Avoiding artifacts in phase shift migration

Stewart A. Levin

Introduction

In SEP-30 Shuki Ronen and Bill Harlan discussed wrap-around and interpolation artifacts in Stolt migration and how, via padding and sinc function interpolation, many could be avoided. Similar difficulties arise in phase shift migration and I will show here how to avoid them. As a side benefit, I obtain a speedup of the phase shift algorithm analogous to one used in finite difference migration.

The thrill of the chase

To generate a couple of figures comparing the artifacts of Stolt and phase shift migration, I ran a simple six point synthetic through both of them. The results, displayed at a very high gain to accentuate artifacts, are in figures 1 and 2, respectively. When played back at an ordinary gain (e.g. figure 3), we see that the wraparound artifacts are a good deal weaker than the primary events they overlay. I was surprised, however, that the phase shift wrap-around was so noticeable. Following conventional wisdom, I zero-padded the time axis (to twice the length) and remigrated. Sure enough (figure 4) some of the artifacts went away and the others were significantly attenuated.

Thinking about it a little, I recalled that multiplying phase shifts in the $f-k$ domain is equivalent to circular convolution in $x-t$ space. When performing such convolutions, we need to zero pad the operator as well as the operand. In the above examples I wasn't doing this. So I went back and zero padded the time axis to twice its length and, in the phase shift migration program, inverse transformed the phase shifts, windowed to an equivalent amount of zero padding, and forward transformed before applying them. (An equivalent operation would be convolving with the sinc function transform of that window.) The result was figure 5. The artifacts were still there and virtually unchanged!

