Stanford Exploration Project, Report 65, January 13, 1998, pages 99–114

Stanford Exploration Project, Report 65, January 13, 1998, pages 99–114

Fast parabolic transforms

Gilles Darche[1]

keywords: *multiples, NMO, velocity, modeling, interpolation, slant stack*

ABSTRACT

I improve here an already existing method of multiples elimination by parabolic transforms. It is based on the approximate parabolic shape of the multiples after NMO correction, and uses a parabolic transform similar in concept to the hyperbolic and slant-stack transforms. This parabolic transform is easy to express in the frequency domain. More important, I will show that its least-squares inverse can also be computed easily, because it requires only the inversion of a Toeplitz matrix. This property makes the transformations especially fast to compute, and is still valid for special cases, like irregular space sampling, or offset-dependent weighting. I will recall the interest of this method for multiple elimination, and extend it to interpolation processes to illustrate the practical advantages of the Toeplitz structure.

INTRODUCTION

The method of multiples elimination by parabolic transforms was first proposed by Hampson (1986 a). It uses the fact that, after NMO correction of a CMP gather, the shape of the multiples is approximately parabolic, while the primaries are more or less horizontal. To suppress the multiples, the process will thus separate the parabolic events from the horizontal events, characterizing them by their curvature parameters. Yilmaz (1989) proposed a similar method, to be applied on the original data (not NMO-corrected), after stretching the time axis.

We need to transform the NMO-corrected data to the $t_0 - p$ domain, t_0 being the 0-offset time, and p the curvature of the parabolas. Actually, the basic transform is a modeling process, transforming a spike in the $t_0 - p$ domain into a parabola in the time-offset domain. In this sense, it is similar to hyperbolic or slant-stack transforms, where a spike is respectively transformed into an hyperbola or a straight line (Thorson, 1984; Kostov, 1989). The modeling can be easily expressed in the frequency domain, as Kostov (1989) already pointed out.

However, the inverse transformation (time-offset to $t_0 - p$) needs the least-squares

[1]**email:** not available

inverse of this modeling operator, for each frequency. Hampson (1986 b) did not indeed mention how he performed this inversion; he just suggested that the size of the matrices involved allow the use of "array processor" routines. Yilmaz (1989) used SVD decompositions, which can be prohibitively expensive, especially since we have to perform an inversion for each frequency!

But, as Kostov (1989) already showed in the case of the slant-stack transform, I will show that the matrix being inverted in this computation has indeed a Toeplitz structure. Thus, the inversion can be performed very easily with Levinson algorithm, and the cost of the process is substantially reduced. Moreover, we can also include offset-dependent weighting without losing this structure; these properties are independent of the space sampling, may it be regular or irregular. For these reasons, this method, already attractive for its efficiency, should be even more appealing and used on a large scale.

This paper will essentially focus on the inversion operation. I will first recall the philosophy of the multiples elimination process, which explains the use of parabolic transforms. Then, I will define the forward and inverse parabolic transforms, and show how the Toeplitz structure appears and can be used. Finally, I will use real data, first to show the efficiency of the multiples elimination process for water-bottom multiples and peglegs. I also extend the applications to interpolation problems, where I will take advantage of the Toeplitz structure to use an offset-dependent weighting on irregularly sampled data.

WHY PARABOLIC TRANSFORMS?

Parabolic shape of the multiples

Let's suppose the NMO-correction of a CMP gather is performed with the velocity curve of the primaries (velocity V_p). A multiple with RMS velocity V_m and 0-offset time t_0 will be moved to the position:

$$t(h) = t_0 + \sqrt{t_0^2 + \frac{h^2}{V_m^2}} - \sqrt{t_0^2 + \frac{h^2}{V_p^2}} \tag{1}$$

Defining the residual slowness $1/V_r$ and the parameter p by:

$$\frac{1}{V_r^2} = \frac{1}{V_m^2} - \frac{1}{V_p^2} \ , \qquad p = 1/(2t_0V_r^2) \ ,$$

a series expansion of $t(h)$ will give:

$$t(h) \simeq t_0 + ph^2 \quad (\text{for } \frac{h}{t_0V_r} \ll 1). \tag{2}$$

V_r^2 is guaranteed to be positive, since we expect the velocity of a multiple to be smaller than the velocity of a primary at the same 0-offset time. So, for "not too large" offsets, the shape of a multiple turns out to be parabolic after NMO-correction.

Multiples elimination

After NMO correction, the primaries are described by approximately horizontal lines: their corresponding p-parameters are close to 0. On the other hand, multiples, and especially long-period multiples, should be characterized by large p-parameters.

Thus, the idea is to transform the NMO-corrected data set $d(t,h)$ into another set of values $U(t_0,p)$, where t_0 and p are the parameters describing the parabolas previously defined. Once we have these values $U(t_0,p)$, the separation of the multiples and the primaries is determined by the choice of a cut-off p-parameter: above this parameter, we set all $U(t_0,p)$ to 0. To avoid truncation effects, the setting to 0 actually uses a tapering process.

