A FAST UNCOILED RANDOMIZED QR DECOMPOSITION METHOD FOR 5D SEISMIC DATA RECONSTRUCTION

SHENGHOU WANG[1], JIANJUN GAO[1,2] and JINGYE LI [2]

[1] *School of Geophysics and Information Technology, China University of Geosciences (Beijing), 29 Xueyuan Road, Haidian District, Beijing 100083, P.R. China.*
gaojianjun@cugb.edu.cn
[2] *State Key Laboratory of Petroleum Resources and Prospecting (China University of Petroleum, Beijing), Beijing 102249, P.R. China.*

(Received August 30, 2017; revised version accepted April 10, 2018)

ABSTRACT

Wang, S., Gao, J. and Li, J., 2018. A fast uncoiled randomized QR decomposition method for 5D seismic data reconstruction. *Journal of Seismic Exploration*, 27: 255-276.

The low rank matrix completion methods have been widely applied to reconstruct multidimensional irregular seismic data. The existing literature shows that well sampled seismic data can be represented by a low rank block Hankel or block Toeplitz matrix. However, incomplete data and random noise can destroy the low rank property of the block matrix. Hence, the recovery of missing seismic traces can be treated as a rank reduction problem. This paper presents a fast rank reduction algorithm named randomized QR decomposition to interpolate the pre-stack 5D irregular missing seismic traces. Compared with the popular matrix rank reduction algorithms, such as the Singular Value Decomposition (SVD) and the Lanczos bidiagonalization decomposition method, this method has higher computational efficiency and faster reconstruction speed. Moreover, for the computationally low efficient problem of the diagonal averaging operation of the rank-reduced level-4 block Toeplitz matrix, a fast uncoiled diagonal averaging strategy is designed. The new diagonal averaging algorithm can greatly reduce the amount of data storage and decrease the computational cost. In the end, the validity of the proposed method is verified by synthetic data experiments and a field data test.

KEY WORDS: rank reduction, randomized QR decomposition, multilevel Toeplitz structures, 5D seismic data reconstruction.

0963-0651/18/$5.00

INTRODUCTION

The irregular missing seismic data usually cause bad effects on multichannel processing, such as the surface-related multiple elimination (SRME), wave-equation migration, time-lapse seismic repeatability and so on. Therefore, it is necessary to do seismic data reconstruction before the multichannel seismic data processing. In the past few years, four main categories of reconstruction methods are developed to interpolate the irregular missing seismic traces. The first category of methods is based on the mathematical transformations, for example, the Radon transform (Trad et al., 2002; Xue et al., 2014; Tang and Mao, 2014), the Fourier transform (Duijndam et al., 1999; Liu and Sacchi, 2004; Xu et al., 2005; Zwartjes and Gisolf, 2007; Gao et al., 2010, 2013a), the Curvelet transform (Herrmann and Hennenfent, 2008; Naghizadeh and Sacchi, 2010; Yang et al., 2012; Cao and Wang, 2015; Zhang et al.,2017), the Seislet transform (Gan et al., 2015; Gan et al., 2016; Liu et al., 2016) and the Dreamlet transform (Wang et al., 2014, 2015). The second category of methods is based on the prediction error filtering (Spitz, 1991; Porsani, 1999; Naghizadeh and Sacchi, 2007, 2009). These methods utilize the predictability of linear events in the frequency-space domain to interpolate the missing traces. The third category is based on wave equation principles (Ronen, 1987; Kaplan et al., 2010; Kutscha and Verschuur, 2012). The methods in this category usually require some subsurface information to do the reconstruction, especially the velocity information. The fourth category of methods is based on the rank reduction theory. These methods presume that the properly sampled ideal data can be represented by a low-rank matrix or tensor. However, missing traces and random noise in the observed data will destroy the low-rank property and increase the rank of the matrix or tensor. Therefore, seismic data reconstruction can be treated as a rank reduction problem. In these methods, Trickett et al. (2010) developed a Cadzow Filtering (CF) method to interpolate the irregular missing traces. Oropeza and Sacchi (2011) developed a matrix rank reduction method named Multichannel Singular Spectrum Analysis (MSSA) to interpolate the 3D irregular seismic data and adopted a Randomized SVD algorithm to accelerate the rank reduction operation. Gao et al. (2013b) developed the MSSA method and proposed a fast rank-reduction algorithm, which combined the Lanczos bidiagonalization algorithm and a fast level-4 block Toeplitz matrix-vector multiplication via the 4D Fast Fourier Transform (FFT) algorithm in order to reconstruct 4D spatial data. Furthermore, Jia et al. (2016) also developed a fast orthogonal rank-one matrix pursuit (OR1MP) algorithm to improve the calculation efficiency for the rank reduction process in the MSSA reconstruction method. In addition, Chiron et al. (2014) developed an efficient randomized QR decomposition method for the de-noising of spectroscopic data. Cheng and Sacchi (2016) utilized the randomized QR decomposition algorithm to reduce the rank of a level-2 block Hankel matrix and presented a fast anti-diagonal averaging strategy for the block Hankel matrix to reconstruct 3D irregular seismic data.

In this paper, we introduce the compressed sensing theory into the matrix rank reduction and utilize it to decrease the size of the huge level-4 block Toeplitz matrix ahead of the rank reduction operation. Thereupon, a new fast rank reduction method named the uncoiled randomized QR decomposition (abbreviated as URQR) is developed to reconstruct the pre-stack 5D seismic volume. The URQR method possesses the following merits, firstly, a column linearly independent random matrix is constructed and utilized to decrease the size of the huge level-4 block Toeplitz matrix before implementing the rank reduction operation. Secondly, the QR decomposition algorithm rather than the truncated SVD or the Lanczos bidiagonalization algorithm is adopted to reduce the rank of this compressed size matrix. Thirdly, a fast uncoiled diagonal averaging algorithm during the process of averaging along diagonals of the level-4 Toeplitz matrix is developed to avoid the construction and storage of the huge rank-reduced level-4 block Toeplitz matrix. As a result, the computer memory footprint can be greatly decreased. In addition, a fast level-4 block Toeplitz matrix-vector multiplication algorithm via 4D FFT technique is adopted to improve the computational efficiency of matrix-matrix and matrix-vector multiplication in the stage of rank reduction and diagonal averaging processing.

THEORY

The existing literature indicates that well-sampled ideal 5D seismic data can be embedded into a low-rank level-4 block Toeplitz matrix for each monochromatic frequency slice in the frequency domain (Oropeza and Sacchi, 2011; Gao et al., 2013b). Any missing traces and random noise will increase the rank of this block matrix. Hence, the derivation of the proposed URQR algorithm is also based on this principle. In addition, similar to the treatment of other peers, the low rank reconstruction method is implemented in the frequency-midpoint-offset domain in this paper.

Randomized QR decomposition theory

In the frequency-midpoint-offset domain, the observed pre-stack 5D seismic is denoted by $D^{obs}(f, cmp_x, cmp_y, h_x, h_y)$, where the index variables cmp_x, cmp_y, h_x and h_y represent the spatial coordinates of the inline, crossline midpoint and inline, crossline offset. We define the seismic volume for one single frequency slice via a 4D array $\mathbf{D}^{obs}$ with elements $\mathbf{D}^{obs}_{i_1,i_2,i_3,i_4}$ after the data are binned in the midpoint-offset domain, where the indices i_1, i_2, i_3 and i_4 denote the spatial coordinates cmp_x, cmp_y, h_x and h_y, respectively. For each monochromatic frequency f, the seismic data can be embedded into a level-4 block Toeplitz matrix $\mathbf{A}^{(4)}$ with size $(L_1L_2L_3L_4)\times(K_1K_2K_3K_4)$, where $L_j + K_j - 1 = N_j,\ \ j = 1,2,3,4,\ \ N_j$ represents the number of traces in the spatial j-th direction of $\mathbf{D}^{obs}$, L_j is usually chosen to be $L_j = \lfloor N_j/2 \rfloor + 1$, where $\lfloor \cdot \rfloor$ indicates the integer part of its argument. If $\mathbf{A}^{(4)}$ is constructed by the well sampled regular data, $\mathbf{A}^{(4)}$ has the low rank property. Assume

the rank of $\mathbf{A}^{(4)}$ is r, $r \le \min\{L_1L_2L_3L_4, K_1K_2K_3K_4\}$, we adopt the randomized QR decomposition algorithm (Halko et al., 2011, Chiron et al., 2014, Cheng and Sacchi, 2016) to reduce the rank of $\mathbf{A}^{(4)}$. First of all, we generate a column linearly independent random matrix $\mathbf{\Omega}$ to shrink the size of matrix $\mathbf{A}^{(4)}$ and obtain the compressed size matrix $\mathbf{B}$,

$$\mathbf{B} = \mathbf{A}^{(4)}\mathbf{\Omega} \quad , \tag{1}$$

here, the size of $\mathbf{\Omega}$ is $(K_1K_2K_3K_4)\times r$. The size of $\mathbf{B}$ is $(L_1L_2L_3L_4)\times r$.