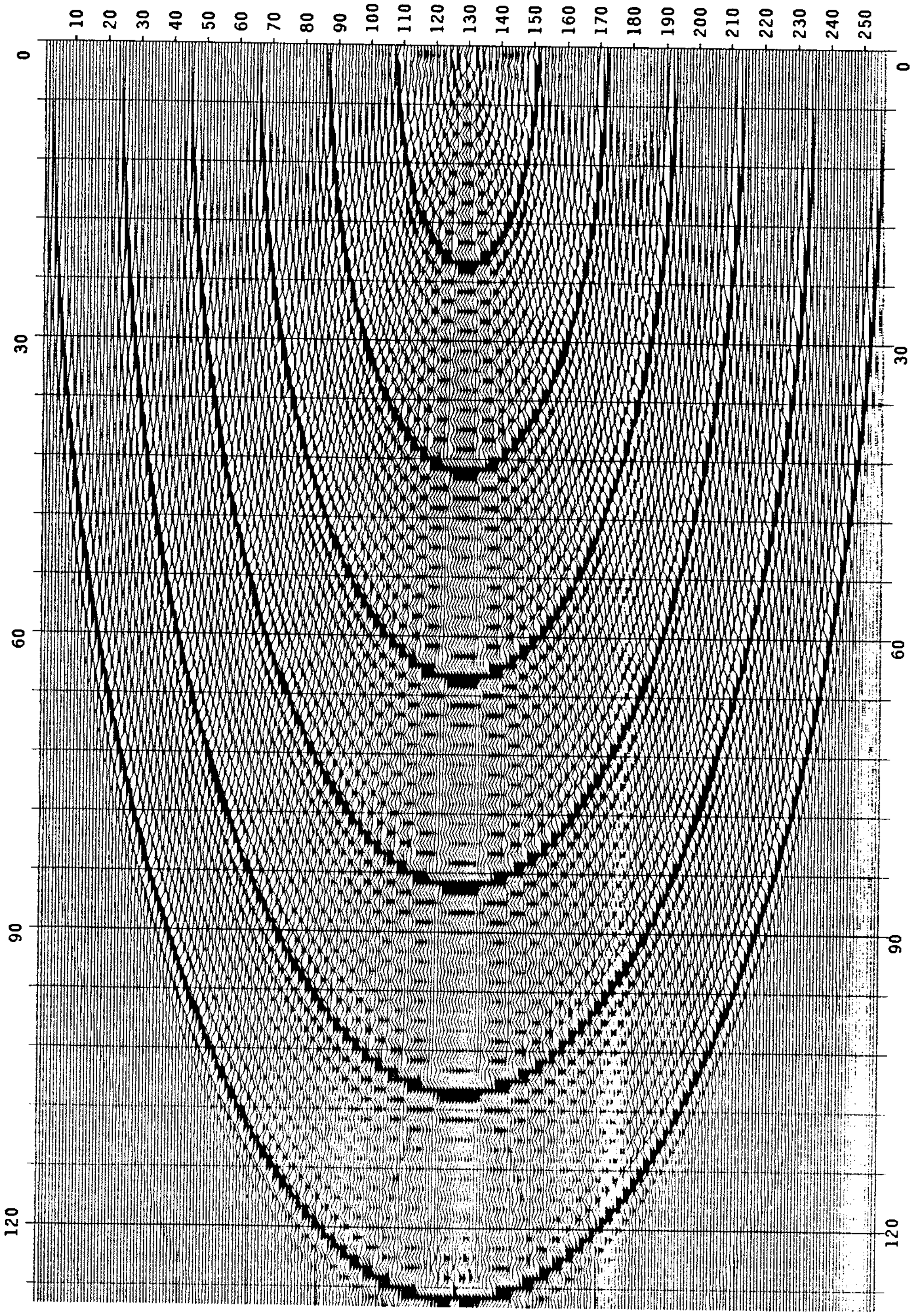

FIG. 1. Stolt migration of a six point synthetic. Playback gain is 10 times normal.

