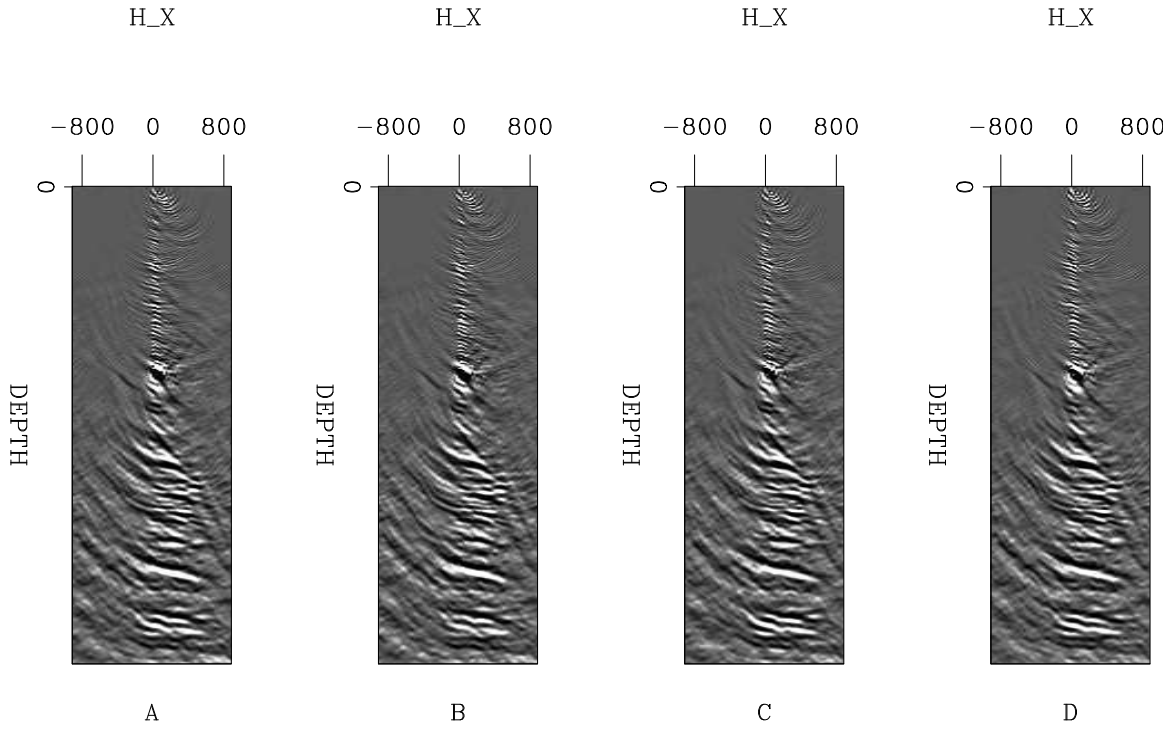

Figure 2: Five sub-surface offset gathers. B and C are one midpoint in X before and after A. E is one midpoint in Y before A. Note the spatial similarity, which lends itself to compression. [**ER**]

were achievable by compressing a 4-D volume (t, h_x, h_y, s). Further, Villasenor et al. (1996) states that the header information was the limiting factor in achieving even higher ratios.

Sub-surface offset gathers potentially represent an even higher, up to six, dimensional problem. To test compressibility, I used a 4-D volume (x, y, z, h_x) of dimensions (32,32,400,64). Figure 2 shows one of these sub-surface offset gathers and its neighbors, note the similarly. Following Villasenor et al. (1996), I chose the 9/7 biorthonormal transform(Antonini et al., 1992) used in JPEG compression. Figure 3 shows the resulting transform space and a histogram of the absolute values. I then used several different thresholds (throwing away 90%, 95%, 98%, and 99% of the data in the wavelet domain). Figure 4 shows the result of transforming this thresholding volume back into the space-domain. The resulting images are near-perfect at 95% and potentially acceptable at 98%. This translates into an acceptable compression ratio of approximately 30:1.

