Computers & Geosciences 105 (2017) 21–33

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Fast hyperbolic Radon transform represented as convolutions in log-polar coordinates

Viktor V. Nikitin[a,*], Fredrik Andersson[a], Marcus Carlsson[a], Anton A. Duchkov[b]

[a] Centre for Mathematical Sciences, Lund University, Sölvegatan 18, Box 118, SE-22100 Lund, Sweden
[b] Institute of Petroleum Geology and Geophysics SB RAS, 3, Ac. Koptyuga ave., 630090 Novosibirsk, Russian Federation

ARTICLE INFO

Keywords:
Radon transforms
Multiples
Interpolation
FFT
GPU

ABSTRACT

The hyperbolic Radon transform is a commonly used tool in seismic processing, for instance in seismic velocity analysis, data interpolation and for multiple removal. A direct implementation by summation of traces with different moveouts is computationally expensive for large data sets. In this paper we present a new method for fast computation of the hyperbolic Radon transforms. It is based on using a log-polar sampling with which the main computational parts reduce to computing convolutions. This allows for fast implementations by means of FFT. In addition to the FFT operations, interpolation procedures are required for switching between coordinates in the time-offset; Radon; and log-polar domains. Graphical Processor Units (GPUs) are suitable to use as a computational platform for this purpose, due to the hardware supported interpolation routines as well as optimized routines for FFT. Performance tests show large speed-ups of the proposed algorithm. Hence, it is suitable to use in iterative methods, and we provide examples for data interpolation and multiple removal using this approach.

1. Introduction

In the processing of Common-Midpoint gathers (CMPs), the hyperbolic Radon transform has proven to be a valuable tool for instance in velocity analysis (Clayton and McMechan, 1981; Greenhalgh et al., 1990); aliasing and noise removal (Turner, 1990); trace interpolation (Averbuch et al., 2001; Yu et al., 2007); and attenuation of multiple reflections (Hampson, 1986). The hyperbolic Radon transform is defined as

$$\mathcal{R}_{\rm h}f(\tau, q) = \int_{-\infty}^{\infty} f(\sqrt{\tau^2 + q^2x^2}, x)dx, \tag{1}$$

where the function $f(t, x)$ usually corresponds to a CMP gather. Here, the parameter q characterizes an effective velocity value; and τ represents the intercept time at zero offset.

Several versions of Radon transforms are used in seismic processing, e.g., straight-line, parabolic, and hyperbolic Radon transforms. In many applications there is a need for a sparse representation of seismic data using hyperbolic wave events. One way to get sparse representations is by using iterative thresholding algorithms with sparsity constraints (Daubechies et al., 2004; Sacchi and Ulrych, 1995). Popular applications using such representations are seismic data interpolation and wavefield separation (Jiang et al., 2016; Trad, 2003). Since iterative schemes for computing such representations require the application of the forward and adjoint operators several times, it becomes important to use fast algorithms to limit the total computational cost.

Note that the direct summation over hyperbolas in (1) has a computational complexity of $O(N^3)$, given that the numbers of samples for the variables t, x, τ, q are $O(N)$. There are many effective ($O(N^2 \log N)$) methods for rapid evaluation of the traditional Radon transforms, or the parabolic Radon transform, see Beylkin (1984); Fessler and Sutton (2003); Schonewille and Duijndam (2001). The hyperbolic Radon transform is, however, more challenging. Nonetheless, a fast ($O(N^2 \log N)$) method for hyperbolic Radon transforms was recently presented in Hu et al. (2013). The method is based on using the fast butterfly algorithms described in O'Neil (2007) and Candès et al. (2009), and versions addressing computational efficiency are presented in Poulson et al. (2014), Li et al. (2015b, 2015a) and Li and Yang (2016).

A fast method for the standard Radon transform was proposed in Andersson (2005) by expressing the Radon transform and its adjoint in terms of convolutions in log-polar coordinates. To use this approach, it is necessary to resample data in log-polar coordinates, and this requires some interpolation method. An important property of such a scheme is that since the interpolation procedures are performed in the

[*] Corresponding author.
E-mail address: nikitin@maths.lth.se (V.V. Nikitin).

http://dx.doi.org/10.1016/j.cageo.2017.04.013
Received 22 April 2016; Received in revised form 18 April 2017; Accepted 26 April 2017
Available online 28 April 2017

Radon and image domains, the errors will be kept local. On the contrary, the interpolation errors in the frequency domain will create global non-systematic errors when switching back to the time-space domain. Since these non-localized errors contribute to the general structure of the function, they have to be mainly controlled.

Computationally efficient algorithms for GPUs were presented for the log-polar-based approach in Andersson et al. (2016). In this paper we propose to use the same approach and construct algorithms with complexity $O(N^2 \log N)$ for evaluation of the hyperbolic Radon transform. We present computational performance tests confirming the expected accuracy and the computational complexity, as well as predicted computational speed-ups for parallel implementations. Finally, we present several synthetic and real data tests using the hyperbolic Radon transform for data interpolation and multiple attenuation.