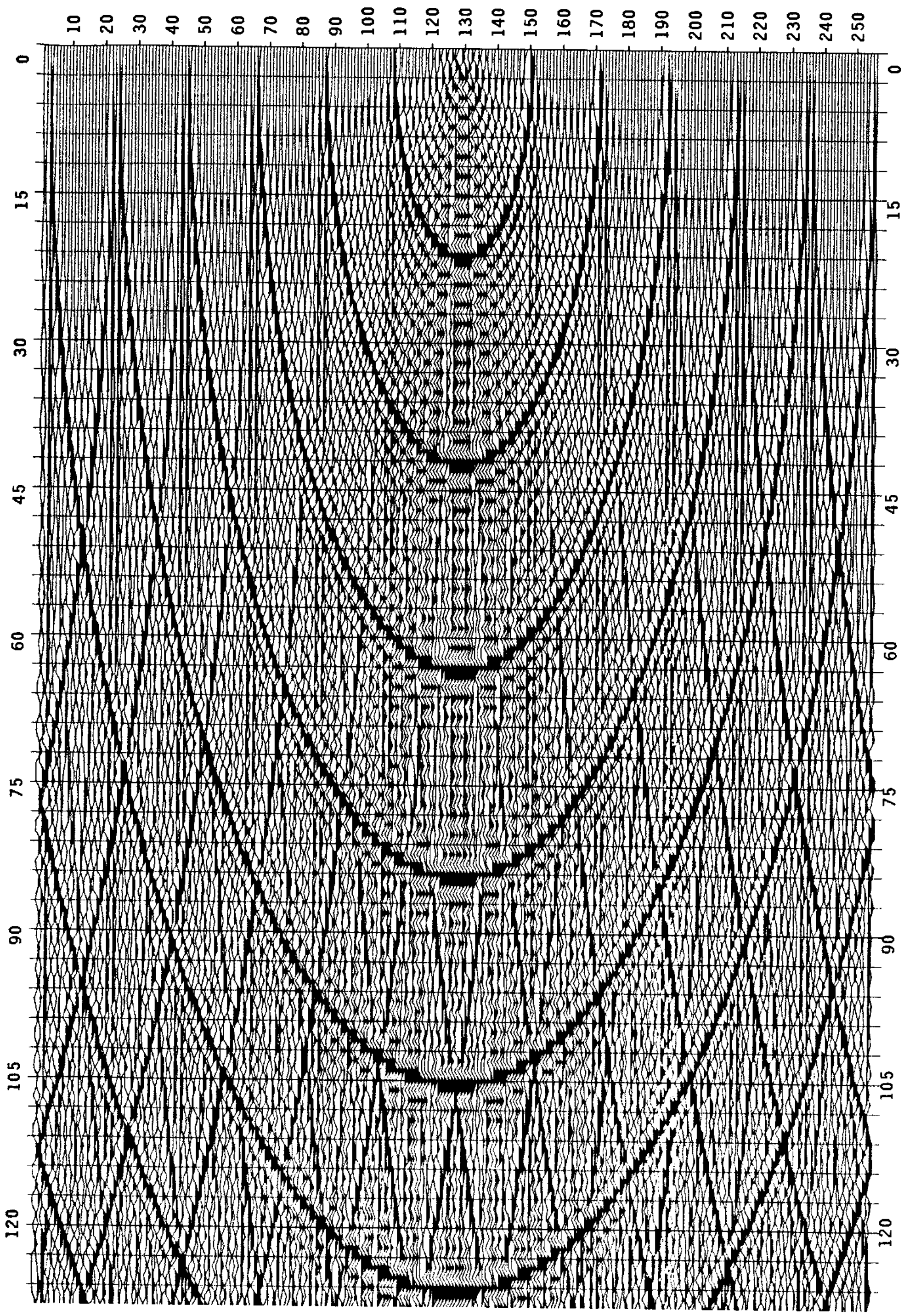

FIG. 2. Phase shift migration of the same synthetic. Playback gain is again 10×.