Some restrictions apply. The multiples should be long-period, to have large values of p. Also, the parabolic modeling is less valid at far offsets. Finally, it is preferable to have only flat reflectors to have this parabolic modeling. However, I haven't yet studied the effect of dip on the parabolic transforms.

Finally, the problem is now how to transform the data $d(t,h)$ into the values $U(t_0,p)$, and vice-versa.

EXPRESSIONS OF THE PARABOLIC TRANSFORMS

Modeling operation

The modeling equation is similar to Thorson's equation for hyperbolic transforms. It expresses the NMO-corrected data $d(t,h)$ as a function of the field U in the $t_0 - p$ domain:

$$d(t,h) = \sum_{t_0} \sum_{p} U(t_0,p)\delta[t_0 - (t - ph^2)] \tag{3}$$

whereas, for Thorson's equation, the δ term has to be replaced by $\delta[\,t_0 - \sqrt{t^2 - p^2h^2}\,]$. Similarly, it can be compared to the slant-stack modeling:

$$d(t,h) = \sum_{\tau} \sum_{p} U(\tau,p)\delta[\tau - (t - ph)]\ .$$

Equation (3) will transform a spike in the $t_0 - p$ domain into a parabola (or straight line if $p = 0$) in the time-offset domain.

I symbolize equation (3) as: $d = L.U$. Equation (3) can be expressed in the frequency domain. For the $\tau - p$ transform, Kostov (1989) showed that:

$$d(\omega,h) = \sum_{p} U(\omega,p)e^{-j\omega ph}\ .$$

For the parabolic transform, we get:

$$d(\omega,h) = \sum_{p} U(\omega,p)e^{-j\omega ph^2} \tag{4}$$

So, for a fixed ω, an element of the matrix L is:

$$L_{ik} = \exp(-j\omega p_k h_i^2)\ .$$

Stacking along parabolas actually corresponds to computing $U = L^T.d$ (as Thorson pointed out in the case of hyperbolic stacking). We will transform the data $d(t,h)$ into the $t_0 - p$ domain by using the least-squares inverse of L, i.e. $(L^TL)^{-1}L^T$ in the overdetermined case (or $L^T(LL^T)^{-1}$ in the underdetermined case):

$$U(\omega,.) = (L^TL)^{-1}.L^Td(\omega,.) \qquad (\omega \text{ is fixed}) \tag{5}$$

Inverse transformation

In the case of hyperbolic transforms, the inversion (5) is expensive. However, for parabolic as for slant-stack transforms, the computation of $(L^TL)^{-1}L^T$ is easy once we transform the initial data into the frequency domain. First, the size of the matrix L is consequently reduced: we have to invert matrices of size $n_p \times n_p$ (n_p is the number of p-parameters used in the computation).

What is more important is the Toeplitz structure of the matrix L^TL. Kostov (1989) showed this property in the case of the slant-stack transform. This property still holds for the parabolic transform. Effectively, assuming that the p-parameters are regularly sampled (with a step δp), $(L^TL)_{ik}$ is independent of $(i-k)$:

$$(L^TL)_{ik} = \sum_h \exp(j\omega(i-k)(\delta p)h^2)\ .$$

This diagonal invariance defines the Toeplitz structure of the matrix L^TL.

Moreover, introducing an offset-dependent weighting does not perturb this Toeplitz structure. Effectively, this is equivalent to using a diagonal positive matrix W to compute the transform:

$$U(\omega,.) = (L^TWL)^{-1}.L^TWd(\omega,.)\ .$$

But, if $W = diag(w(h))$, we see that:

$$(L^TWL)_{ik} = \sum_h w(h)\exp(j\omega(i-k)(\delta p)h^2)\ ,$$

and the diagonal invariance, so the Toeplitz structure, are preserved.

Notice also that this Toeplitz structure holds for any kind of spatial sampling: particularly, it is true for any *irregular* sampling, for example if some traces are missing. Finally, we can even choose a frequency-dependent weighting, since the inversions are performed separately at each frequency. For example, if the far offsets are not sufficiently muted after NMO correction, their low-frequency content due to stretching could influence the inversion process. Thus, it can seem reasonable to give low weights at low frequencies to these far offsets.

Consequently, using Levinson algorithm, the transform $d(t,h) \rightarrow U(t_0,p)$ turns out to be very easy in the frequency domain. For each frequency, if we use n_p parameters p, it requires $O(n_p^2)$ operations, compared to $O(n_p^3)$ operations for more general methods, like Cholevski or SVD decompositions, or the conjugate-gradient method.

Restrictions

Up to now, I only spoke of the *overdetermined* problem, which implies the inversion of the matrix L^TL. Things are different in the *underdetermined* case, where we have to invert LL^T. Effectively, this matrix is not Toeplitz, even for a regular sampling in offset. The expression of a component of LL^T is:

$$(LL^T)_{ik} = \sum_p \exp(j\omega p(h_i^2 - h_k^2)) \ .$$

With a regular sampling δh in offset, supposing h_0=0, this term becomes:

$$(LL^T)_{ik} = \sum_p \exp(j\omega p(i^2 - k^2)(\delta h)^2) \ .$$

Consequently, $(LL^T)_{ik}$ does not depend only on $i-k$, and the structure of the matrix LL^T is not Toeplitz. So, it is not possible to use Levinson algorithm in the underdetermined transformation $d(t,h) \rightarrow U(t_0,p)$, where we have more parameters p than offsets h.

