Stanford Exploration Project, Report 94, November 11, 1997, pages 213–222

Short Note

Inverse NMO stack in depth-variable velocity

Yalei Sun[1]

keywords: *inverse NMO stack, conjugate gradient, multiple suppression*

INTRODUCTION

Inverse NMO stack is a procedure which combines conventional NMO and stacking into one step. By solving a set of simultaneous equations using optimization methods, such as conjugate gradient, inverse NMO stack tries to find the most "reasonable" stack trace for a CMP gather. Claerbout (1994) discusses inverse NMO stack in constant velocity to illustrate how back projection can be upgraded towards inversion. In this note, I extend his idea to the case of depth-variable velocity.

When implementing the inverse NMO stack operator, I find some interesting features linking with the concepts of "push" and "pull" (Claerbout, 1995), which are adjoint to each other. In this note, I call this adjoint *implementation adjoint.* There is another kind of adjoint associated with a geophysical operator, which I call *geophysical adjoint.* For instance, the process of generating a CMP gather from a stack trace and the process of creating a stack trace from a CMP gather are adjoint to each other. Ideally, the geophysical adjoint should be independent from the implementation adjoint. In practice, it is not true. The geophysical adjoint is always coupled with the implementation adjoint. For example, if the geophysical forward operation is "push", the adjoint operation must be "pull", and vice versa.

In this note, I discuss the relation between the geophysical adjoint and the implementation adjoint. I argue that this relation is inevitable in our computer implementation. In other words, this relation will only disappear when the geophysical adjoint operation can be implemented in continuous space, instead of a discrete space.

Finally, I investigate the possibility of using this method in suppressing nonhyperbolic moveout and random noise. The result is encouraging.

[1]**email:** yalei@sep.stanford.edu

INVERSE NMO STACK OPERATOR

The inverse NMO stack operator is based on the hyperbolic moveout equation

$$t = \sqrt{\tau^2 + \frac{x^2}{v^2(\tau)}}, \tag{1}$$

Here, t is the nonzero-offset traveltime, τ zero-offset traveltime. x is the shot-geophone offset and $v(\tau)$ is the RMS depth-variable velocity.

The model $\mathbf{m}$ is the stack trace, while the CMP gather is the data $\mathbf{d}$. Correspondingly, $\mathbf{m}$ is located in the *model space* and $\mathbf{d}$ in the *data space*. The hyperbolic moveout operator $\mathbf{L}$ links the model $\mathbf{m}$ with the data $\mathbf{d}$ by

$$\mathbf{d} = \mathbf{L}\mathbf{m}. \tag{2}$$

Equation (2) is a set of simultaneous equations, which is typically over-determined. It is very hard to find the exact solution $\mathbf{m}$ for known data $\mathbf{d}$. Therefore, we have to reformulate this problem as an optimization problem and try to find the optimal solution in some sense, such as least square. The conjugate-gradient algorithm is used to search for the optimal solution by minimizing the residual

$$\mathbf{0} \approx \mathbf{r} = \mathbf{L}\mathbf{m} - \mathbf{d}. \tag{3}$$

IMPLEMENTATION ADJOINT AND GEOPHYSICAL ADJOINT

Like many other geophysical operators, the inverse NMO stack operator can be coded in either "push" or "pull" forms, namely, we can loop over the output space and "pull" a corresponding value from the input space, or loop over the input space and "push" the value to an appropriate location in the output space. The adjoint of a pull operator is a push operator and vice versa, which I call implementation adjoint.

On the other hand, there is another scheme of implementing the geophysical forward and adjoint operation, which seems to be more reasonable. That is, using "push" in both the forward and adjoint operations. The reason I say it seems to be more reasonable is because it is a kind of energy-conserved scheme. For example, starting with a forward "push" operation, we push all the input energy into the output space, then we push the energy back into the input space by applying a adjoint "push" operation. Seemingly, there is no energy loss during the forward & adjoint operation, and we should be able to get the exact original input. Actually, since all the implementations are discrete schemes, some kind of interpolation is inevitable. The discrete adjoint "push" operation is not exact adjoint as it should be in the continuous space. Therefore, the output of adjoint operation will not be the same as the input.