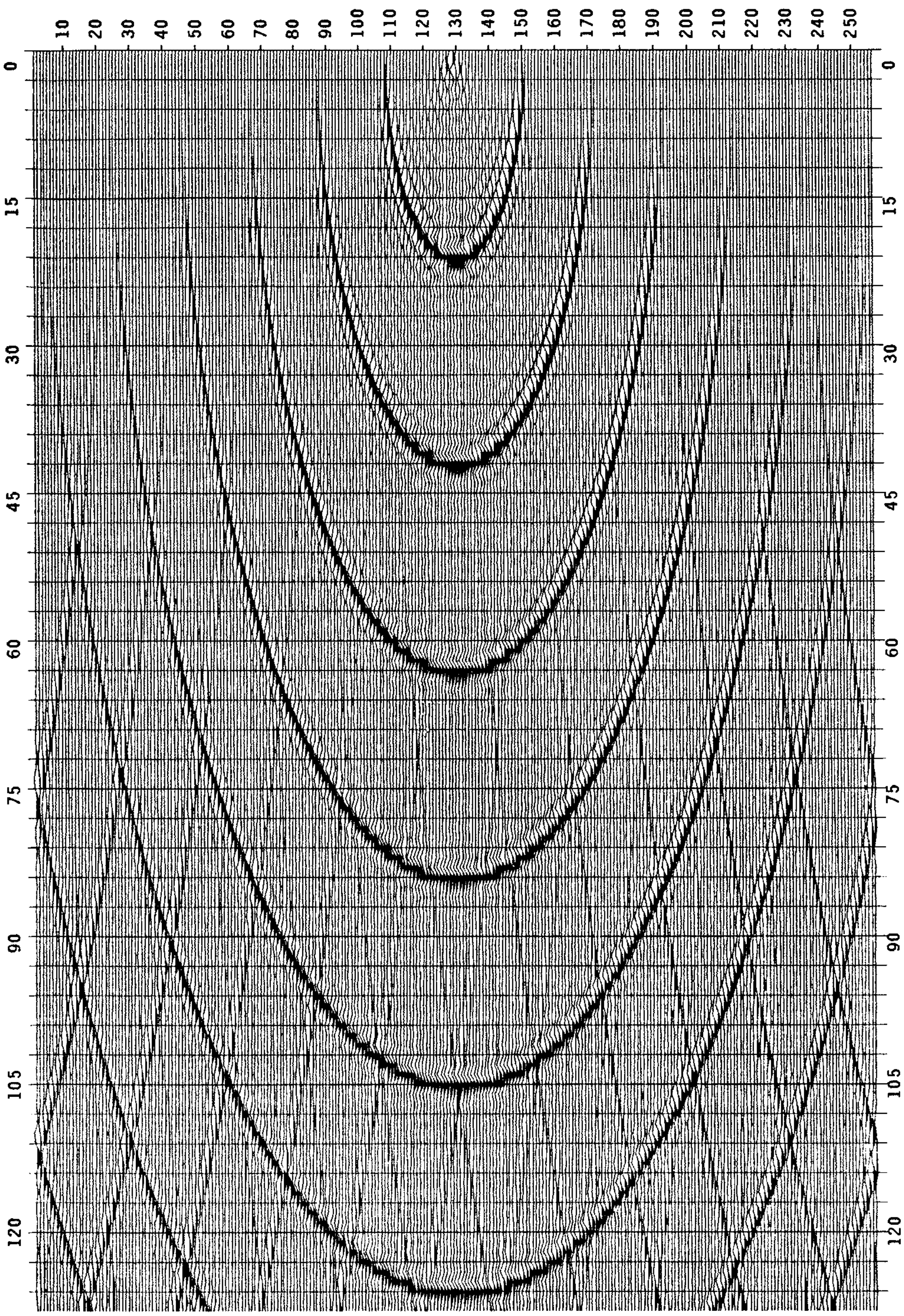

FIG. 3. Previous phase shift migration of the same synthetic replotted at normal scale. The wrap-around is seen to be weaker than the primary events.