We get a more interesting result if we look at the compressibility of angle gathers. Figure 5 shows the same locations as Figure 2 but in the angle domain. Figure 6 shows the resulting wavelet domain space and a histogram of its absolute values. Note an even greater clustering around 0. Figure 7 shows the result of thresholding the wavelet domain at 95%, 98%, 99%, 99.5%. Note how we see near perfect reconstruction of

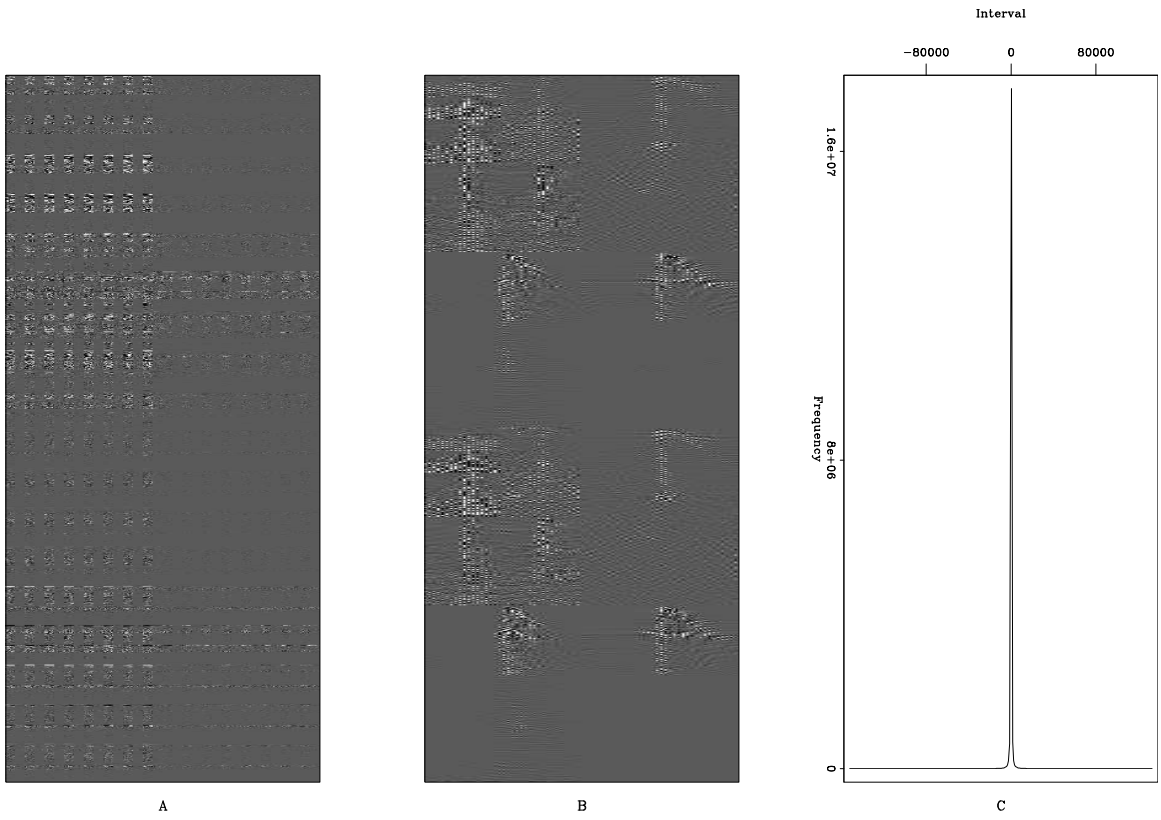

Figure 3: Panel A shows the wavelet domain representation of the 4-D volume used in this experiment. Panel B shows a zoom into portion of the wavelet domain. Panel C shows a histogram of the wavelet domain values. Note how the vast majority of the values are nearly zero. [**ER**]

the angle gather up to 99% and potential up to 99.5%. This is a 100:1 to 200:1 compression ratio (which translates into 20 to 50x reduction in sub-surface offsets).