I implement the inverse NMO stack operator using two interpolation methods, nearest-neighborhood interpolation and linear interpolation. If I implement the forward operation by "push" and the adjoint operation by "pull", or vice versa, the results are correct and independent of the interpolation scheme and the velocity type, as shown in Figure 1. In other words, if the geophysical forward and adjoint operation loop over the same space (model or data), the result will be acceptable.

Then I try to implement both the forward and adjoint operations using "push" or "pull". As shown in Figure 2, the results are unsatisfactory, no matter what interpolation and velocity I use. It means that, if the forward and adjoint operation loop over different spaces, we cannot recover the original input.

In seismic processing, we start with a CMP gather in the data space and want to find the best-fit stack trace in the model space. Therefore, we only need to think about the adjoint operation. In this case, I claim that the "pull" implementation is the only choice and the "push" implementation will give an incorrect result. Because in equation (1), we assume the RMS velocity is the zero-offset velocity. The "push" scheme actually uses the RMS velocity $v(t)$ instead of $v(\tau)$.

Figure 3 shows a CMP gather and its estimated depth-variable RMS velocity. Starting with this CMP gather, I try both the "pull" and "push" scheme to invert for the stack trace. Figure 4 shows the results from the two schemes and also the difference between the two schemes. The amplitude of the "pull" scheme result is higher than that of the "push" scheme result. This means the kinematic relation is better observed in the "pull" scheme. It is also very clear that the difference becomes small in the deep zone. This is due to the fact that the hyperbolic event becomes flatter with increasing time, which makes $v(t)$ similar to $v(\tau)$.

In order to verify the above conclusion, I compare the "pull" scheme result with the conventional NMO+stack result (Figure 5). The difference between these two approaches is negligible. This proves that the inverse NMO stack is equivalent to NMO + stacking.

NONHYPERBOLIC EVENTS SUPPRESSION

In this section, I make use of the characteristic of the inverse NMO stack operator to remove nonhyperbolic events and random noise. Here, the hyperbolic events only refer to the P-wave. Therefore, the S-wave and the multiples will be regarded as nonhyperbolic events.

Figure 6 is a synthetic CMP gather consisting of P-wave, S-wave and random noise. After applying the adjoint and forward operation, we get a new CMP gather. As shown in Figure 6, most of the random noise and the two S wave events have been removed from the CMP gather with little loss of P wave information.

Figure 7 is a more complicated CMP gather. In addition to P-wave, S-wave and random noise, this CMP gather contains multiples. As shown in Figure 7, after the