We can reason similarly for the transform $U(t_0,p) \rightarrow d(t,h)$. Suppose that, rather than using simply the modeling matrix L, we want to compute the least-squares inverse of L^T: $(LL^T)^{-1}L$ in the overdetermined case, $(L^TL)^{-1}L^T$ in the underdetermined case. Thus, Levinson algorithm could only be applied in the underdetermined case.

Summary of the transforms

Starting from a CMP-gather, we apply a NMO correction with the velocity curve of the primaries, obtaining a data set $d(t,h)$. Then, we apply a Fourier transform in the time direction to transform the data set to $d(\omega,h)$. For each value of ω, we obtain the field $U(\omega,p)$ by solving the system:

$$(L^TWL)U(\omega,.) = L^TWd(\omega,.) \ .$$

This system is solved using Levinson algorithm, since the matrix L^TWL is Toeplitz. This property is independent of the weighting matrix W; it just needs to be diagonal. It is also true for any spatial sampling.

We get the field $U(t_0,p)$ by applying the inverse Fourier in the time direction. After having filtered the field U (for multiple removal, for instance), we come back to the time-offset domain with the modeling operator L. To do so, we transform the field U to the frequency domain, and for each frequency ω, we multiply it by the matrix L (depending on ω):

$$d(\omega, .) = L.U(\omega, .) \ .$$

Finally, applying an inverse Fourier transform and an inverse NMO correction brings us back to the original time-offset domain.

EXAMPLES ON REAL DATA

Different applications of the parabolic transforms

Basically, the parabolic transforms maps the NMO-corrected data to the $t_0 - p$ domain. The main seismic events will transform to different regions of the mapping. Thus, by putting to 0 the values of $U(t_0, p)$ inside (or outside) these regions, we should be able to separate the events according to their nature.

I mentioned already the multiples elimination method, proposed by Hampson (1986). The primaries should correspond to small values of p, while the multiples (especially water-bottom multiples and peglegs) should map to large values of p. However, when the primaries are too deep, their peglegs tend to have also small p-parameters, since $p = 1/(t_0 V_r^2)$, and the time t_0 is large: it is more difficult to separate these multiples from the primaries at the same t_0.

Then, for land data, the ground-roll, having a very low velocity, should map to regions of very large values of p. Consequently, canceling the $U(t_0, p)$ in this region should suppress the ground-roll.

Finally, as we will see on real marine data, the direct arrivals and their low-frequency trend will also map to large values of p, because they have a linear moveout. However, they actually cover a vast region of the $t_0 - p$ domain, and it is more difficult to suppress them without affecting the other events.

However, the role of the mute is essential. After NMO correction, the seismic events at far offsets are stretched, especially for small 0-offset times. Then, they will map to a vast region of p, since they can be described by several parabolas. If a mute is not sufficient, it is also possible to give low weights to these far offsets, since the Toeplitz structure makes weighting easy to use. Furthermore, the seismic events at far offsets don't fit the parabolic modeling, which justifies in this context the under-weighting in this region.

Actually, I will use for my examples two marine shot-gathers, known to contain no dipping events, nor important diffractions. Consequently, the NMO correction can be reliable. The first one contains very strong long-period water-bottom multiples, and I will show on it the efficiency of the multiples elimination. The second one will be used for interpolation purposes. Some traces are missing : the mapping to the

$t_0 - p$ domain will be performed with irregular sampling, but the mapping back to the time-offset will use a regular sampling, and thus interpolate the missing traces. On this example, I will also use offset-dependent weights, to stress the importance of traces near the missing traces.

Multiples elimination

I use here the wz24 (Western Geophysical) shot gather. It was recorded in Alaska, which explains a hard ocean floor, due to arctic conditions. This hard ocean floor creates strong water-bottom multiples.

On Figure 1, I display the input shot gather, with the description of the recording parameters. The water-bottom multiples are easily visible: they are separated by lags of 0.5 seconds. Next to the input shot-gather, I also display a first attempt of removal of these multiples, with a method I describe in this report (Darche, 1990). It consists in doing a Burg-type adaptive processing in the $\tau - p$ domain, after a slant-stack transform. This process is especially useful to have a good estimate of the velocity curve of the primaries: a velocity analysis of the input gather only reveals velocity picks due to the strong multiples. However, I don't consider it as optimal in this case, because the multiples are not efficiently removed at far offsets. The velocity curve obtained from the first multiple removal enables us to perform the NMO correction of the input gather, and the result is also displayed on Figure 1.

Then, I transform the NMO-corrected data to the $t_0 - p$ domain, applying the least-squares inverse $(L^T L)^{-1} L^T$ in the frequency domain, with uniform weights. The advantage of this data-set is that the velocities of the primaries increase very rapidly with time, so that the multiples have far smaller velocities: their parabolic shape is more evident. The result is displayed on Figure 2, where I also mention the parameters of the transform.