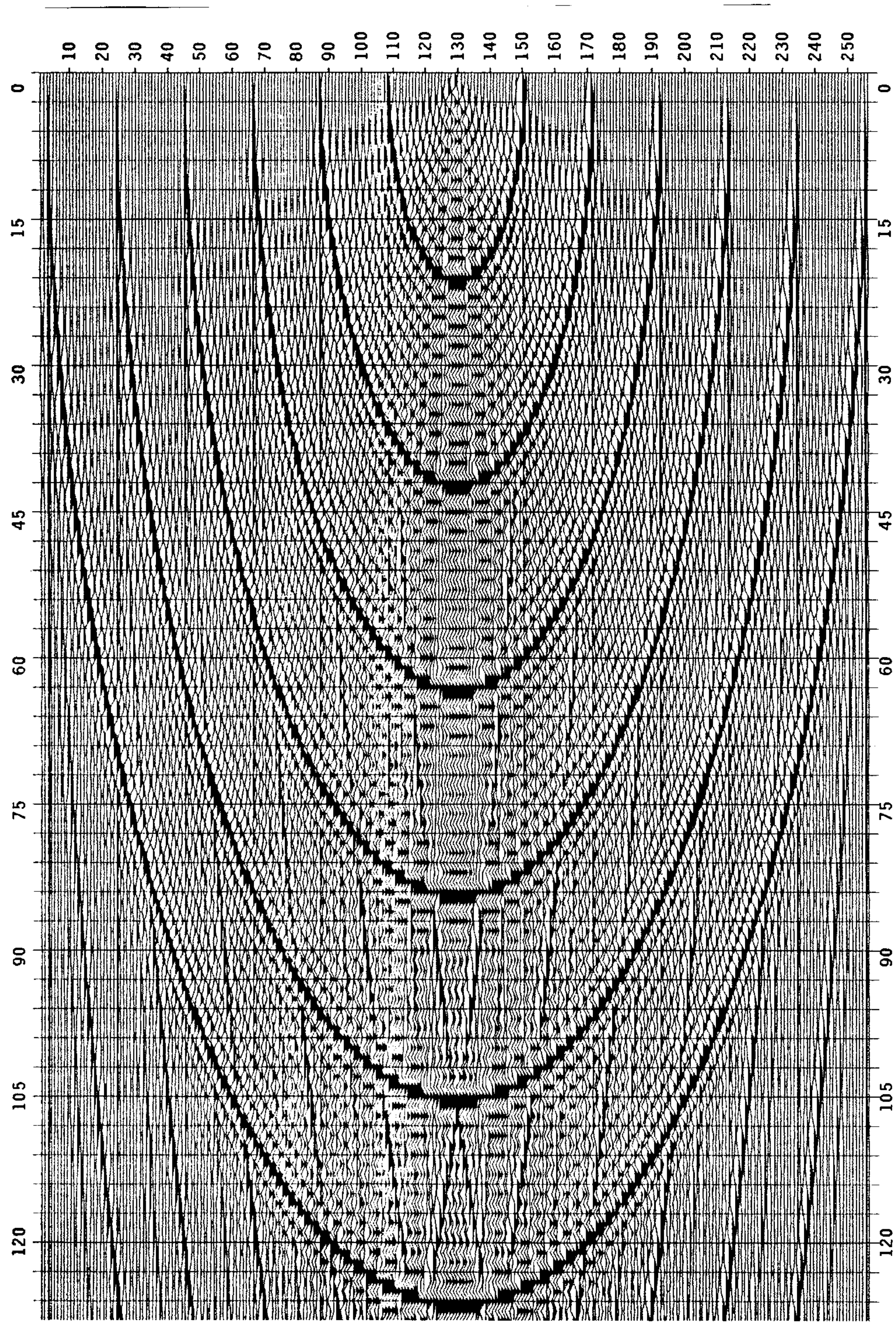

FIG. 4. Phase shift migration of the synthetic, zero padded to twice its original length during computations. Comparing to figure 2, we see a reduction in wraparound artifacts.

This was puzzling. After some more thought, however, I realized what was still wrong - the windowing of the operator needed to be done before discretization of the phase shifts rather than after, otherwise the migration operator would already be wrapped-around. This meant that the continuous version of the phase shifts needed to be convolved, analytically or numerically, with the windowing sinc function.

Since there wasn't, in general, any analytic expression that handled arbitrary $v(z)$ and the requisite numerical integration appeared to add prohibitive cost to phase shift migration, I looked for a more practical alternative method for reducing these artifacts to the negligible levels that the Stolt migration exhibited.

Taking a clue from Kirchoff migration, I decided to use dip control to limit the vertical extent of the phase shift summation operator. The required expressions for a layered $v(z)$ medium are given in JFC's notes on Stolt stretch and may be derived from Snell's law or stationary phase. They are

$$t = \int_0^{\tau_0} \frac{d\tau}{\sqrt{1 - p^2 v^2(\tau)}} \tag{1a}$$

$$x = \int_0^{\tau_0} \frac{p v^2(\tau) d\tau}{\sqrt{1 - p^2 v^2(\tau)}} \tag{1b}$$

where τ_0 is the traveltime of a vertical ray and p is the time slope given by k/ω. If the data only extends to time t_{max}, then we discard from consideration those values of p that give rise to arrival times greater than t_{max}. To minimize truncation artifacts it is helpful to taper p smoothly beyond this limit up to, say, that p which arrives , via formula (1a), *1.25* t_{max}. Having thus computed $p_{max}(\tau_0)$ we may then insert that value into (1b) to determine how much lateral padding is needed to insure that events do not migrate through the padding into the other side of the data.

For constant velocity, v, equations (1) may be integrated analytically to give

$$p_{max} = \frac{1}{v}\left(1 - \frac{\tau_0^2}{t_{max}^2}\right)^{1/2} \tag{2a}$$

$$x_{max} = p v^2 t_{max} \quad . \tag{2b}$$

Applying these formulas to our example yielded the pleasing result in figure 6. Virtually all wrap-around has been eliminated.