Framing the compression in terms of angle gathers requires a change in formulation. The **L** in equation 2 now becomes chain of two operators. The first translates from sub-surface offset to angle. The second is multi-dimensional wavelet transform.

DISCUSSION AND CONCLUSIONS

Today, angle gather construction is acomputational bottleneck in terms of computation, cache misses, and IO. Compressive sensing, which reconstructs a sub-sampled signal by a ℓ_1 inversion of the data transformed into a sparse basis function, offers a potential solution. I showed that the angle gather constructions meets the two criterion for compressive sensing: the data is highly compressible using multi-dimensional wavelets and reducing the data size dramatically reduces computational cost. The

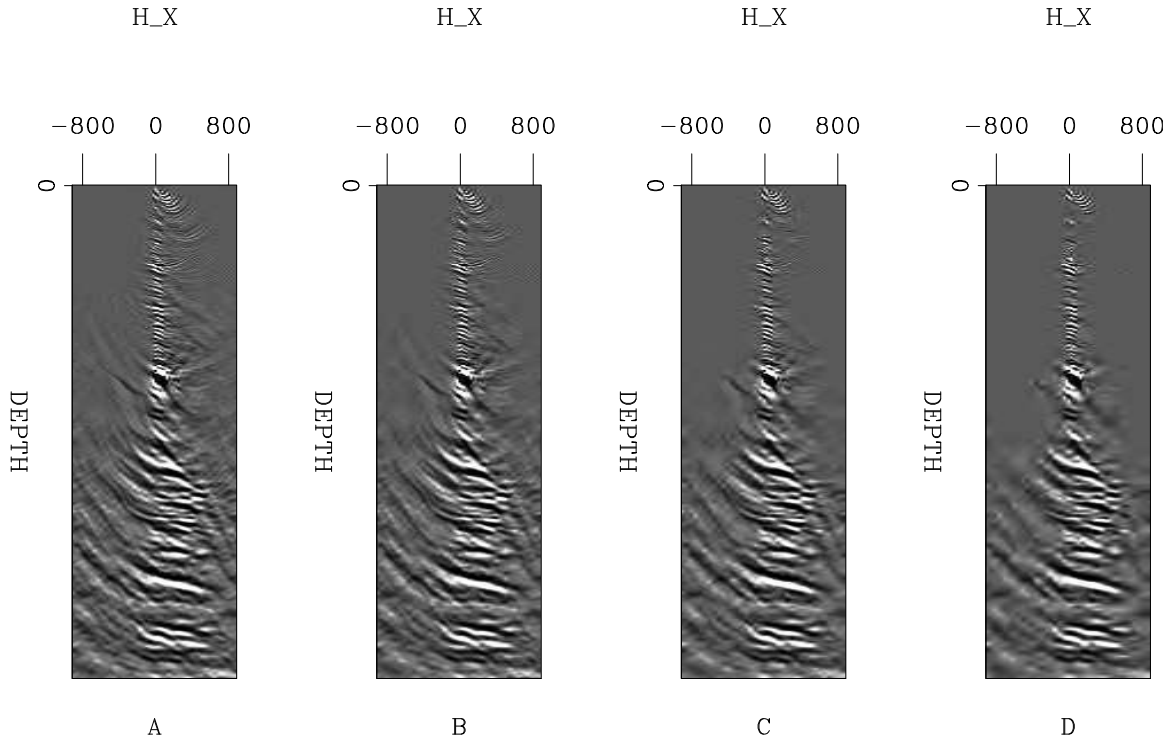

Figure 4: The result of zeroing the smallest values of the wavelet domain representation shown in Figure 3A. All four panels show the same sub-surface offset gather shown in Figure 2. A shows the result of clipping 90% of the values; B, 95%; C, 98%; and D,99%. Note how the reconstructed gather is nearly identical up to a 98% clip. [**ER**]

next step in this work is to apply an ℓ_1 solver to reconstruct a sub-sampled offset gather.

REFERENCES

Antonini, M., M. . Barlaud, P. Mathieu, and I. Daubechies, 1992: IEEE Trans. on Image Proc., 205–220.