Then QR decomposition is implemented on matrix $\mathbf{B}$,

$$\mathbf{B} = \mathbf{QR} \quad , \tag{2}$$

where $\mathbf{Q}$ is a small $(L_1L_2L_3L_4)\times r$ orthogonal matrix and $\mathbf{R}$ is a $r\times r$ upper triangular matrix. Next, projecting the orthonormal basis matrix $\mathbf{Q}$ onto $\mathbf{A}^{(4)}$, the low-rank approximation matrix $\tilde{\mathbf{A}}^{(4)}$ of $\mathbf{A}^{(4)}$ is computed as,

$$\tilde{\mathbf{A}}^{(4)} = \mathbf{Q}\mathbf{Q}^{H}\mathbf{A}^{(4)} . \tag{3}$$

Finally, apply the diagonal average of the rank-reduced matrix $\tilde{\mathbf{A}}^{(4)}$ to recover the missing samples in the 4D spatial data $\mathbf{D}$

$$\tilde{\mathbf{D}}_{i_1,i_2,i_3,i_4} = \left\{ \sum_{j_4=j_{\min}^{(4)}}^{j_{\max}^{(4)}} \sum_{j_3=j_{\min}^{(3)}}^{j_{\max}^{(3)}} \sum_{j_2=j_{\min}^{(2)}}^{j_{\max}^{(2)}} \sum_{j_1=j_{\min}^{(1)}}^{j_{\max}^{(1)}} \tilde{\mathbf{A}}^{(4)}_{p_4,v_4} \right\} \times \mathbf{W}_{i_1,i_2,i_3,i_4} , \tag{4}$$

where $\mathbf{W}$ is a 4D averaging operator whose elements are the diagonal averaging coefficients. The size of operator $\mathbf{W}$ is the same as $\tilde{\mathbf{D}}$. The detailed definition of $\mathbf{W}$ is shown in Appendix A. In addition,

$$j_{\min}^{(k)} = \max(i_k - K_k + 1, 1), \qquad j_{\max}^{(k)} = \min(i_k, L_k), k = 1,2,3,4,$$

$$p_4 = (j_4 - 1)L_3L_2L_1 + (j_3 - 1)L_2L_1 + (j_2 - 1)L_1 + j_1,$$

$$v_4 = (K_4 - i_4 + j_4 - 1)K_3K_2K_1 + (K_3 - i_3 + j_3 - 1)K_2K_1 + (K_2 - i_2 + j_2 - 1)K_1 + K_1 - i_1 + j_1$$

For eq. (4), the present popular averaging calculation methods first require to compute and store $\tilde{\mathbf{A}}^{(4)}$ in eq. (3), and then average along the diagonals of $\tilde{\mathbf{A}}^{(4)}$. The calculation and storage of matrix $\tilde{\mathbf{A}}^{(4)}$ makes the low rank matrix completion method unfeasible for large seismic field data reconstruction.

Fast level-4 block Toeplitz matrix-vector multiplication via 4D FFT

In order to decrease the computational cost of matrix naive multiplication operations in eq. (1) and eq. (3), the multiplication among the ordinary matrix $\mathbf{\Omega}$, $\mathbf{Q}^H$ and the level-4 block Toeplitz matrix $\mathbf{A}^{(4)}$ can be efficiently implemented by the fast matrix-vector multiplication algorithm via a 4D FFT (Gao et al., 2013b). For a monochromatic frequency *f*, the seismic data $\mathbf{D}^{obs}$ can be embedded into a level-4 block Toeplitz matrix,

$$\mathbf{A}^{(4)} = \begin{bmatrix} \mathbf{A}^{(3)}_{K_4} & \mathbf{A}^{(3)}_{K_4-1} & \mathrm{L} & \mathbf{A}^{(3)}_{1} \\ \mathbf{A}^{(3)}_{K_4+1} & \mathbf{A}^{(3)}_{K_4} & \mathrm{L} & \mathbf{A}^{(3)}_{2} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \mathbf{A}^{(3)}_{N_4} & \mathbf{A}^{(3)}_{N_4-1} & \mathrm{L} & \mathbf{A}^{(3)}_{L_4} \end{bmatrix}, \tag{5}$$

where, the element $\mathbf{A}^{(3)}_{i_4}$, i_4=1,2,…, N_4, is a level-3 block Toeplitz matrix and can be written as,

$$\mathbf{A}^{(3)}_{i_4} = \begin{bmatrix} \mathbf{A}^{(2)}_{K_3,i_4} & \mathbf{A}^{(2)}_{K_3-1,i_4} & \mathrm{L} & \mathbf{A}^{(2)}_{1,i_4} \\ \mathbf{A}^{(2)}_{K_3+1,i_4} & \mathbf{A}^{(2)}_{K_3,i_4} & \mathrm{L} & \mathbf{A}^{(2)}_{2,i_4} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \mathbf{A}^{(2)}_{N_3,i_4} & \mathbf{A}^{(2)}_{N_3-1,i_4} & \mathrm{L} & \mathbf{A}^{(2)}_{L_3,i_4} \end{bmatrix}, \tag{6}$$

here, the element $\mathbf{A}^{(2)}_{i_3,i_4}$, i_3=1,2,…, N_3, is a level-2 block Toeplitz matrix,

$$\mathbf{A}^{(2)}_{i_3,i_4} = \begin{bmatrix} \mathbf{A}^{(1)}_{K_2,i_3,i_4} & \mathbf{A}^{(1)}_{K_2-1,i_3,i_4} & \mathrm{L} & \mathbf{A}^{(1)}_{1,i_3,i_4} \\ \mathbf{A}^{(1)}_{K_2+1,i_3,i_4} & \mathbf{A}^{(1)}_{K_2,i_3,i_4} & \mathrm{L} & \mathbf{A}^{(1)}_{2,i_3,i_4} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \mathbf{A}^{(1)}_{N_2,i_3,i_4} & \mathbf{A}^{(1)}_{N_2-1,i_3,i_4} & \mathrm{L} & \mathbf{A}^{(1)}_{L_2,i_3,i_4} \end{bmatrix}, \tag{7}$$

the element $\mathbf{A}^{(1)}_{i_2,i_3,i_4}$ in eq. (7), i_2=1,2,…, N_2, is a level-1 block Toeplitz matrix which is also an ordinary Toeplitz matrix,

$$\mathbf{A}^{(1)}_{i_2,i_3,i_4} = \begin{bmatrix} \mathbf{D}^{obs}_{K_1,i_2,i_3,i_4} & \mathbf{D}^{obs}_{K_1-1,i_2,i_3,i_4} & \mathrm{L} & \mathbf{D}^{obs}_{1,i_2,i_3,i_4} \\ \mathbf{D}^{obs}_{K_1+1,i_2,i_3,i_4} & \mathbf{D}^{obs}_{K_1,i_2,i_3,i_4} & \mathrm{L} & \mathbf{D}^{obs}_{2,i_2,i_3,i_4} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \mathbf{D}^{obs}_{N_1,i_2,i_3,i_4} & \mathbf{D}^{obs}_{N_1-1,i_2,i_3,i_4} & \mathrm{L} & \mathbf{D}^{obs}_{L_1,i_2,i_3,i_4} \end{bmatrix}. \tag{8}$$

The more detailed description of $\mathbf{A}^{(4)}$ can be seen in Gao et al. (2013b). Considering $\boldsymbol{x}$ is a $(K_1K_2K_3K_4)\times 1$ column vector, the naive multiplication between the level-4 block Toeplitz matrix $\mathbf{A}^{(4)}$ and the column vector $\boldsymbol{x}$ can be implemented by the 4D FFT technique. First of all, one requires to expand the level-4 block Toeplitz matrix $\mathbf{A}^{(4)}$ to a level-4 block circulant matrix $\mathbf{C}^{(4)}$ with size $M\times M$ by appending zeros to each dimension of $\mathbf{D}^{obs}$, where $M = M_1M_2M_3M_4$, $M_i = 2^{\lceil \log_2 N_i \rceil}$, i = 1,2,3,4, where $\lceil\cdot\rceil$ represents the upper integer of its argument. Secondly, a 4D array $\tilde{\mathbf{C}}^{(4)}$ is formed to store the first column elements of each sub-block matrix of $\mathbf{C}^{(4)}$. The elements of each dimension of $\tilde{\mathbf{C}}^{(4)}$ can be directly obtained from the data $\mathbf{D}^{obs}$,

$$\tilde{\mathbf{C}}^{(4)}_{i_1,i_2,i_3,i_4} = \begin{cases} \mathbf{D}^{obs}_{n_1,n_2,n_3,n_4}, & i_j = 1,\ldots,M_j - K_j + 1, \\ \mathbf{D}^{obs}_{m_1,m_2,m_3,m_4}, & i_j = M_j - K_j + 2,\ldots,M_j \\ & j = 1,2,3,4. \end{cases}, \tag{9}$$

where $n_j = N_j - L_j + i_j$, $m_j = i_j - M_j + K_j - 1$. Finally, we reshape the vector $\boldsymbol{x}$ to a 4D array $\tilde{\mathbf{x}}$ and then expand $\tilde{\mathbf{x}}$ to have the same size as the 4D array $\tilde{\mathbf{c}}^{(4)}$ by embedding zeros. The product between $\tilde{\mathbf{c}}^{(4)}$ and $\tilde{\mathbf{x}}$ can be written as (Gao et al., 2013b),