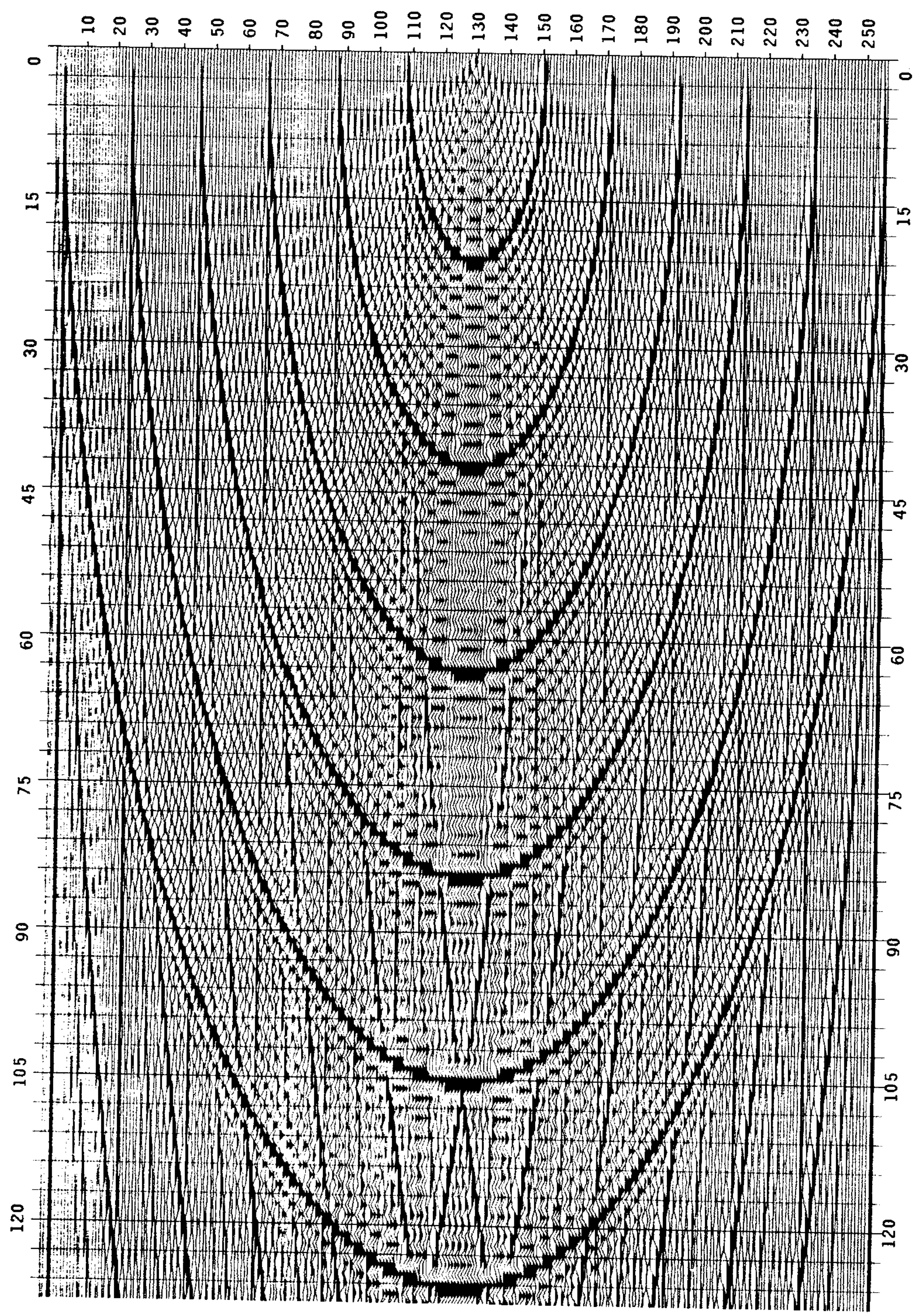

FIG. 5. Phase shift migration of the synthetic, with both data and operators zero extended to twice the section length. Comparing to figure 4, we see no reduction in wraparound artifacts.