On Figure 2, we can notice three basic topics. First, due to the stretch at far offsets, the water-bottom reflection does not focus around p=0, because of the stretch of the far-offsets after NMO correction. Then, the water-bottom multiples are clearly visible at 1 and 1.5 second, with two peaks around $p = 0.1\ s.km^{-2}$. I isolated the region in which I find the multiples with a thick boundary line. Finally, no major event is visible after 2 seconds.

I isolate the water-bottom multiples by putting to 0 the values $U(t_0, p)$ outside the corresponding region, described on Figure 2. The choice of the cut-off parameters is rather subjective, but is sufficient to prove the efficiency of the separation. I transform back the remaining values to the time-offset domain with the modeling operator L, and finally apply an inverse NMO correction. Thus, I obtain the field of multiples in the original shot-gather, as represented on Figure 3. The primary field is obtained by difference, and is also represented on Figure 3. You can also compare them with the results obtained by Zhang (1990; this report), who applied a migration scheme to remove the water-bottom multiples on the same shot-gather.

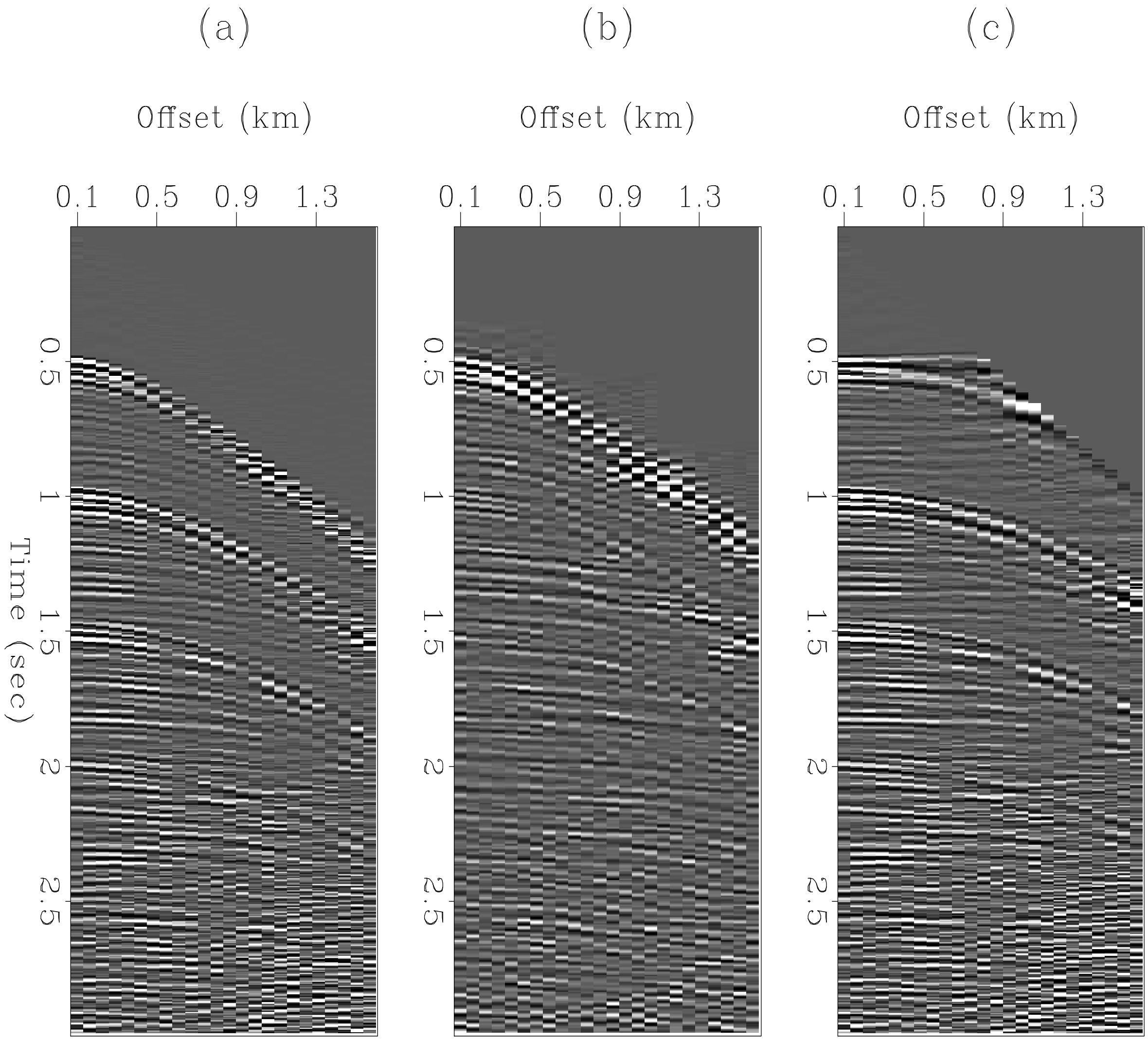

Figure 1: Shot-gather wz24. (a) Input: dt=4 msec, 3 sec of data; 24 traces, distant of 65 meters. (b) After multiples elimination in the $\tau - p$ domain, with a Burg-type adaptive algorithm ($\lambda = 0.98$). (c) after NMO correction of the input gather, using a velocity analysis performed on (b); mute applied on the last traces. gilles2-H24input [NR]