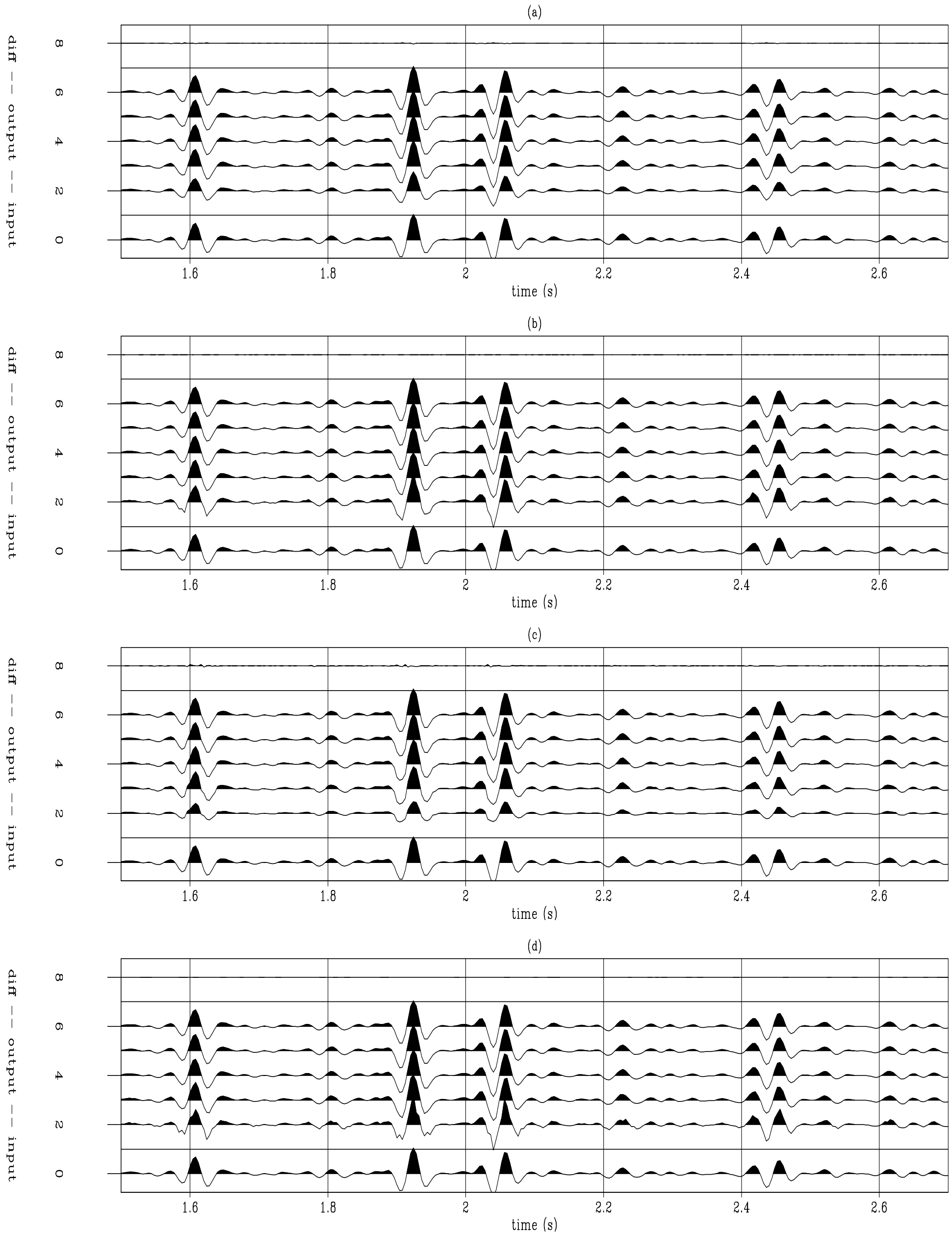

Figure 1: Start with a stack trace in the model space (trace 0) and generate a CMP gather using the forward operator. Then calculate the best stack trace from the CMP gather (trace 2 to 6, from one to five iterations). Zero trace 1 and 7 represent the separations between the input, output, and difference. The difference (trace 8) is produced by subtracting the final output (trace 6) from the input (trace 0). Velocity is constant. **(a)** forward is "push", adjoint "pull", and linear interpolation. **(b)** forward is "pull", adjoint "push", and linear interpolation. **(c)** forward is "push", adjoint "pull", and nearest-neighborhood interpolation. **(d)** forward is "pull", adjoint "push", nearest-neighborhood interpolation. All the four results are accurate, which means the inverse NMO stack is numerically invertible in these specific implementation schemes. $\boxed{\text{yalei2-dependent}}$ [ER]