Baraniuk, R., J. Romberg, and M. Wakin, 2008, Tutorial on compressive sensing: `http://en.pudn.com/downloads154/sourcecode/zip/detail685097_en.html`.

Candes, E. J. . and D. L. Donoho, 1999, Curvelets - a surprisingly effective nonadaptive representation for objects with edges: Curves and Surfaces.

Clapp, R. G., H. Fu, and O. Lindtjorn, 2010, Sellecting the right hardware for reverse time migration: Leading Edge, **29**, 48–58.

Donoho, D. L., 2006, Compressive sensing: Information Theory, IEEE Transactions on, **52**, 1289–1306.

Mallat, S. G., 1999, A wavelet tour of signal processing: Academic Press.

Mallat, S. G. and Z. Zhang, 1993, Matching pursuits with time-frequency dictionaries:

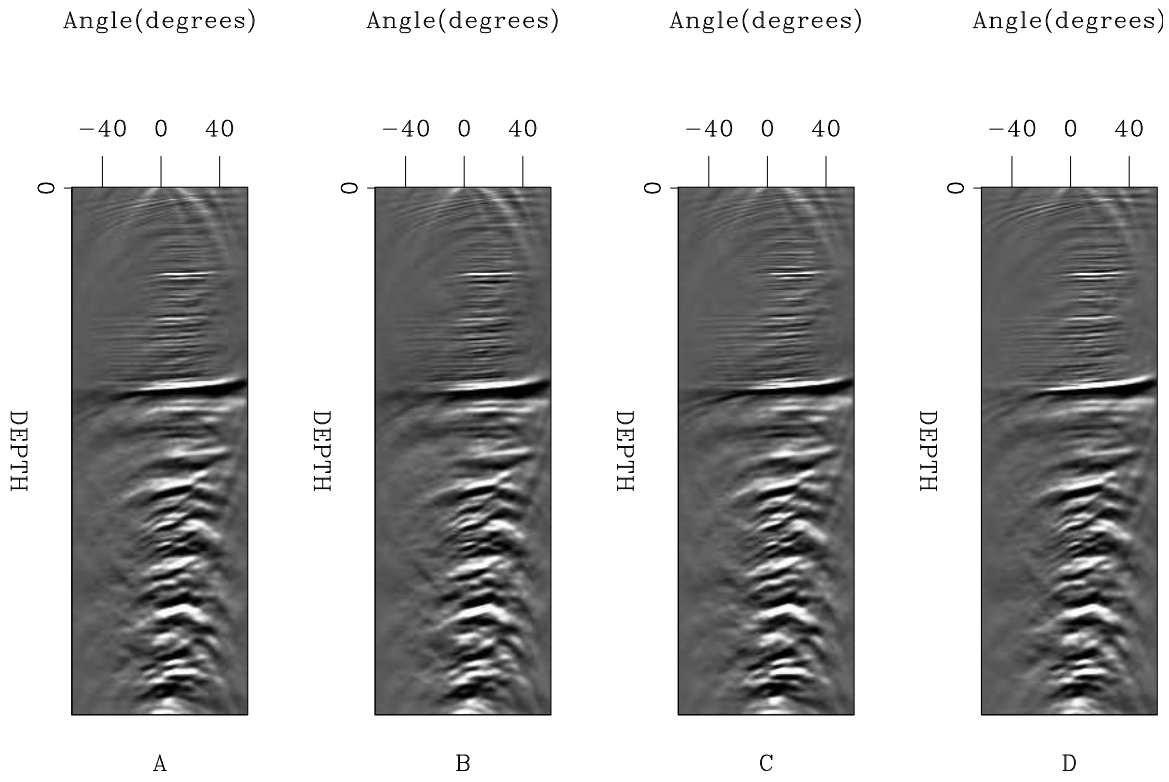

Figure 5: Four angle gathers at the same locations seen in Figure 2. [**ER**]

IEEE Transactions on Signal Processing, **41**, 3397–3415.