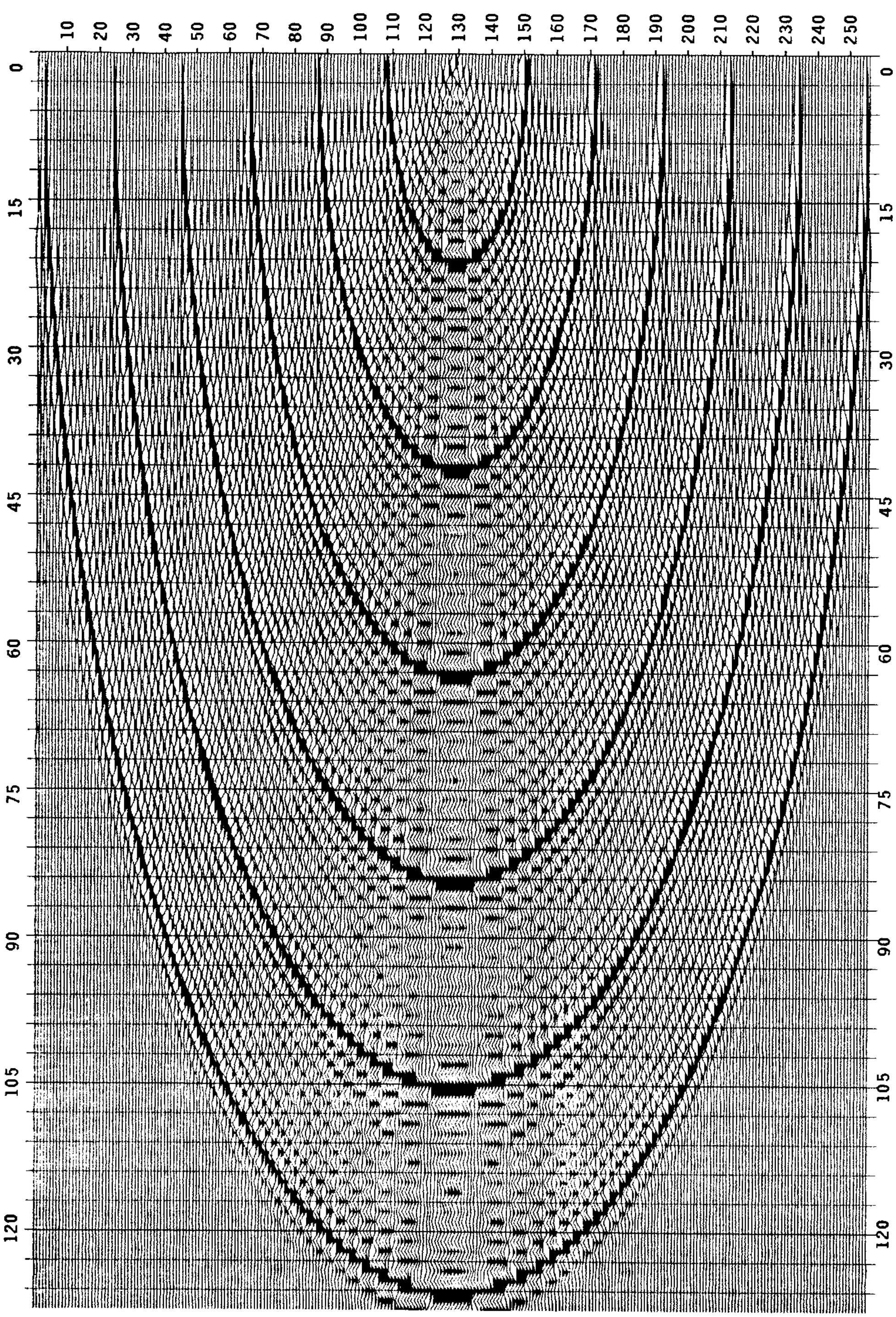

FIG. 6. Phase shift migration of the synthetic, with dip controlled operators. Comparing to our previous figures, we see virtually complete elimination of wraparound artifacts.

A side benefit

One notable feature of using dip control is the reduction of computational cost. The deeper we are, the smaller p_{max} is and the fewer $\omega - k$ pairs we need to process. This savings is analogous to that of finite difference migration where, for output traveltime τ, we downward continue only those data with t greater than τ. In the above example the savings was substantial - CPU time was cut approximately in half.

Conclusion

Just as with Stolt migration, the discrete Fourier transform introduces wraparound artifacts in phase shift migration. The removal of these artifacts, while possible with careful sinc function convolution, is better done for phase shift migration with the use of dip control.

REFERENCES

Harlan, W., Avoiding interpolation artifacts in Stolt migration: SEP-30, p. 103-110.
Lynn, W., Implementing f-k migration and diffraction: SEP-11, p. 18-28.
Ronen, S., Stolt migration; interpolation artifacts: SEP-30, p. 95-101.