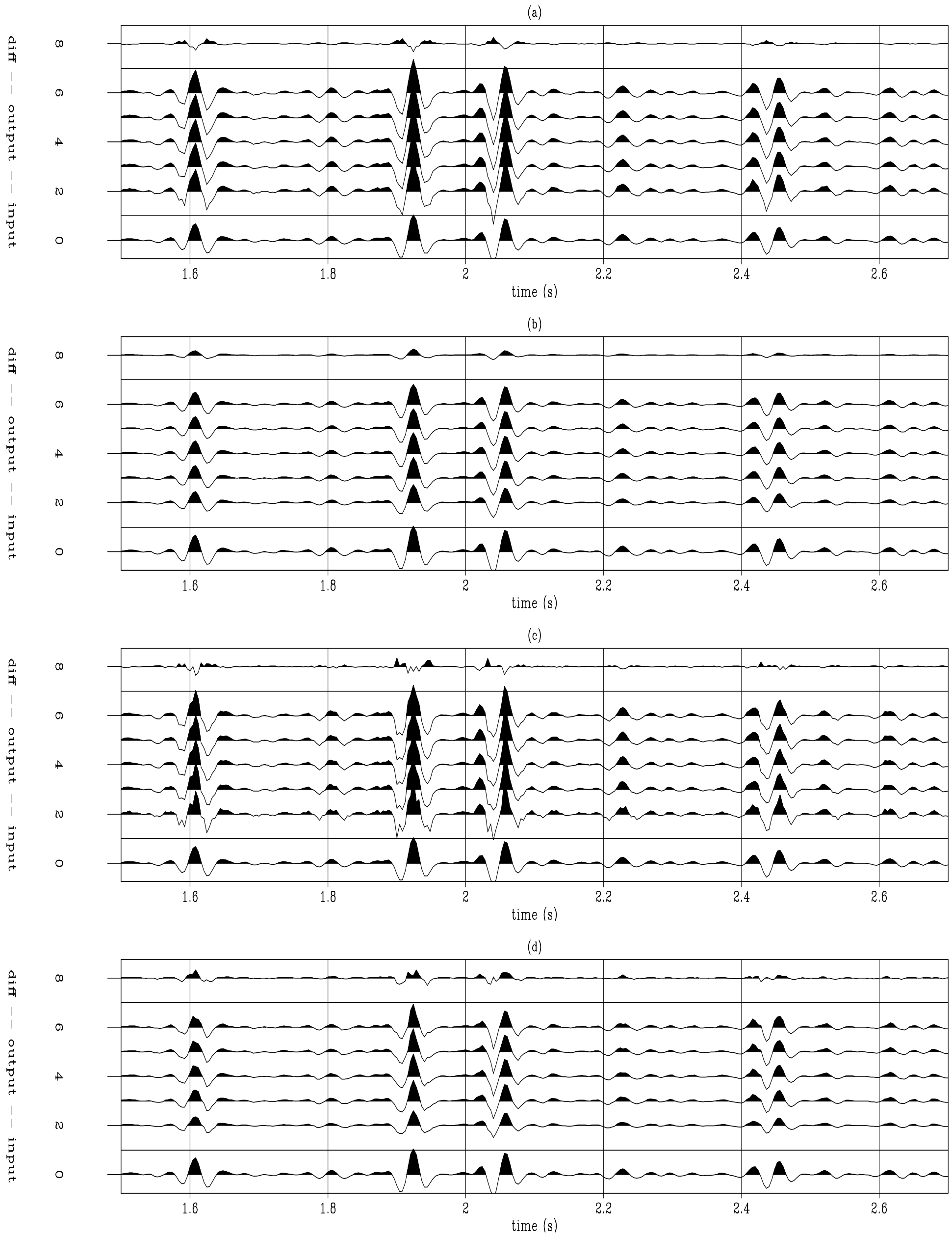

Figure 2: Same as Figure 1. **(a)** forward is "push", adjoint "push", and linear interpolation. **(b)** forward is "pull", adjoint "pull", and linear interpolation. **(c)** forward is "push", adjoint "push", and nearest-neighborhood interpolation. **(d)** forward is "pull", adjoint "pull", nearest-neighborhood interpolation. It is interesting to notice that the result of double "push" is always smaller than the original input and the result of double "pull" is always larger than the original input. The difference between the final output and input decreases with time. Because in the deep zone, the effect of interpolation is less than that in the shallow zone. yalei2-independent [ER]