Micikevicius, P., 2009, 3d finite difference computation on gpus using cuda: GPGPU, **2**.

Nemeth, T., J. Stefani, W. Liu, R. Dimond, O. Pell, , and R. Ergas, 2008, An implementation of the acoustic wave equation on fpgas: 78th Ann. Internat. Meeting, Expanded Abstracts, 2874–2877, Soc. Expl. Geophys.

Nguyen, A., N. Satish, J. Chhugani, C. Kim, and P. Dubey, 2010, 3.5-d blocking optimization for stencil computations on modern cpus and gpus: Super Computing 2010, Expanded Abstracts, 2874–2877, Super Computing.

Sava, P., 2007, Stereographic imaging condition for wave-equation migration: Geophysics, **72**, A87–A91.

Sava, P. and S. Fomel, 2006, Time-shift imaging condition in seismic migration: Geophysics, **71**, 5209–5217.

Sava, P. C. and S. Fomel, 2003, Angle-domain common-image gathers by wavefield continuation methods: Geophysics, **68**, 1065–1074.

Villasenor, J. P., R. A. Ergas, and P. L. Donoho, 1996, Seismic data compression using high-dimensional wavelet transforms: Snowbird, UT, USA, Expanded Abstracts, 396–405, IEEE Computer Society Press.