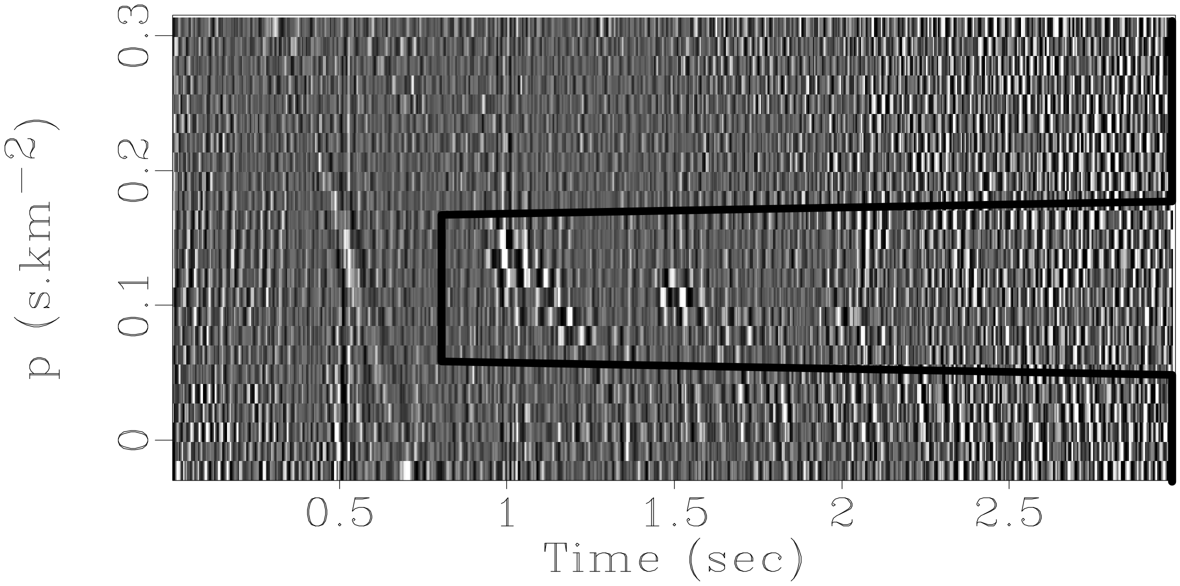

Figure 2: Parabolic mapping of the NMO-corrected shot gather. I used 24 parameters p, so that the system is still overdetermined. I also superposed the boundary of the region limiting the multiples, in which I will put the values to 0. gilles2-H24nmopar [NR]

The multiples have been efficiently removed. Some major events, especially after 1.5 seconds, have been emphasized. I think a problem can come from the definition of the multiple region: if we take it too wide (in p), we localize the energy of the remaining events in a narrow band of the $t_0 - p$ domain, as though they were purely parabolic, and we could lose some amplitude-versus-offset information.

Finally, I advise *not* to work directly with the region defining the *primaries* on Figure 2. Effectively, it appears that the water-bottom reflection, especially, should be described by a wider range of p-parameters than the one I used: the modeling operator could not restitute it entirely, and it would be modified in the process. This is due in particular to the stretching at far-offsets, to the mute, and also to refracted arrivals. On the other hand, the multiples are known to be approximately parabolic, and their energy is concentrated in the region I defined, so we are sure to restitute them entirely.

Interpolation

I now use another shot-gather, wz27 (Western Geophysical), also from Alaska. It also contains strong water-bottom multiples, but this is not the subject of this section. This shot-gather is interesting, because it contains 119 traces, but every other trace in the second half of the array is missing.