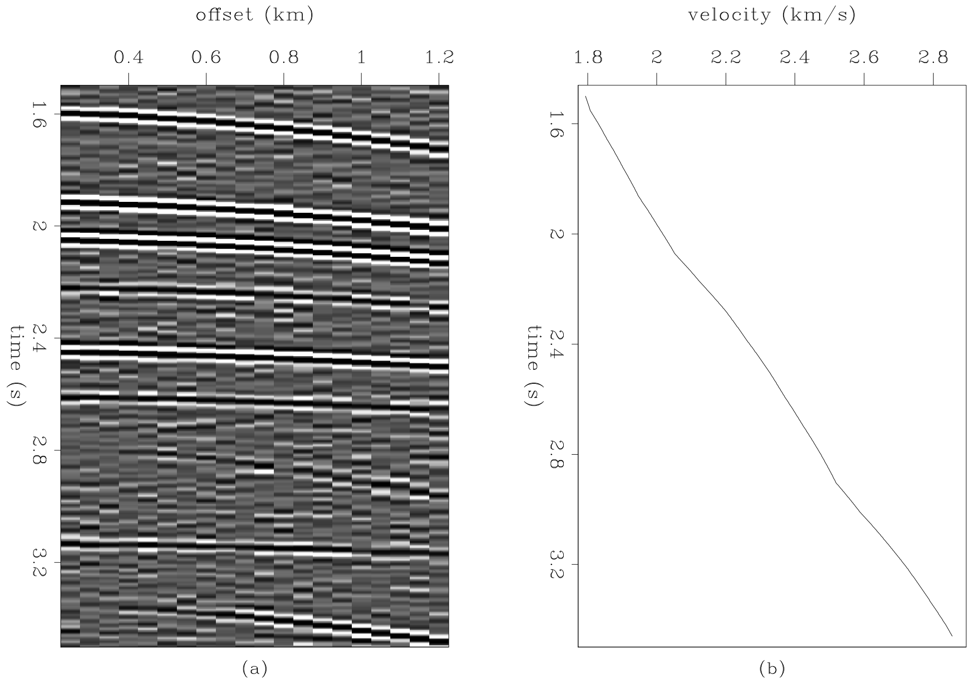

Figure 3: **(a)** A synthetic CMP gather and **(b)** its estimated RMS velocity. yalei2-hp-input [ER]