$$\tilde{\mathbf{y}} = \mathrm{IFFT4}D[\mathrm{FFT4}D(\tilde{\mathbf{c}}) \circ \mathrm{FFT4}D(\tilde{\mathbf{x}})] \quad , \tag{10}$$

here, ‘o’ denotes the element-by-element multiplication. We extract the first $L_1\times L_2\times L_3\times L_4$ elements of the 4D array $\tilde{\mathbf{y}}$, then reshape $\tilde{\mathbf{y}}$ into a column vector $\mathbf{y}$ of size $(L_1L_2L_3L_4)\times 1$, here, $\mathbf{y} = \mathbf{A}^{(4)}\boldsymbol{x}$. The cost of the naive multiplication between the matrix $\mathbf{A}^{(4)}$ and the vector $\boldsymbol{x}$ is $O((L_1L_2L_3L_4)^2)$, however, the cost of the fast multiplication algorithm via 4D FFT technique is only $O(M\log_2 M)$. Thereupon, the matrix production of $\boldsymbol{\Omega}$ and $\mathbf{A}^{(4)}$, $\mathbf{Q}^H$ and $\mathbf{A}^{(4)}$ can be transformed into matrix-vector multiplication form and then be efficiently implemented via the fast level-4 block Toeplitz matrix-vector multiplication technique.

Fast uncoiled diagonal average of the level-4 block Toeplitz matrix

Because the requirement of constructing and storing the huge rank-reduced matrix $\tilde{\mathbf{A}}^{(4)}$ in eq. (4) causes low calculation efficiency and will require a huge computer memory, we exploit the uncoiled anti-diagonal averaging algorithm, which is formerly used for Hankel matrix rank reduction in 1D spectroscopic data de-noising (Chiron et al., 2014) to the diagonal average of the level-4 block Toeplitz matrix in 5D seismic data reconstruction. We avoid constructing and storing the level-4 block Toeplitz

matrix $\tilde{\mathbf{A}}^{(4)}$ in eq. (4) to decrease the computer memory footprint by fully exploiting the special structure of the level-4 block Toeplitz matrix (Gao et al., 2013b). Set,

$$\mathbf{U} = \mathbf{Q}^H \mathbf{A}^{(4)} \quad , \tag{11}$$

here, $\mathbf{U}$ is computed via the fast level-4 block Toeplitz matrix-vector multiplication described in the above section. Then eq. (3) is rewritten as,

$$\tilde{\mathbf{A}}^{(4)} = \mathbf{Q}\mathbf{U} \, . \tag{12}$$

The fast uncoiled diagonal average of $\tilde{\mathbf{A}}^{(4)}$, which is utilized to compute the reconstructed data $\tilde{\mathbf{D}}_{i_1,i_2,i_3,i_4}$, is expressed by,

$$\begin{aligned}
\tilde{\mathbf{D}}_{i_1,i_2,i_3,i_4} &= \left\{ \sum_{j_4=j_{\min}^{(4)}}^{j_{\max}^{(4)}} \sum_{j_3=j_{\min}^{(3)}}^{j_{\max}^{(3)}} \sum_{j_2=j_{\min}^{(2)}}^{j_{\max}^{(2)}} \sum_{j_1=j_{\min}^{(1)}}^{j_{\max}^{(1)}} \tilde{\mathbf{A}}^{(4)}_{p_4,v_4} \right\} \times \mathbf{W}_{i_1,i_2,i_3,i_4} \\
&= \{ \sum_{k=1}^{r} \sum_{j_4=j_{\min}^{(4)}}^{j_{\max}^{(4)}} \sum_{j_3=j_{\min}^{(3)}}^{j_{\max}^{(3)}} \sum_{j_2=j_{\min}^{(2)}}^{j_{\max}^{(2)}} \sum_{j_1=j_{\min}^{(1)}}^{j_{\max}^{(1)}} \mathbf{Q}_{p_4,k} \times \mathbf{U}_{k,v_4} \} \times \mathbf{W}_{i_1,i_2,i_3,i_4} \, , \\
&= \{ \sum_{k=1}^{r} \mathbf{Q}_{(k)}^{(4)} \mathbf{U}_{(k)}^{(4)} \} \times \mathbf{W}_{i_1,i_2,i_3,i_4}
\end{aligned} \tag{13}$$

where the index parameters p_4 and v_4 have the same expressions as that defined in eq. (4). The variable k denotes the k-th column of the orthogonal matrix $\mathbf{Q}$ in eq. (2) and the k-th row of the matrix $\mathbf{U}$ in eq. (12). The parameter r represents the rank of $\tilde{\mathbf{A}}^{(4)}$. $\mathbf{Q}_{(k)}^{(4)}$ denotes a level-4 block Toeplitz matrix whose first row is formed by a $1 \times K_4$ row vector $[\mathbf{Q}_1^{(3)},0,\ldots,0]$ and first column is formed by a $N_4 \times 1$ column vector $[\mathbf{Q}_1^{(3)}, \mathbf{Q}_2^{(3)},\ldots, \mathbf{Q}_{p_4}^{(3)}\ldots, \mathbf{Q}_{L_4}^{(3)},0,\ldots,0]^T$. The matrix $\mathbf{Q}_{(k)}^{(4)}$ is defined as,

$$\mathbf{Q}_{(k)}^{(4)} = \begin{bmatrix}
\mathbf{Q}_1^{(3)} & 0 & 0 & \mathrm{L} & 0 \\
\mathbf{Q}_2^{(3)} & \mathbf{Q}_1^{(3)} & 0 & \mathrm{L} & 0 \\
\mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\
\mathbf{Q}_{p_4-1}^{(3)} & \mathbf{Q}_{p_4-2}^{(3)} & \mathbf{Q}_{p_4-3}^{(3)} & \mathrm{L} & 0 \\
\mathbf{Q}_{p_4}^{(3)} & \mathbf{Q}_{p_4-1}^{(3)} & \mathbf{Q}_{p_4-2}^{(3)} & \mathrm{L} & \mathbf{Q}_1^{(3)} \\
\mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{L} & \mathrm{M} \\
\mathbf{Q}_{L_4}^{(3)} & \mathbf{Q}_{L_4-1}^{(3)} & \mathbf{Q}_{L_4-2}^{(3)} & \mathrm{L} & \mathbf{Q}_{L_4-K_4+1}^{(3)} \\
0 & \mathbf{Q}_{L_4}^{(3)} & \mathbf{Q}_{L_4-1}^{(3)} & \mathrm{L} & \mathbf{Q}_{L_4-K_4+2}^{(3)} \\
0 & 0 & \mathbf{Q}_{L_4}^{(3)} & \mathrm{L} & \mathbf{Q}_{L_4-K_4+3}^{(3)} \\
\mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\
0 & 0 & 0 & \mathrm{L} & \mathbf{Q}_{L_4}^{(3)}
\end{bmatrix}, \tag{14}$$

where $\mathbf{Q}_{p_4}^{(3)}$ is a level-3 block Toeplitz matrix, p_4=1,2,…, L_4. $\mathbf{U}_{(k)}^{(4)}$ denotes a $K_4 \times 1$ column vector $\mathbf{U}_{(k)}^{(4)} = [\mathbf{U}_{K_4}^{(3)}, \mathbf{U}_{K_4-1}^{(3)}, ..., \mathbf{U}_{\nu_4}^{(3)}, ..., \mathbf{U}_1^{(3)}]^T$, the element $\mathbf{U}_{\nu_4}^{(3)}$ denotes a $K_3 \times 1$ column vector, ν_4=1,2,…, K_4. The more detailed definition of $\mathbf{Q}_{p_4}^{(3)}$ and $\mathbf{U}_{\nu_4}^{(3)}$ can be seen in Appendix B. The product $\mathbf{Q}_{(k)}^{(4)}\mathbf{U}_{(k)}^{(4)}$ can also be computed via the fast level-4 block Toeplitz matrix-vector multiplication algorithm, which is described in eq. (10). From eq. (13), we find hat the data $\mathbf{D}_{i_1,i_2,i_3,i_4}$ can be directly obtained from the product of $\mathbf{Q}_{(k)}^{(4)}$ and $\mathbf{U}_{(k)}^{(4)}$. It is not necessary to first generate $\mathbf{A}^{(4)}$ and then do diagonal averaging to get $\mathbf{D}_{i_1,i_2,i_3,i_4}$ in eq. (4). The total multiplication cost between $\mathbf{Q}_{(k)}^{(4)}$ and $\mathbf{U}_{(k)}^{(4)}$ via the uncoiled diagonal averaging method is $O(rN\log_2 N)$, $N = N_1N_2N_3N_4$, which is less than the cost of standard averaging along diagonals $O(rKL)$ with $K = K_1K_2K_3K_4$, $L = L_1L_2L_3L_4$. The computer memory of uncoiled diagonal averaging method is $8 \times c_1c_2c_3c_4$ bytes, where $c_1 = N_1 + K_1, c_2 = N_2 + K_2$, $c_3 = N_3 + K_3$ and $c_4 = N_4 + K_4$, which is also less than the $8 \times KL$ bytes memory footprint of the standard diagonal averaging method. Let R_1 represent the computational cost ratio of the standard diagonal averaging method to the uncoiled diagonal averaging method, then

$$R_1 = \frac{r \times K \times L}{r \times N \log_2 N} \approx \frac{L}{64 + 16\log_2 L} \quad , \tag{15}$$

where we set $L_j = K_j \approx \frac{N_j}{2}$ and $N_i = N_j$, $i, j = 1,2,3,4$. Let R_2 represent the memory footprint ratio of the standard diagonal averaging method to the uncoiled diagonal averaging method,

$$R_2 = \frac{8 \times KL}{8 \times c_1c_2c_3c_4} \approx \frac{L}{81}. \tag{16}$$

We construct a series of level-4 block Toeplitz matrices with size N_i = 6, 7, …,13, i = 1,2,3,4 to show the calculation performance of the fast uncoiled diagonal averaging algorithm. Fig.1 and Fig. 2 show the ratio curve of R_1 and R_2, respectively. From Fig. 1, we see that compared with the standard diagonal averaging method, the computational cost of the uncoiled diagonal averaging method is effectively decreased. From Fig. 2, we find that the memory footprint of the uncoiled diagonal averaging method is also evidently cut down along with the increase of the dimensional size N_i.