On Figure 4, I display the input shot-gather. I also represent the NMO-corrected

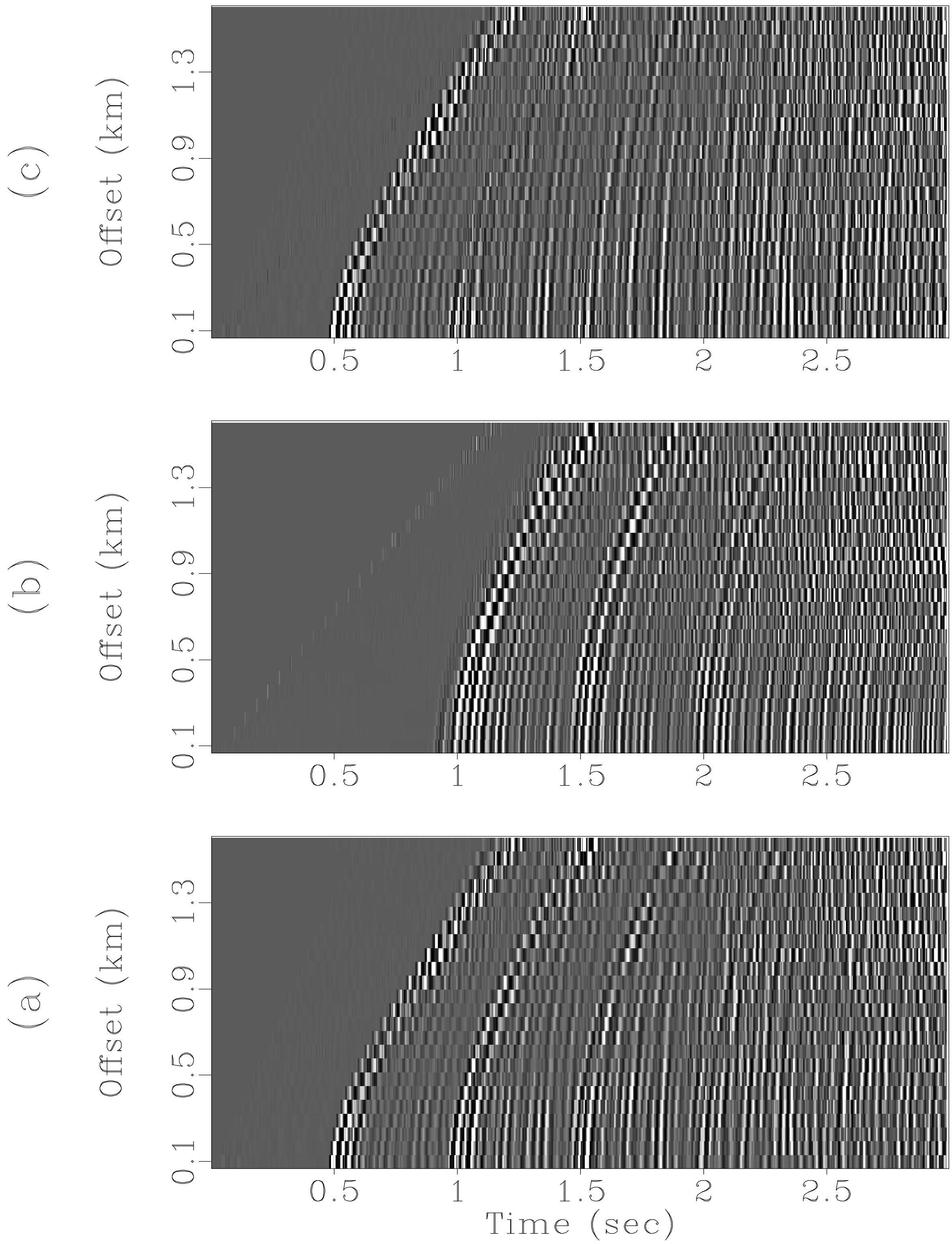

Figure 3: Separation of the multiples and primaries. (a) Input shot-gather (b) Field of the multiples. (c) Field of the primaries. $\boxed{\text{gilles2-H24result}}$ [NR]

data; the velocity analysis was again performed after a first attempt to remove of the water-bottom multiples in the $\tau - p$ domain (Darche, 1990).

To perform the interpolation, I will first transform the data to the $t_0 - p$ domain. But, to compute the least-squares inverse of the modeling operator L, I will use an irregular space sampling, since I don't want to include the missing traces in the transform. I will apply equation (5) with the set of offsets of the real traces (94 traces at all). This is not the same as applying it to the regularly sampled gather, and considering the missing traces are filled with zeros: the inverse would try to reconstitute these zeros.

In fact, I overweight the far offsets, to emphasize the interpolation process on the last traces. I will use a weighting matrix W identical at all frequencies, but in which the values $w(h)$ are 1 in the first 60 traces, and 4 in the last traces, in the region where traces are missing. The corresponding values $U(t_0, p)$ appear on Figure 4.

Notice on Figure 4 that it is difficult to separate the primaries and the multiples peaks in the $t_0 - p$ domain. Effectively, the lag of the multiples is more or less 100 msec, and the multiples are rather short-period multiples. This explains why we see both kinds of events around $p = 0$. Once again, the water-bottom reflection does not focus at $p = 0$, due to the stretching at far offsets. Actually, the problem is worse than on the wz24, because some linear events (direct arrivals and low-frequency oscillations) are present on the input: after NMO correction, they cannot be modeled as parabolas.

To perform the interpolation, I use the modeling operator with modified parameters: 119 traces, regularly sampled, so that the energy in the $t_0 - p$ domain is spread over the whole space domain. I apply an inverse NMO-correction to the processed data, and I display the results on Figure 5.

The interpolation result has several interesting features. First, as expected, the far offsets have been successfully interpolated, since the parabolic transform was overweighted in this region. This is true either for the hyperbolic events (primaries and multiples) or the linear events, as it can be seen on the residuals. However, the near traces have been slightly modified. Especially, they have been roughened, losing some lateral coherency. Once again, the choice of the weighting was reducing the role of these traces, and it is not surprising to see some variations.

Actually, the choice of the weights is quite subjective. Effectively, I said that the parabolic model is not reliable at far offsets, and overweighting the far offsets might seem contradictory. On the other hand, the curvature of a parabola is better determined at far offsets. So, there is a trade-off between these criteria. I noticed on this data set that, if I underweight the last traces, the events are not well restored, and they tend to disappear on the last traces. Moreover, some strong aliased features at far offsets appear with this kind of weighting.