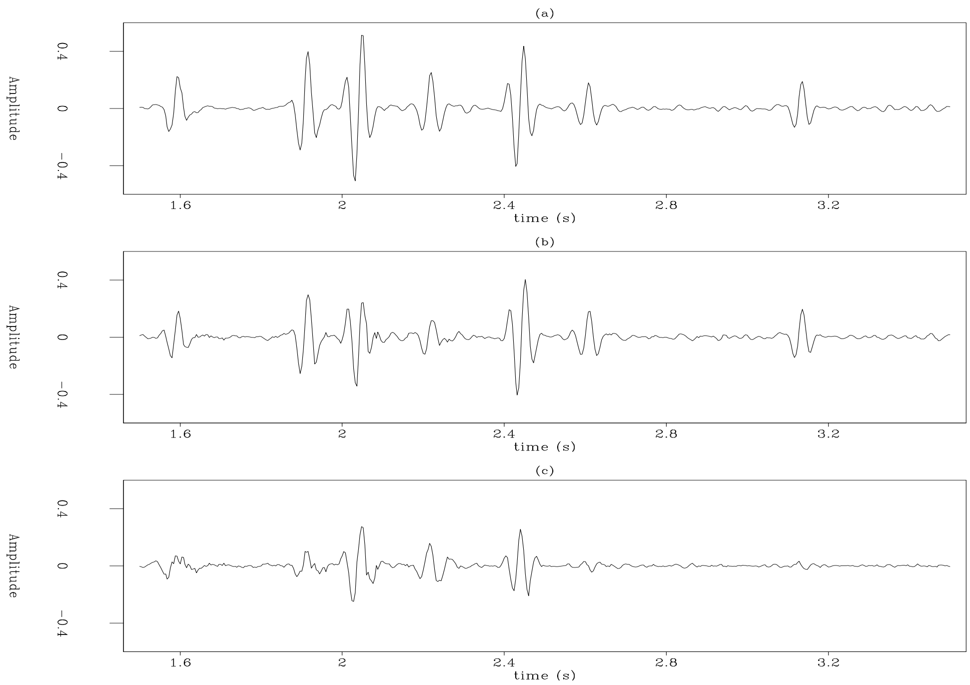

Figure 4: Start with a CMP gather and end up with a stack trace. **(a)** the "pull" scheme result, **(b)** the "push" scheme result, and **(c)** the significant difference. yalei2-hp-space [ER]

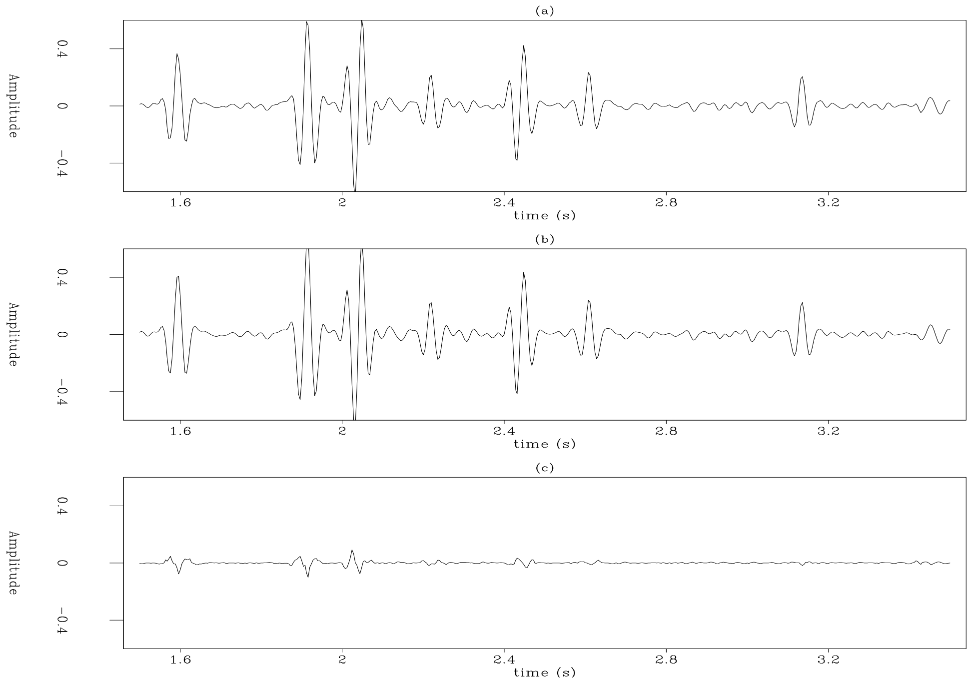

Figure 5: Same as Figure 4. **(a)** the "pull" scheme result, **(b)** the NMO+stack result, and **(c)** the minor difference. $\boxed{\text{yalei2-hp-stack}}$ [ER]