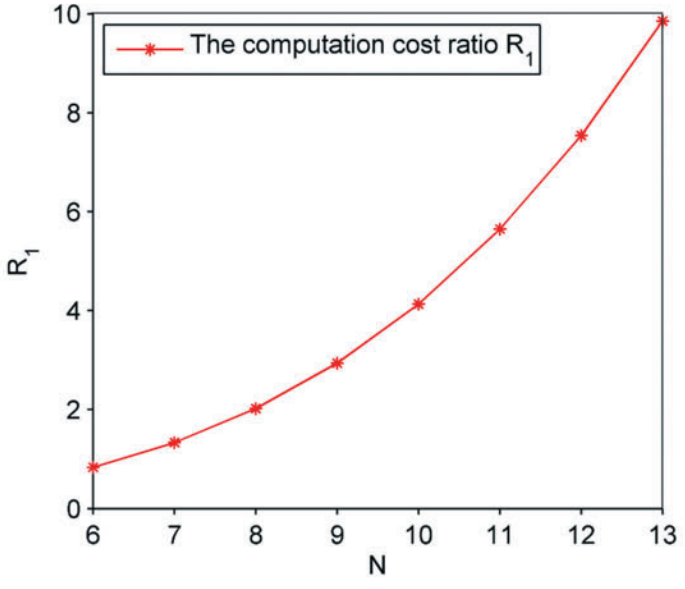

Fig. 1. The computational cost ratio curve R_1 between the standard diagonal averaging method and the uncoiled diagonal averaging method.

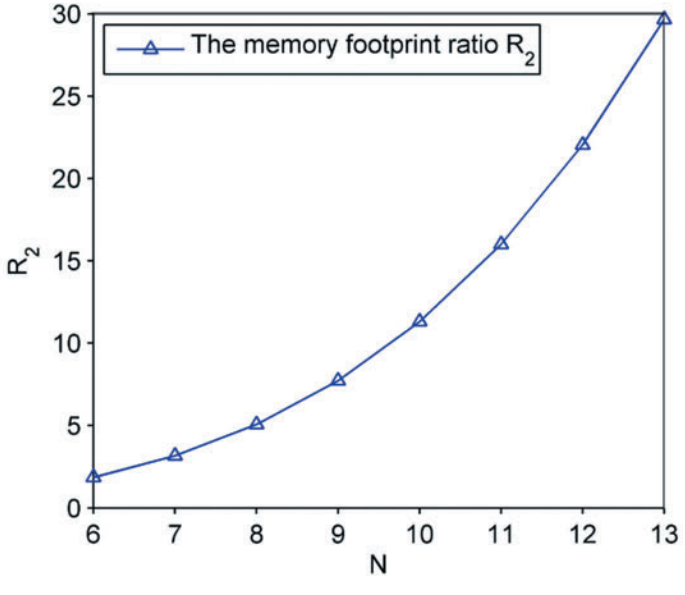

Fig. 2. The computer memory footprint ratio curve R_2 between the standard diagonal averaging method and the uncoiled diagonal averaging method.

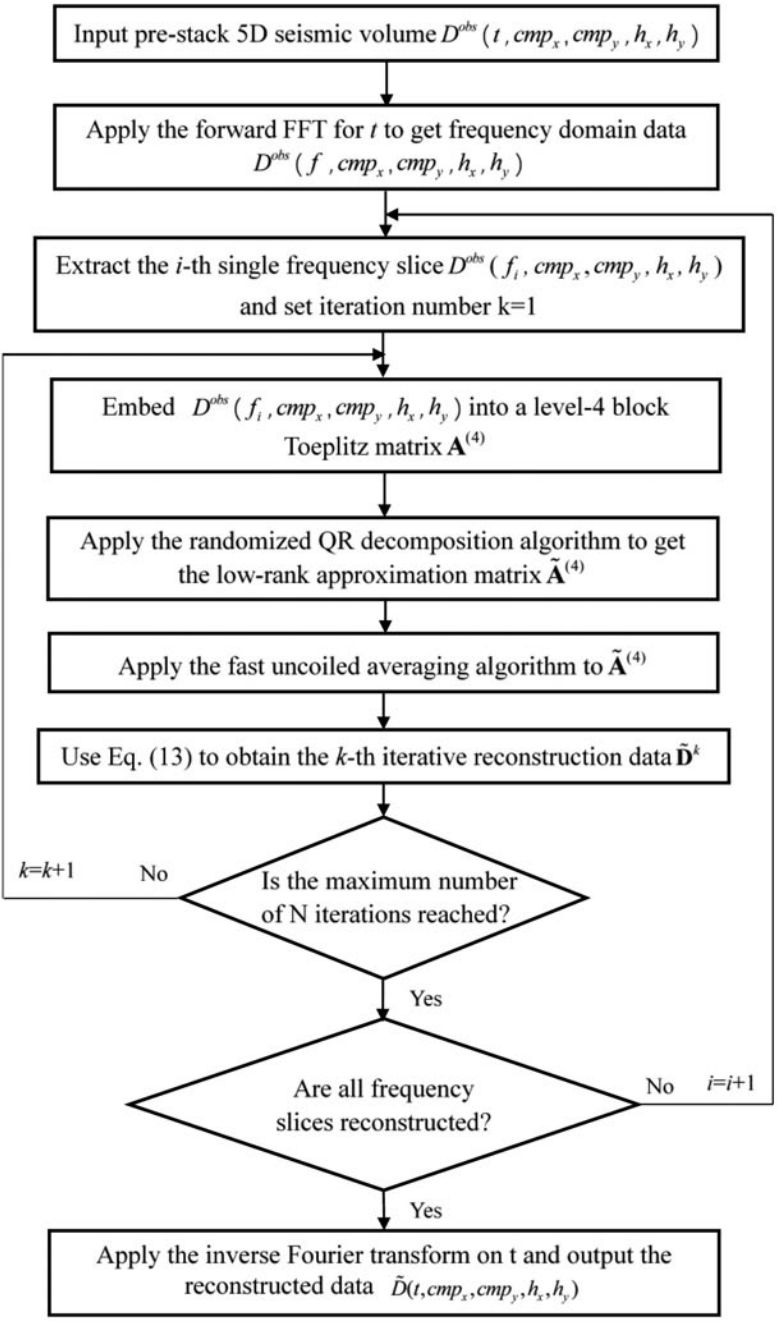

Fig. 3. The iterative reconstruction flow chart of the URQR method in the $f-cmp_x-cmp_y-h_x-h_y$ domain.

5D seismic data reconstruction via POCS method

For the observed data $\mathbf{D}^{obs}$ with irregular missing samples in the frequency domain, we utilize the Project Onto Convex Sets (POCS) iterative method described in Oropeza and Sacchi (2011) and Gao et al. (2015) to recover the missing samples. The k-th iterative reconstruction algorithm is expressed by

$$\tilde{\mathbf{D}}^{k+1} = (\mathbf{I} - \alpha\mathbf{S})P(\tilde{\mathbf{D}}^{k}) + \alpha\mathbf{D}^{obs},\ k = 0,1,\ldots,N_{iter}, \tag{17}$$

where $\mathbf{I}$ is a 4D array composed of ones, $\mathbf{S}$ is a 4D sampling operator with element 0 and 1 to identify the observed and decimated traces. The weight factor α is used to suppress the noise contained in the observed data during the reconstruction, α = (0,1]. The operator $P(\cdot)$ indicates the following operations, (1) constructing the 4D array $\tilde{\mathbf{c}}^{(4)}$ in eq. (9), (2) reducing the rank of $\mathbf{A}^{(4)}$ via randomized QR decomposition, and (3) applying the uncoiled diagonal averaging operation on $\tilde{\mathbf{A}}^{(4)}$ to get the recovered data $\tilde{\mathbf{D}}^{k}$ in eq. (13). $\mathbf{D}^{obs}$ denotes the input observed data, $\tilde{\mathbf{D}}^{0} = \mathbf{D}^{obs}$. In summary, the URQR reconstruction flow in the frequency-cmp_x-cmp_y-h_x-h_y domain is shown in Fig. 3.