Another point is that it could be preferable to suppress (by dip-filtering) from the beginning the linear events, which cannot be modeled as parabolas after NMO

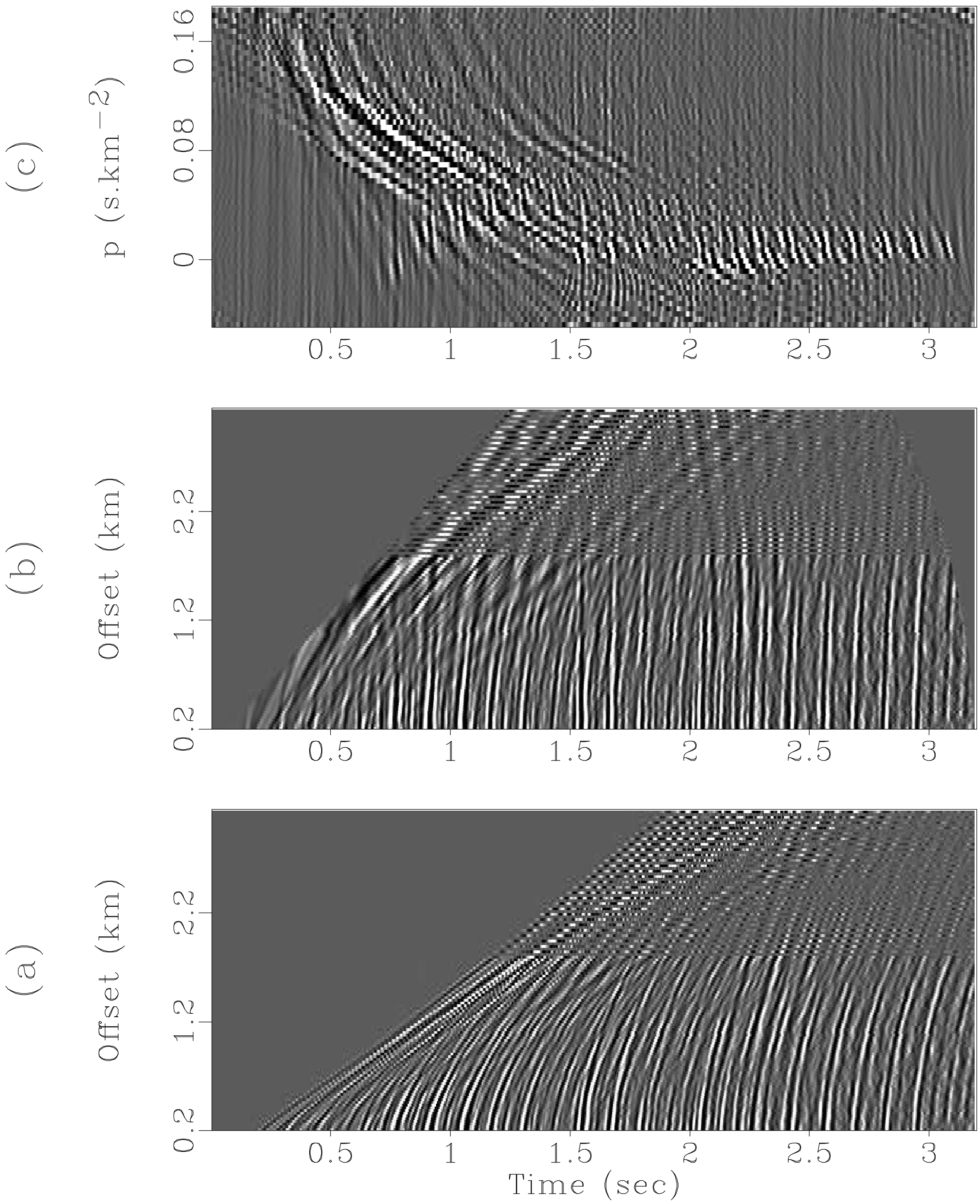

Figure 4: Shot-gather wz27. (a) Input data; dt=4 msec, 119 traces: in the last 60 last traces, every other trace is missing. (b) After NMO correction; the velocity analysis was performed after initial removal of the water-bottom multiples and peglegs. (c) Parabolic mapping; the missing traces are not taken into account in the transform, and the sampling is irregular. The far offsets are overweighted; using 94 traces in the time-offset domain, I take 60 p-parameters to have an overdetermined system. $\boxed{\text{gilles2-H27iinput}}$ [NR]

correction. If we keep them, they tend to spread their energy on a vast region of p-parameters.

In conclusion, the interpolation can be considered successful, in the sense that the missing traces have been efficiently restored. The choice of the weights matters, according to the purpose of the process: not disturbing the near traces (far offsets underweighted), or restoring precisely the far traces (far offsets overweighted). A trade-off is to choose an uniform weighting.

CONCLUSION

The applications of parabolic transforms to NMO-corrected data range from multiple and noise removal to interpolation processes. I showed in this paper that they can be computed easily using Levinson algorithm, which makes their use very attractive. Moreover, this efficiency is still valid when we use irregularly sampled data, and offset-dependent weighting, which is particularly useful, for example for interpolation purposes. Some practical problems remain, especially at far offsets: stretching after NMO, deviation from the parabolic shape, but they can be avoided by using an appropriate offset-dependent weighting.