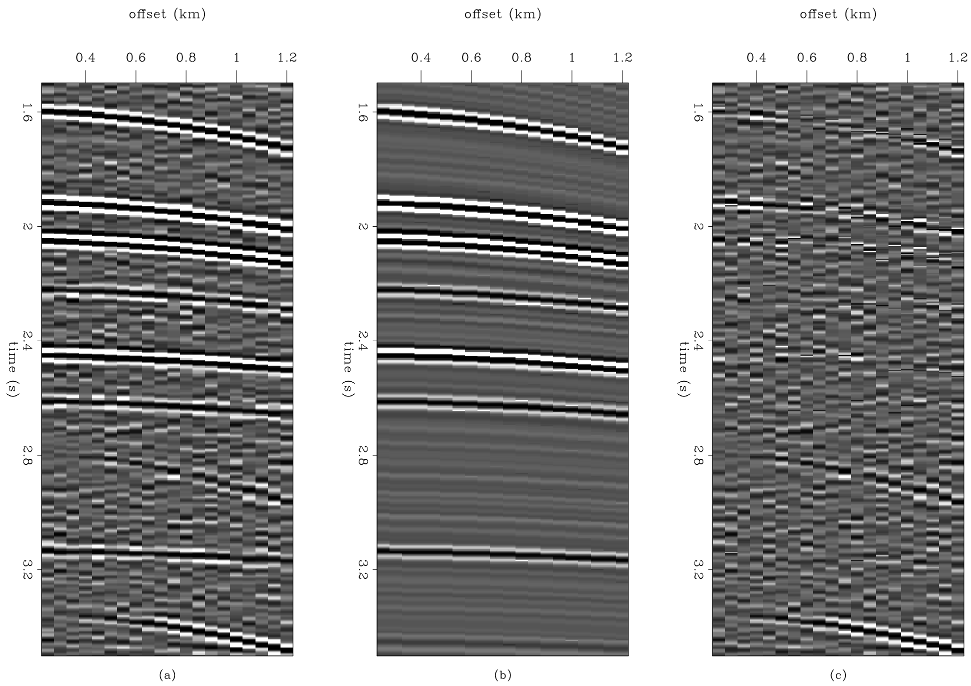

Figure 6: **(a)** A synthetic CMP gather having only primary and random noise. **(b)** The same CMP gather after adjoint & forward operation. **(c)** Difference. $\boxed{\text{yalei2-hp-cmp}}$ [ER]

inverse NMO stack operation, not only are the random noise and the two S-wave removed from the CMP gather, but also the multiples are well separated from the P waves.

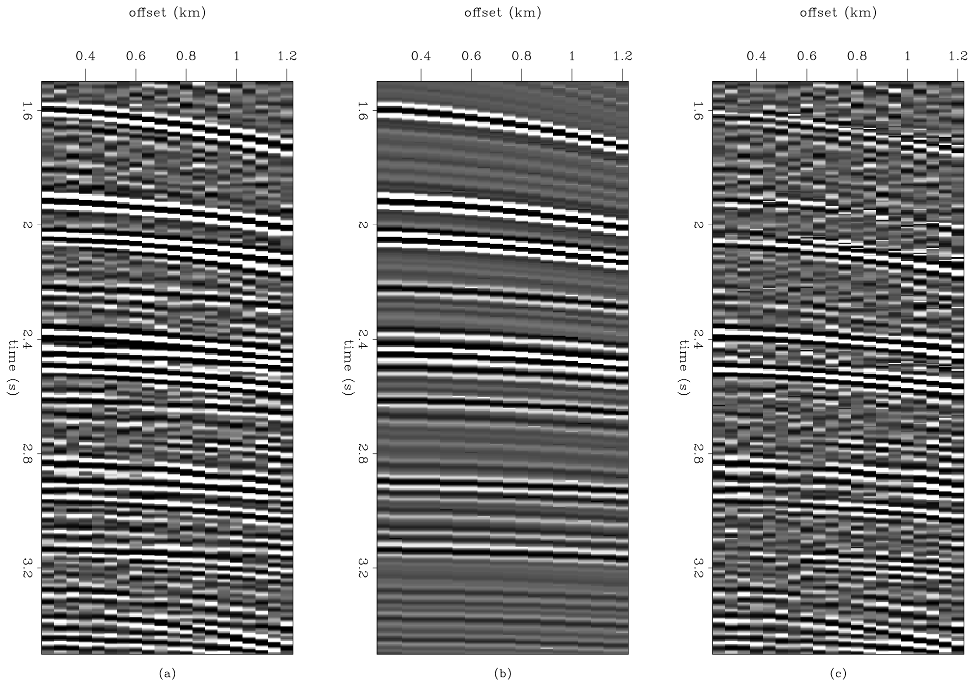

Figure 7: **(a)** A synthetic CMP gather having primaries and multiples. **(b)** The same CMP gather after adjoint/forward operation. **(c)** Difference. yalei2-hm-cmp [ER]

Figure 8 is a real CMP gather. The white and black events represent P wave energy. After applying the operator to this gather, the P wave energy has been preserved and the multiples and random noise are removed.