EXAMPLES

Synthetic data experiments

The first example shows the reconstruction performance of the proposed algorithm in comparison to the typical implementation of rank reduction using the randomized SVD (RSVD) method and Lanczos bidiagonalization (Lanczos) method (Oropeza and Sacchi, 2011; Gao et al., 2013). We synthetize a series of 5D seismic data models that consist of $N\times N\times N\times N$ traces, N = 6, 7, . . . ,13 and 256 time samples per trace. Each data model includes three linear events and the signal to noise ratio is $S/N = 10^6$. We set the iteration number $N_{iter} = 10$ and the weight factor $\alpha = 1.0$. For each volume reconstruction case, we run the code 10 times for each method and record the computation time. Fig. 4 shows the averaging computational time comparison of these three methods. From Fig. 4, we see that the URQR algorithm is more efficient than the RSVD algorithm and the Lanczos algorithm, especially for the large N case.

The second example shows the reconstruction accuracy comparison of these three algorithms. We utilize the data model with size 10×10×10×10 traces and 256 time samples per trace, which is already synthesized in the first example to do the experiment. Fig. 5 shows the reconstruction quality for different percentages of decimated traces. The reconstruction quality was measured by a quality factor Q, Q being defined as

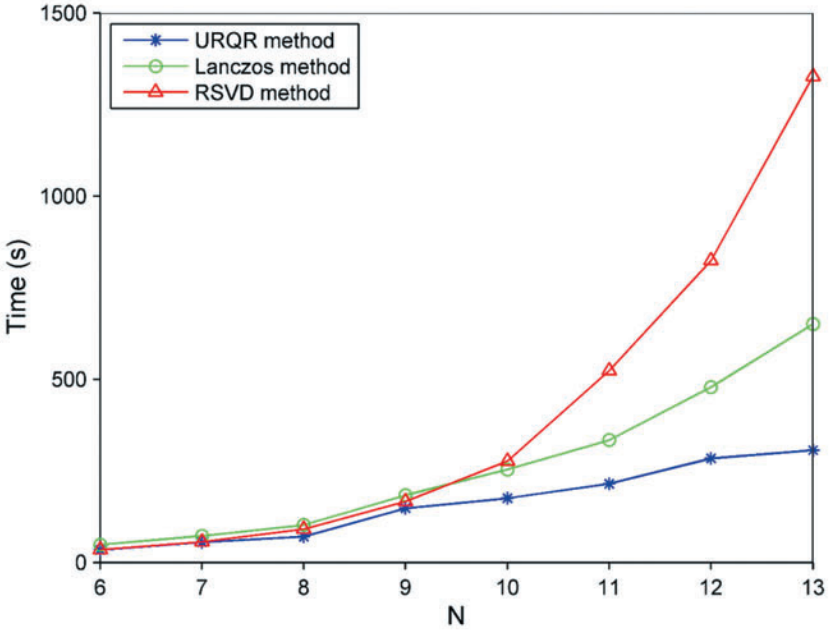

Fig. 4. The computational time comparison of the URQR method, RSVD method and Lanczos method.

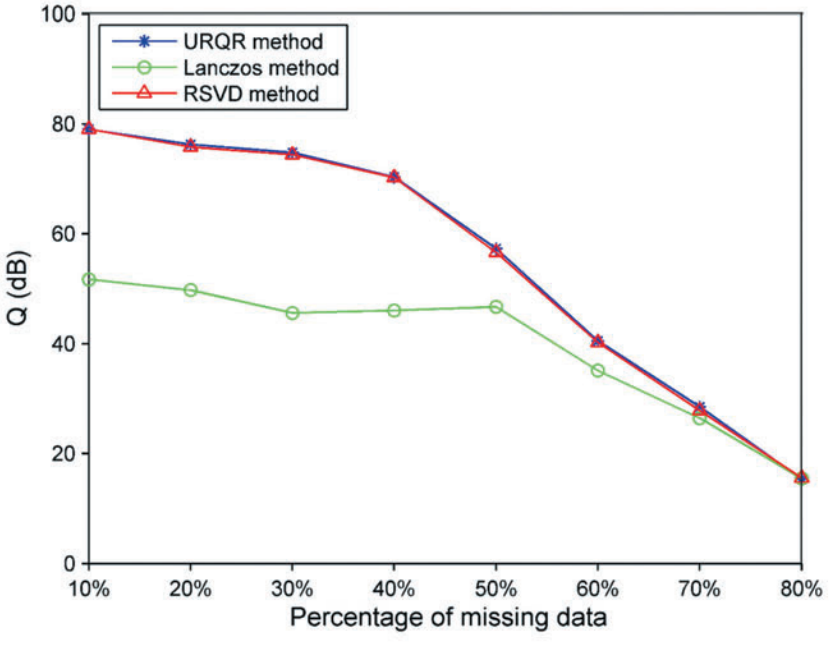

Fig. 5. The reconstruction quality comparison of the URQR method, RSVD method and Lanczos method.

$$Q = 10\log_{10}\frac{\left\|D^{true}\right\|_F^2}{\left\|D^{recon} - D^{true}\right\|_F^2}, \tag{18}$$

where D^{true} and D^{recon} represent the original fully sampled noise-free data and the reconstructed data, respectively. From Fig. 5 we see that the approximate rank reduction solution obtained from the URQR method has similar reconstruction quality with the RSVD method, but they both perform better than the Lanczos method.

In the next example, we exhibit the reconstruction result on the 4D spatial volume with 10×10×10×10 traces, which is the same data used in the second example. We randomly decimate 50% of seismic traces to form the incomplete data and then use the URQR method to do the reconstruction. Fig. 6 shows a slice view of CMP gathers with fixed $CMP_y = 2$, offset $h_y = 2$ and $CMP_x = 1,3,\ldots,9$. From the difference section in Fig. 6(d), we see that the energy of missing traces are well restored and all three linear events are recovered the natural continuity in spatial direction. We also randomly take out one single trace from the decimated data in Fig. 6(b) to show the reconstruction quality. Fig. 7 displays the reconstruction result of one single trace with fixed $CMP_x = 7$, $CMP_y = 2$, offset $h_x = 5$ and offset $h_y = 2$. From the red line, we can see that the amplitude of this missing trace is well recovered.

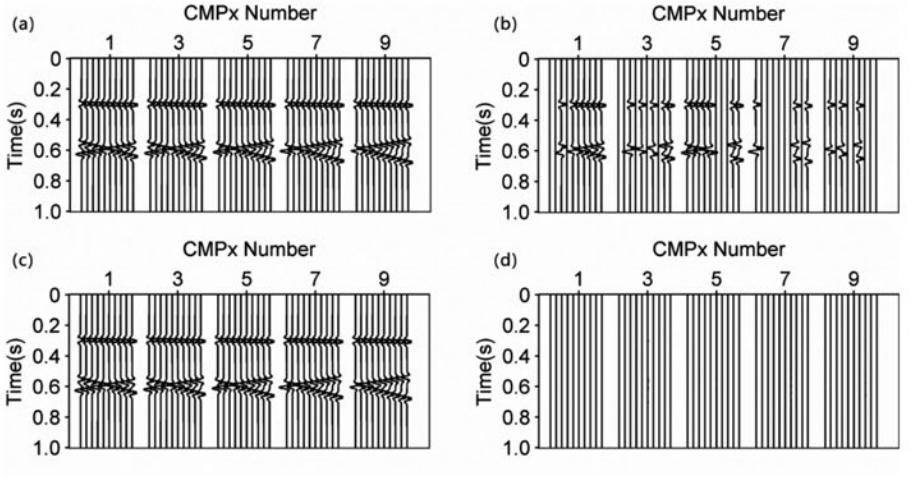

Fig. 6. The slice view of the CMP gathers for the noise-free data reconstruction. (a) Original data. (b) Decimated data with 50% of the traces missing. (c) Reconstructed data. (d) Difference between panel (a) and panel (c).

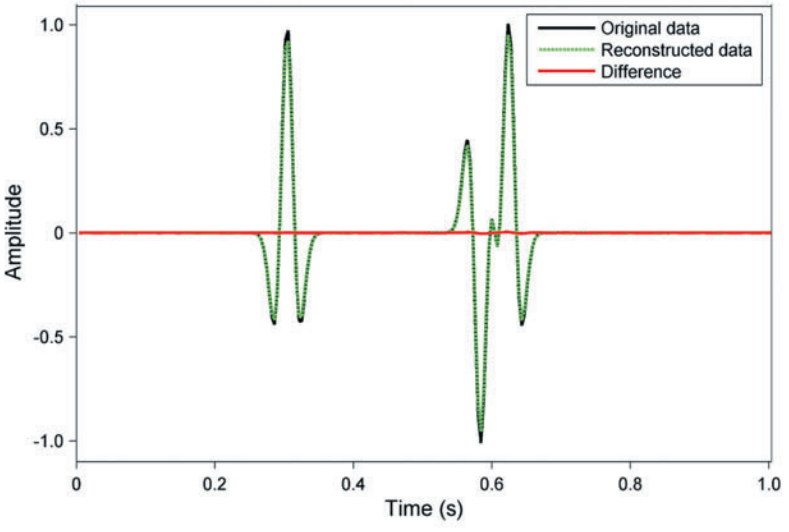

Fig. 7. The reconstruction result of one single trace. The coordinate of the trace is $CMP_x = 7$, $CMP_y = 2$, offset $h_x = 5$, offset $h_y = 2$.