ACKNOWLEDGEMENTS

I thank Clement Kostov for having introduced me to the world of slant-stack, hyperbolic and parabolic transforms. I am also grateful to Lin Zhang for having worked for the same purpose on the data I used: it helped me a lot to check the possible mistakes in my methods (and my programs!).

REFERENCES

Darche, G., 1990, Generalized adaptive deconvolution: SEP **65**.

Hampson, D., 1986a, Inverse velocity stacking for multiple elimination: SEG abstracts.

Hampson, D., 1986b, Informal communication.

Kostov, C. , 1989, Slant-stack transforms: SEP **61**, 261-279.

Thorson, J., 1984, Velocity and slant-stack transforms: Ph.D. thesis, Stanford U., SEP **39**.

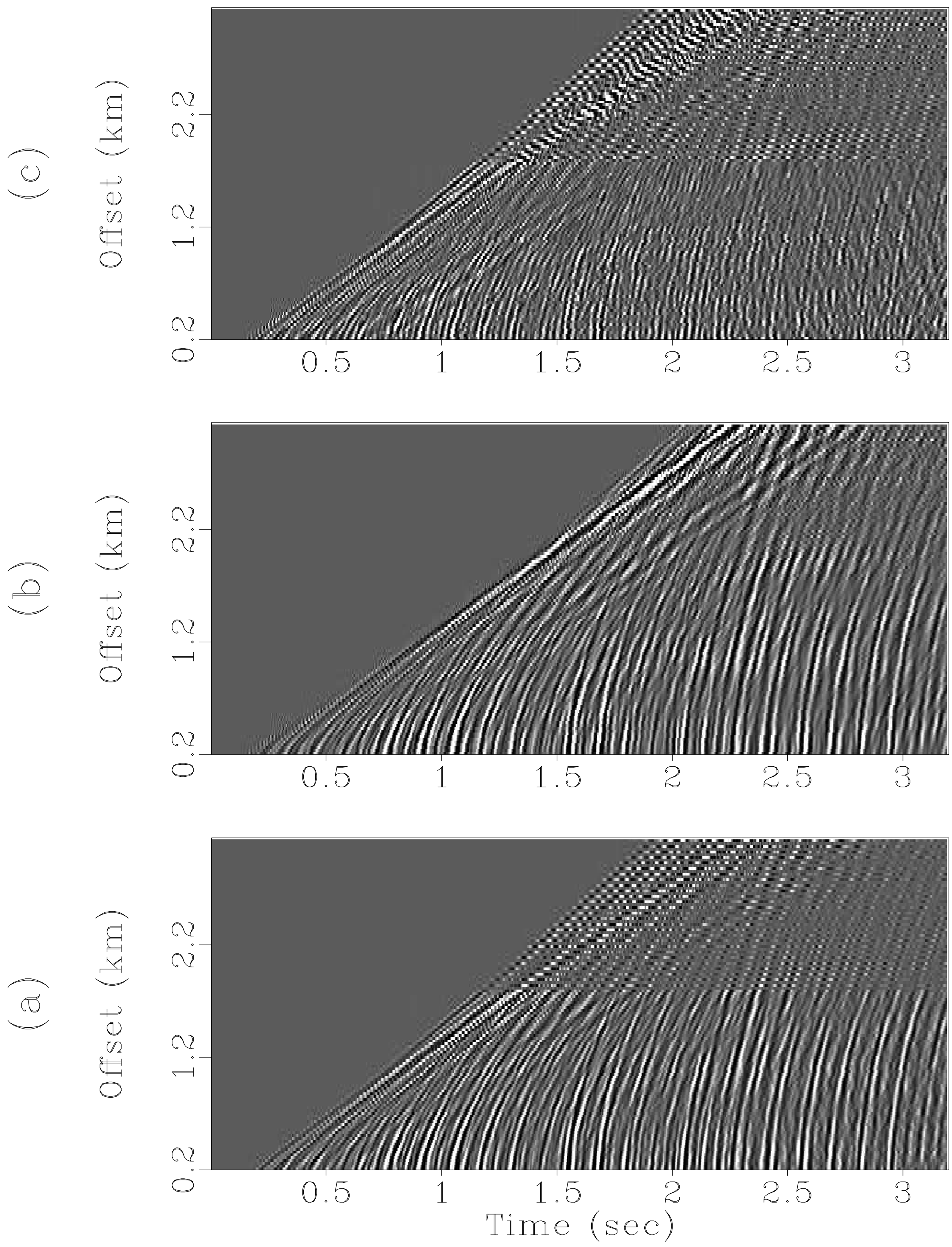

Figure 5: Interpolation result. (a) Input gather. (b) Interpolated gather; starting from the parabolic mapping of Figure 4, I come back to the time-offset domain, using a modeling operator with 119 offsets. (c) Difference between the input and the interpolated gathers. gilles2-H27iresult [NR]

Yilmaz, O., 1989, Velocity-stack processing: Geophysical processing, vol **37**, 357-382.

Zhang, L., 1990, Eliminating the water-bottom multiples by imaging: SEP **65**.