CONCLUSION AND DISCUSSION

Inverse NMO stack operator combines the conventional NMO and stacking operation into a single step. The performance of inverse NMO stack is similar to the cascaded NMO+stack. Due to the characteristic of the discrete implementation of geophyiscal operators, the interpolation procedure is unavoidable. Therefore, the geophyiscal adjoint and implementation adjoint are coupled, although they are essentially different in meaning.

The inverse NMO stack operator is based on hyperbolic moveout relation. Therefore, it can be used to remove nonhyperbolic events and random noise. The algorithm is very efficient. But since the operator does not take amplitude into consideration, it cannot handle the AVO effects correctly. The application of this approach is largely confined by this limitation.

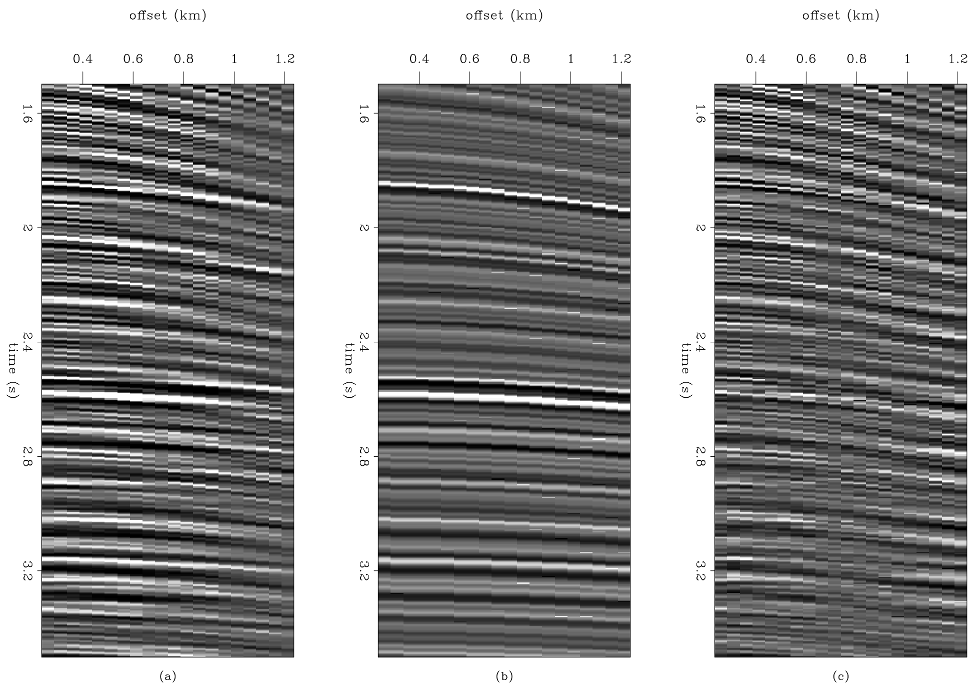

Figure 8: **(a)** A real CMP gather having primaries and multiples. **(b)** The same CMP gather after adjoint/forward operation. **(c)** Difference. yalei2-mo-cmp [ER]

ACKNOWLEDGEMENTS

I thank Bob Clapp and Sergey Fomel for their thorough proofreading.

REFERENCES

Claerbout, J. F., 1994, Applications of Three-Dimensional Filtering: Stanford Exploration Project.

Claerbout, J., 1995, Ellipsoids versus hyperboloids: SEP–**89**, 201–206.