Real data test

Based on the model data reconstruction results, we further test the performance of the proposed method on a real pre-stack 5D data. The data volume consists of 15×15 CMP bins and 13×13 offsets per bin. The CMP grid is 43×100 m and the offset grid is 567×50 m, besides, 18% of the points is covered with observations. We select a time window in the interval 900–1250 ms that corresponds to 351 samples. We set the iteration number $N_{iter} = 100$, the weight factor $\alpha = 0.8$, the rank $r = 10$ and the reconstruction frequency bandwidth as 1-80 Hz. Then, we reconstruct the data via the URQR method, RSVD method and Lanczos method. Fig. 8 displays five common offset gathers for fixed $CMP_y = 4$, $h_y = 6$, $CMP_x = 1, 2, \ldots, 15$ and $h_x = 5, 6, \ldots, 9$. Fig. 8(a) shows the original observed data prior to reconstruction. Figs. 8(b), 8(c), and 8(d) show the reconstruction results of these three methods. Fig. 9 shows five common middle point gathers for fixed $CMP_x = 6$, $h_x = 8$, $h_y = 1, 2, \ldots, 13$ and $CMP_y = 6, 7, \ldots, 10$. It can be seen that the missing traces are well recovered and the curved events are also well reconstructed in Fig. 8(b) and Fig. 9(b). From the red rectangular area, we see that the energy of the recovered events of the URQR method is similar as the RSVD method, but they both are stronger than the Lanczos methods under the same reconstruction conditions.

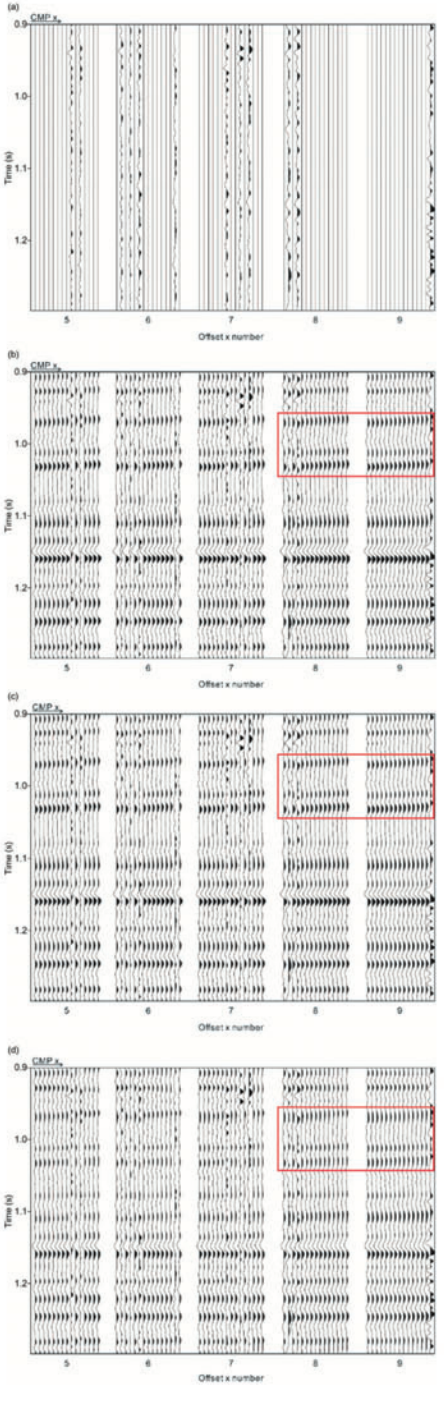

Fig. 8. The slice view of the common offset gathers for the field data reconstruction with fixed $CMP_y = 4$ and $h_y = 6$. (a) Input data. (b) URQR reconstructed result. (c) RSVD reconstructed result. (d) Lanczos reconstructed result.

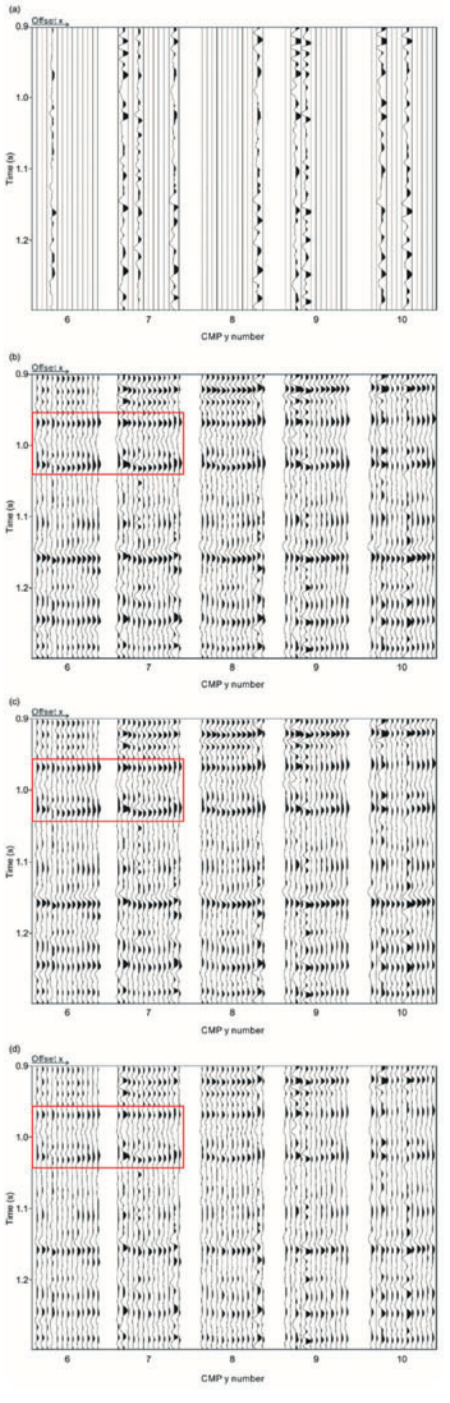

Fig. 9. The slice view of the CMP gathers for the real data reconstruction with fixed $CMP_x = 6$ and $h_x = 8$. (a) Input data. (b) URQR reconstructed result. (c) RSVD reconstructed result. (d) Lanczos reconstructed result.

CONCLUSIONS

In this paper, a new, fast rank-reduction algorithm is developed to reconstruct the pre-stack 5D irregular seismic volume. The presented method successfully introduces the compressive sensing theory into the matrix rank reduction processing to achieve the aim of compressing the size of the huge level-4 block Toeplitz matrix and improving the computation efficiency of the matrix rank reduction operation. Furthermore, the fast uncoiled diagonal averaging algorithm for the level-4 block Toeplitz matrix and the fast level-4 block Toeplitz matrix-vector multiplication algorithm via 4D FFT are also adopted to accelerate the matrix rank reduction processing. In addition, avoiding the constructing and storing of the huge rank-reduced level-4 block Toeplitz matrix at the diagonal averaging step could also save computer memory. In the data reconstruction experiments, the reconstruction performances of the RSVD method, Lanczos method and URQR method are compared. The reconstruction results show that the proposed URQR method has a larger reconstruction speed and better reconstruction quality compared with the other two methods.

The URQR fast rank reduction algorithm is a typical MSSA reconstruction method. The theoretic basis of MSSA reconstruction method is the predication filtering theory, hence, the URQR method is more suitable for reconstructing linear or quasi-linear events data. For real seismic data containing curved events or complex geological structures, people should divide the data into windows in time and spatial directions and then do the reconstruction in each small window. In addition, we want to point out that the present URQR method is just fit for reconstructing irregular missing traces in a grid with uniform spatial sampling. It cannot interpolate aliasing data with regular missing traces. The anti-aliasing interpolation of URQR method is still an open problem. Fortunately, Naghizadeh and Sacchi (2013) already offered some hints to achieve this.

ACKNOWLEDGMENT

We would like to thank the financial support from the National Key Research and Development Program of China (No. 2017YFC0307405), the Fundamental Research Fund for Central Universities (No. 2652017440), the National Natural Science Foundation of China (No. 41304102), the Foundation of State Key Laboratory of Petroleum Resources and Prospecting China University of Petroleum (Beijing) (No. PRP/open-1611).

REFERENCES

Cao, J. and Wang, B., 2015. An improved projection onto convex sets method for simultaneous interpolation and denoising. Chin. J. Geophys., 58: 2935-2947.

Chiron, L., Agthoven, M., Kieffer, B., Rolando, C. and Delsuc, M., 2014. Efficient denoising algorithms for large experimental datasets and their applications in Fourier transform ion cyclotron resonance mass spectrometry. Proc. Natl. Acad. Sci. U.S.A., 111: 1385-1390.