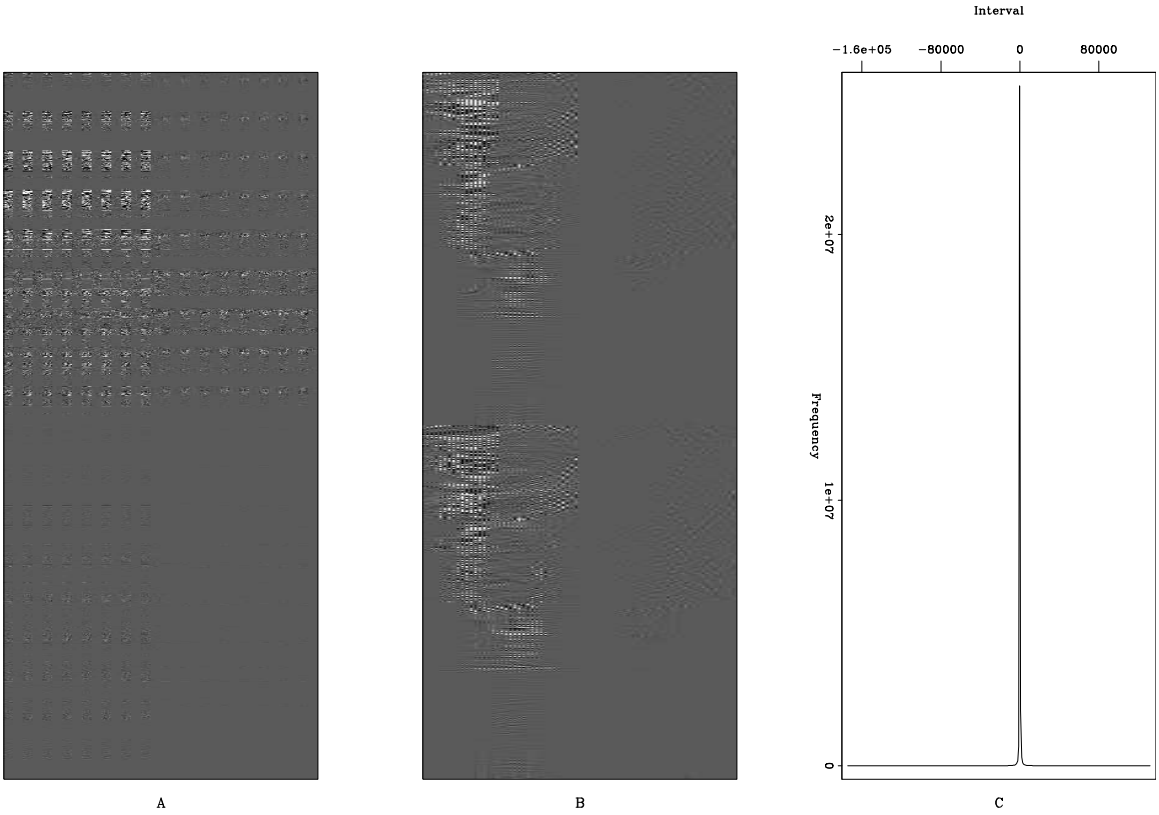

Figure 6: Panel A shows the wavelet domain representation of the 4-D volume used in this experiment transformed into the angle domain. Panel B shows a poriton of the wavelet domain. Panel C shows a histogram of the wavelet domain values. Note how the vast majority of the values are nearly zero. [**ER**]

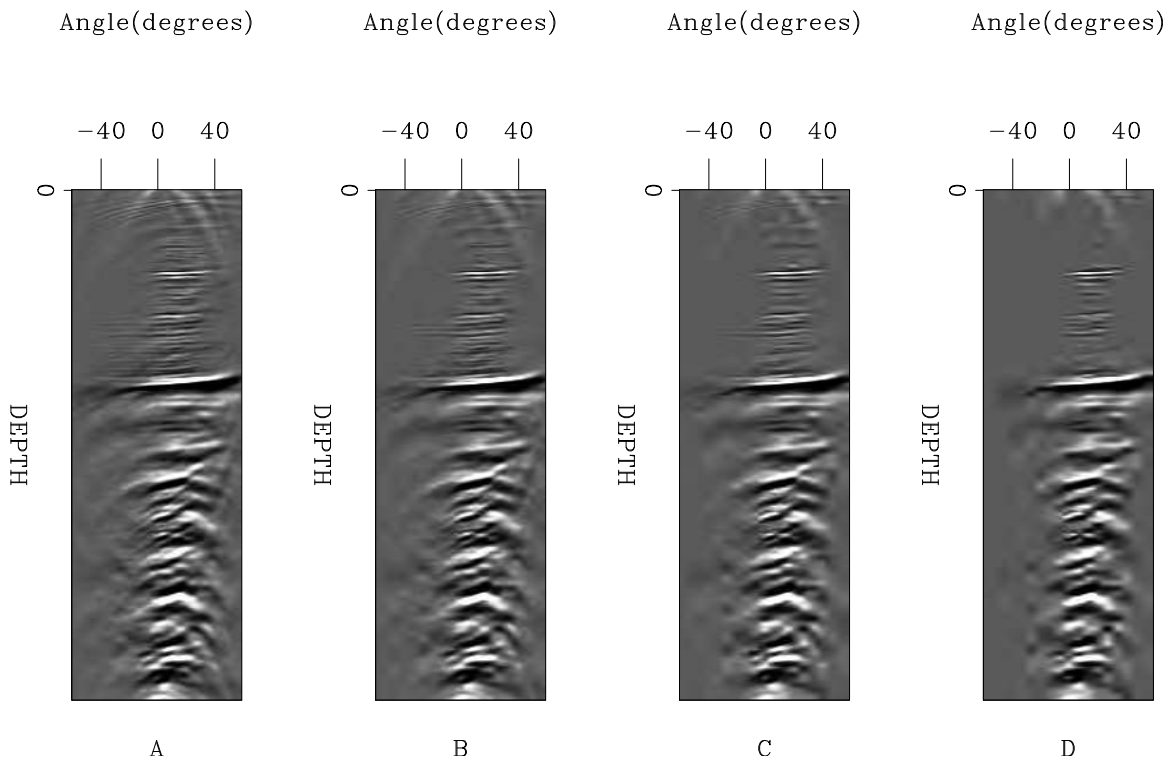

Figure 7: The result of zeroing the smallest values of the wavelet domain representation shown in Figure 3A. All four panels show the same sub-surface offset gather shown in Figure 5. A shows the result of clipping 95% of the values; B, 98%; C, 99%; and D,99.5%. Note how the reconstructed gather is nearly identical up to a 99% clip. [**ER**]