Cheng, J. and Sacchi, M., 2016. Fast and memory-efficient singular spectrum analysis for seismic data reconstruction and denoising. Expanded Abstr., 87th Ann. Internat. SEG Mtg., Houston: 4064-4068.

Duijndam, A., Schonewille, M. and Hindriks, C., 1999. Reconstruction of band-limited signals, irregularly sampled along one spatial direction. Geophysics, 64: 524-538.

Gan, S., Wang, S., Chen, Y., Zhang,Y. and Jin, Z., 2015. Dealiased seismic data interpolation using seislet transform with low-frequency constraint. IEEE Geosci. Remote Sens. Lett., 12: 2150-2154.

Gan, S., Chen, Y., Wang, S., Chen, X., Huang, W. and Chen, H., 2016. Compressive sensing for seismic data reconstruction using a fast projection onto convex sets algorithm based on the seislet transform. J. Appl. Geophys., 130: 194-208.

Gao, J., Chen, X., Liu, G. and Ma, J., 2010. Irregular seismic data reconstruction based on exponential threshold model of POCS method. Appl. Geophys., 7: 229-238.

Gao, J., Stanton, A., Naghizadeh, M., Sacchi, M. and Chen, X., 2013a. Convergence improvement and noise attenuation considerations for beyond alias projection onto convex sets reconstruction. Geophys. Prospect., 61: 138-151.

Gao, J., Sacchi, M. and Chen, X., 2013b. A fast reduced-rank interpolation method for prestack seismic volumes that depend on four spatial dimensions. Geophysics, 78: V21-V30.

Gao, J., Stanton, A. and Sacchi, M., 2015. Parallel matrix factorization algorithm and its application to 5D seismic reconstruction and denoising. Geophysics, 80: V173-V187.

Halko, N., Martinsson, P. and Tropp, J., 2011. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev., 53: 217-288.

Herrmann, J. and Hennenfent, G., 2008. Non-parametric seismic data recovery with curvelet frames. Geophys. J. Int., 173: 233-248.

Jia, Y., Yu, S., Liu, L. and Ma, J., 2016. A fast rank-reduction algorithm for three-dimensional seismic data interpolation. J. Appl. Geophys., 132: 137-145.

Kaplan, S., Naghizadeh, M. and Sacchi, M., 2010. Data reconstruction with shot-profile least-squares migration. Geophysics, 75: WB121-WB136.

Kutscha, H. and Verschuur, E. Data reconstruction via sparse double focal transform: an overview. IEEE Signal Proc. Mag., 29: 53-60.

Liu, B. and Sacchi, M., 2004. Minimum weighted norm interpolation of seismic records. Geophysics, 69: 1560-1568.

Liu, C., Li, P. and Liu, Y., 2013. Iterative data interpolation beyond aliasing using seislet transform. Chin. J. Geophys., 56: 1619-1627.

Liu, W., Cao, S., Gan, S., Chen, Y., Zu, S. and Jin, Z., 2016. One-step slope estimation for dealiased seismic data reconstruction via Iterative Seislet thresholding. IEEE Geosci. Remote Sens. Lett., 13: 1462-1466.

Naghizadeh, M. and Sacchi, M., 2007. Multi-step autoregressive reconstruction of seismic records. Geophysics, 72: V111-V118.

Naghizadeh, M. and Sacchi, M., 2009. f-x adaptive seismic trace interpolation. Geophysics, 74: V9-V16.

Naghizadeh, M. and Sacchi, M., 2010. Beyond alias hierarchical scale curvelet interpolation of regularly and irregularly sampled seismic data. Geophysics, 75: WB189-WB202.

Naghizadeh, M. and Sacchi, M., 2013. Multidimensional de-aliased Cadzow reconstruction of seismic records. Geophysics, 78: A1-A5.

Oropeza, V. and Sacchi, M., 2011. Simultaneous seismic data de-noising and reconstruction via multichannel singular spectrum analysis. Geophysics, 76: V25-V32.

Porsani, M., 1999. Seismic trace interpolation using half-step prediction filters. Geophysics, 64: 1461-1467.

Ronen, J., 1987. Wave-equation trace interpolation. Geophysics, 52: 973-984.

Spitz, S., 1991. Seismic trace interpolation in the F-X domain. Geophysics, 56: 785-794.

Tang, H. and Mao, W., 2014. Amplitude preserved seismic data reconstruction by 3D high-order parabolic Radon transform. Chin. J. Geophys., 57: 2918-2927.

Trad, D., Ulrych, T. and Sacchi, M., 2002. Accurate interpolation with high resolutions time variant radon transforms. Geophysics, 67: 644-656.
Trickett S., 2010. Rank-reduction-based trace interpolation. Expanded Abstr., 80th Ann. Internat. SEG Mtg., Denver: 1989-1992.
Wang, B., Wu, R. and Chen, X., 2015. Simultaneous seismic data interpolation and denoising with a new adaptive method based on dreamlet transform. Geophys. J. Int., 201: 1182-1194.
Wang, B., Wu, R. and Chen, X., 2014. Dreamlet-based interpolation using POCS method. J. Appl. Geophys., 109: 256-265.
Xu, S., Zhang, Y., Pham, D. and Lambare, G., 2005. Anti-leakage Fourier transform for seismic data regularization. Geophysics, 70: V87-V95.
Xue, Y., Ma, J. and Chen, X., 2014. High-order sparse Radon transform for AVO-preserving data reconstruction. Geophysics, 79: V13-V22.
Yang, P., Gao, J. and Chen, W., 2012. Curvelet-based POCS interpolation of non-uniformly sampled seismic records. J. Appl. Geophys, 79: 90-99.
Zhang, H., Chen, X. and Zhang, L., 2017. 3D simultaneous seismic data reconstruction and noise suppression based on the curvelet transform. Appl. Geophys., 14: 87-95.
Zwartjes, P. and Gisolf, A., 2007. Fourier reconstruction with sparse inversion. Geophys. Prosp., 55: 199-221.

APPENDIX A

THE DEFINITION OF AVERAGING OPERATOR **W**

$\mathbf{W}$ is a 4D averaging operator with the same size as $\tilde{\mathbf{D}}$ and $\mathbf{D}^{obs}$. The elements of $\mathbf{W}$ are written as,

$$\mathbf{W}(i_1,i_2,i_3,i_4)=\begin{cases}\dfrac{1}{i_4}\times\mathbf{W}_3(i_1,i_2,i_3), & 1\le i_4<K_4,\\ \dfrac{1}{K_4}\times\mathbf{W}_3(i_1,i_2,i_3), & K_4\le i_4<L_4,\\ \dfrac{1}{N_4-i_4+1}\times\mathbf{W}_3(i_1,i_2,i_3), & L_4\le i_4\le N_4,\\ \\ where, i_1=1,2,...,N_1, i_2=1,2,...,N_2,\\ \qquad i_3=1,2,...,N_3.\end{cases}, \tag{A-1}$$

here, $\mathbf{W}_3$ is a 3D array whose elements are shown as follows,

$$\mathbf{W}_3(i_1,i_2,i_3)=\begin{cases}\dfrac{1}{i_3}\times\mathbf{W}_2(i_1,i_2), & 1\le i_3<K_3,\\ \dfrac{1}{K_3}\times\mathbf{W}_2(i_1,i_2), & K_3\le i_3<L_3,\\ \dfrac{1}{N_3-i_3+1}\times\mathbf{W}_2(i_1,i_2), & L_3\le i_3\le N_3,\\ where, i_1=1,2,...,N_1, i_2=1,2,...,N_2,\end{cases}, \tag{A-2}$$

The elements of 3D array $\mathbf{W}_3$ are derived from the 2D array $\mathbf{W}_2$, $\mathbf{W}_2$ is defined as,

$$\mathbf{W}_2(i_1,i_2)=\begin{cases}\dfrac{1}{i_2}\times\mathbf{W}_1(i_1), & 1\le i_2<K_2,\\ \dfrac{1}{K_2}\times\mathbf{W}_1(i_1), & K_2\le i_2<L_2,\\ \dfrac{1}{N_2-i_2+1}\times\mathbf{W}_1(i_1), & L_2\le i_2\le N_2,\\ where, i_1=1,2,...,N_1\end{cases}, \tag{A-3}$$

here, $\mathbf{W}_1$ is a $1\times N_1$ vector, whose elements are given by,

$$\mathbf{W}_1(i_1)=\begin{cases}\dfrac{1}{i_1}, & 1\le i_1<K_1\\ \dfrac{1}{K_1}, & K_1\le i_1<L_1\\ \dfrac{1}{N_1-i_1+1}, & L_1\le i_1\le N_1\end{cases}. \tag{A-4}$$

APPENDIX B

THE DEFINITION OF $\mathbf{Q}_{p_4}^{(3)}$ AND $\mathbf{U}_{v_4}^{(3)}$

The element $\mathbf{Q}_{p_4}^{(3)}$ of the level-4 block Toeplitz matrix $\mathbf{Q}_{(k)}^{(4)}$ is a level-3 block Toeplitz matrix, $\mathbf{Q}_{p_4}^{(3)}$ can be written as,

$$\mathbf{Q}_{p_4}^{(3)}=\begin{bmatrix}\mathbf{Q}_{1,p_4}^{(2)} & 0 & 0 & \mathrm{L} & 0\\ \mathbf{Q}_{2,p_4}^{(2)} & \mathbf{Q}_{1,p_4}^{(2)} & 0 & \mathrm{L} & 0\\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M}\\ \mathbf{Q}_{p_3-1,p_4}^{(2)} & \mathbf{Q}_{p_3-2,p_4}^{(2)} & \mathbf{Q}_{p_3-3,p_4}^{(2)} & \mathrm{L} & 0\\ \mathbf{Q}_{p_3,p_4}^{(2)} & \mathbf{Q}_{p_3-1,p_4}^{(2)} & \mathbf{Q}_{p_3-2,p_4}^{(2)} & \mathrm{L} & \mathbf{Q}_{1,p_4}^{(2)}\\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{L} & \mathrm{M}\\ \mathbf{Q}_{L_3,p_4}^{(2)} & \mathbf{Q}_{L_3-1,p_4}^{(2)} & \mathbf{Q}_{L_3-2,p_4}^{(2)} & \mathrm{L} & \mathbf{Q}_{L_3-K_3+1,p_4}^{(2)}\\ 0 & \mathbf{Q}_{L_3,p_4}^{(2)} & \mathbf{Q}_{L_3-1,p_4}^{(2)} & \mathrm{L} & \mathbf{Q}_{L_3-K_3+2,p_4}^{(2)}\\ 0 & 0 & \mathbf{Q}_{L_3,p_4}^{(2)} & \mathrm{L} & \mathbf{Q}_{L_3-K_3+3,p_4}^{(2)}\\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M}\\ 0 & 0 & 0 & \mathrm{L} & \mathbf{Q}_{L_3,p_4}^{(2)}\end{bmatrix}, \tag{B-1}$$

here, $p_3 = 1,...,L_3, p_4 = 1,...,L_4$. The element $\mathbf{Q}^{(2)}_{p_3,p_4}$ of the level-3 block Toeplitz matrix $\mathbf{Q}^{(3)}_{p_4}$ is defined as,

$$\mathbf{Q}^{(2)}_{p_3,p_4} = \begin{bmatrix} \mathbf{Q}^{(1)}_{1,p_3,p_4} & 0 & 0 & \mathrm{L} & 0 \\ \mathbf{Q}^{(1)}_{2,p_3,p_4} & \mathbf{Q}^{(1)}_{1,p_3,p_4} & 0 & \mathrm{L} & 0 \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \mathbf{Q}^{(1)}_{p_2-1,p_3,p_4} & \mathbf{Q}^{(1)}_{p_2-2,p_3,p_4} & \mathbf{Q}^{(1)}_{p_2-3,p_3,p_4} & \mathrm{L} & 0 \\ \mathbf{Q}^{(1)}_{p_2,p_3,p_4} & \mathbf{Q}^{(1)}_{p_2-1,p_3,p_4} & \mathbf{Q}^{(1)}_{p_2-2,p_3,p_4} & \mathrm{L} & \mathbf{Q}^{(1)}_{1,p_3,p_4} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{L} & \mathrm{M} \\ \mathbf{Q}^{(1)}_{L_2,p_3,p_4} & \mathbf{Q}^{(1)}_{L_2-1,p_3,p_4} & \mathbf{Q}^{(1)}_{L_2-2,p_3,p_4} & \mathrm{L} & \mathbf{Q}^{(1)}_{L_2-K_2+1,p_3,p_4} \\ 0 & \mathbf{Q}^{(1)}_{L_2,p_3,p_4} & \mathbf{Q}^{(1)}_{L_2-1,p_3,p_4} & \mathrm{L} & \mathbf{Q}^{(1)}_{L_2-K_2+2,p_3,p_4} \\ 0 & 0 & \mathbf{Q}^{(1)}_{L_2,p_3,p_4} & \mathrm{L} & \mathbf{Q}^{(1)}_{L_2-K_2+3,p_3,p_4} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ 0 & 0 & 0 & \mathrm{L} & \mathbf{Q}^{(1)}_{L_2,p_3,p_4} \end{bmatrix}, \qquad \text{(B-2)}$$

here, $p_i = 1,...,L_i, i = 2,3,4$. The element $\mathbf{Q}^{(1)}_{p_2,p_3,p_4}$ of the level-2 block Toeplitz matrix $\mathbf{Q}^{(2)}_{p_3,p_4}$ is defined as,

$$\mathbf{Q}^{(1)}_{p_2,p_3,p_4} = \begin{bmatrix} \mathbf{Q}_{p+1,k} & 0 & 0 & \mathrm{L} & 0 \\ \mathbf{Q}_{p+2,k} & \mathbf{Q}_{p+1,k} & 0 & \mathrm{L} & 0 \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \mathbf{Q}_{p+p_1-1,k} & \mathbf{Q}_{p+p_1-2,k} & \mathbf{Q}_{p+p_1-3,k} & \mathrm{L} & 0 \\ \mathbf{Q}_{p+p_1,k} & \mathbf{Q}_{p+p_1-1,k} & \mathbf{Q}_{p+p_1-2,k} & \mathrm{L} & \mathbf{Q}_{p+1,k} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{L} & \mathrm{M} \\ \mathbf{Q}_{p+L_1,k} & \mathbf{Q}_{p+L_1-1,k} & \mathbf{Q}_{p+L_1-2,k} & \mathrm{L} & \mathbf{Q}_{p+L_1-K_1+1,k} \\ 0 & \mathbf{Q}_{p+L_1,k} & \mathbf{Q}_{p+L_1-1,k} & \mathrm{L} & \mathbf{Q}_{p+L_1-K_1+2,k} \\ 0 & 0 & \mathbf{Q}_{p+L_1,k} & \mathrm{L} & \mathbf{Q}_{p+L_1-K_1+3,k} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ 0 & 0 & 0 & \mathrm{L} & \mathbf{Q}_{p+L_1,k} \end{bmatrix}, \qquad \text{(B-3)}$$

here, $p_i = 1,...,L_i, i = 1,2,3,4,\ p = (p_4 - 1)L_3L_2L_1 + (p_3 - 1)L_2L_1 + (p_2 - 1)L_1$. The element $\mathbf{Q}_{p+p_1,k}$ is the $(p + p_1)$-th row and k-th column element of the orthogonal matrix $\mathbf{Q}$ in eq. (2).

$\mathbf{U}^{(3)}_{v_4}$ is a $K_3 \times 1$ column vector, it is defined as,

$$\mathbf{U}_{\nu_4}^{(3)} = \begin{bmatrix} \mathbf{U}_{K_3,\nu_4}^{(2)} \\ \mathbf{U}_{K_3-1,\nu_4}^{(2)} \\ \mathbf{M} \\ \mathbf{U}_{\nu_3,\nu_4}^{(2)} \\ \mathbf{M} \\ \mathbf{U}_{1,\nu_4}^{(2)} \end{bmatrix}, \tag{B-4}$$

here, $\nu_3 = 1,...,K_3$. The element $\mathbf{U}_{\nu_3,\nu_4}^{(2)}$ of $\mathbf{U}_{\nu_4}^{(3)}$ is still a $K_2 \times 1$ vector, it can be written as,

$$\mathbf{U}_{\nu_3,\nu_4}^{(2)} = \begin{bmatrix} \mathbf{U}_{K_2,\nu_3,\nu_4}^{(1)} \\ \mathbf{U}_{K_2-1,\nu_3,\nu_4}^{(1)} \\ \mathbf{M} \\ \mathbf{U}_{\nu_2,\nu_3,\nu_4}^{(1)} \\ \mathbf{M} \\ \mathbf{U}_{1,\nu_3,\nu_4}^{(1)} \end{bmatrix}, \tag{B-5}$$

where, $\nu_3 = 1,...,K_3, \nu_4 = 1,...,K_4$. The element $\mathbf{U}_{\nu_2,\nu_3,\nu_4}^{(1)}$ of the vector $\mathbf{U}_{\nu_3,\nu_4}^{(2)}$ is written as,

$$\mathbf{U}_{\nu_2,\nu_3,\nu_4,}^{(1)} = \begin{bmatrix} \mathbf{U}_{k,\nu+K_1} \\ \mathbf{U}_{k,\nu+K_1-1} \\ \mathbf{M} \\ \mathbf{U}_{k,\nu+\nu_1} \\ \mathbf{M} \\ \mathbf{U}_{k,\nu+1} \end{bmatrix}, \tag{B-6}$$

here, $\nu_i = 1,...,K_i, i = 1,2,3,4,$ $\nu = (K_4 - \nu_4)K_3K_2K_1 + (K_3 - \nu_3)K_2K_1 + (K_2 - \nu_2)K_1$, the element $\mathbf{U}_{k,\nu+\nu_1}$ in $\mathbf{U}_{\nu_2,\nu_3,\nu_4}^{(1)}$ is the k-th row and $(\nu+\nu_1)$-th column element of matrix $\mathbf{U}$ in eq. (12).