2. Method

To begin with, we note that functions $f(t, x)$ describing CMP gathers are symmetric with respect to x=0. Hence, by introducing

$$\widetilde{f}(s, y) = \frac{f(\sqrt{s}, \sqrt{y})}{2\sqrt{y}}, \tag{2}$$

it follows that

$$\mathcal{R}_{\mathrm{h}} f(\tau, q) = 2\int_0^\infty f(\sqrt{\tau^2 + q^2 x^2}, x)dx = 2\int_0^\infty \widetilde{f}(\tau^2 + q^2 y, y)dy. \tag{3}$$

The resulting expression in (3) has a form of the Radon transform over straight lines, and a fast algorithm for the evaluation of this was presented in Andersson et al. (2016), referred to as the log-polar Radon transform which is based on rewriting the key operations as convolutions in a log-polar coordinate system. In Section 2.1 we briefly recall the construction of the log-polar Radon transform and discuss how to adjust this method for optimal performance when processing seismic data, and in Section 2.2 we introduce coordinate transforms as well as sampling/interpolation requirements for accurate evaluation of $\mathcal{R}_{\mathrm{h}} f(\tau, q)$.

The fact that the hyperbolic Radon transform can be computed by a combination of a change of variables along with an application of the regular Radon transform implies that it can be evaluated using many different methods for computing the regular Radon transform, including methods with an hierarchical decomposition of the Radon transform (Basu and Bresler, 2000; George and Bresler, 2007), methods based on Fourier slice theorem (Beylkin, 1995; Fessler and Sutton, 2003), or chirp-Z transforms (Averbuch et al., 2006). In Andersson et al. (2016) a thorough comparison between fast implementations of Radon transform is presented, and where the log-polar implementation has an advantage with respect to computational performance in comparison to other approaches.

Another important aspect is the computational accuracy. Depending on how large approximation errors that are acceptable, different computational speedups can be achieved. For some applications rather low level of accuracy (i.e., errors of say 10%) could be acceptable. This could for instance be the case of event detection (but not removal). In this paper, we aim at keeping an accuracy level of a couple of digits, since this seems to be enough for most seismic applications. The accuracy level will be dependent on the interpolation error, and we show that this error can be reduced if higher order interpolation kernels are used.

Different algorithms also have different behavior in terms of how interpolation errors propagate. In this regards, the proposed method has an advantage, since the interpolation takes place in the measurement domain rather than in the frequency domain, which will keep errors more localized.

2.1. Log-polar Radon transform

The standard Radon transform (cf. (3)) can be written in terms of a double integral

$$\mathcal{R}\widetilde{f}(\tau^2, q^2) = \int\int \widetilde{f}(s, y)\delta(s - \tau^2 - q^2 y)dyds, \tag{4}$$

where δ denotes the Dirac distribution. In Andersson et al. (2016) one works with the log-polar coordinates

$$\begin{cases} s = e^{\rho'}\cos(\theta'), \\ y = e^{\rho'}\sin(\theta'), \end{cases} \quad \begin{cases} \tau^2 = \dfrac{e^{\rho}}{\cos(\theta)}, \\ q^2 = -\tan\theta. \end{cases} \tag{5}$$

By introducing $\zeta(\theta, \rho) = \delta(\cos(\theta) - e^{\rho})$, it turns out that the Radon transform can be efficiently evaluated using the *log-polar Radon transform*

$$\mathcal{R}_{\mathrm{lp}}\widetilde{f}(\theta, \rho) = \cos(\theta)\int\int \widetilde{f}(\theta', \rho')e^{\rho'}\zeta(\theta - \theta', \rho - \rho')d\rho' d\theta'$$

where, by abuse of notation, we use the same notation $\widetilde{f}$ for both coordinate representations.

However, the above representation is not suitable for treating functions $\widetilde{f}$ with support near the origin, since the origin is represented by $\rho = -\infty$. A way around this obstacle is to make a translation so that the support of $\widetilde{f}$ is moved away from the origin, and then work with functions supported within a subset of a unit circle-sector of opening angle β as in Fig. 2, right. In this figure the procedures of scaling and rotations are also applied in order to make a proper fit and connection to the interval for variable q from (4). The scale parameter, the rotation angle, as well the opening angle β will be defined in what follows. With this setup, the function $\widetilde{f}$ in log-polar coordinates and its log-polar Radon transform $\mathcal{R}_{\mathrm{lp}}\widetilde{f}(\theta, \rho)$ both have compact support. In this work we consider a simplified version with one circle-sector, because the hyperbolic Radon transform is typically computed for some small interval in the slowness variable q, which is directly related to the angle β. In contrast, three circle-sectors were used in Andersson et al. (2016), see Fig. 3. This is because that paper was aimed at treating data arises from line integrals for all directions, and in this case it is necessary to split the directional interval into at least three parts. The log-polar Radon transform is computed for each of these parts. Computations for all three intervals were done by using a formula for symmetric intervals, cf. (Andersson et al., 2016, Formulas 2.5, 2.6), where preliminary rotation procedures were applied to process non-symmetric intervals, see transformations T_m and S_m from Andersson et al. (2016), Formulas 2.9, 2.12. The final formula for computing the Radon transform by using preliminary rotations and the log-polar Radon transform for the symmetric interval reads as (Andersson et al., 2016, Formula 2.13).

To make this work consistent with our previous paper (Andersson et al., 2016), we deal with a symmetric interval for variable θ, i.e. $\theta \in [-\beta/2, \beta/2]$. Note, that other non-symmetric intervals of size β can be considered as alternatives, but after rotation procedures (similar to transformations T_m and S_m mentioned above) they would give the same end result as using a symmetric interval to begin with. We will refer to the implementation of $\mathcal{R}_{\mathrm{lp}}$ for values $\theta \in [-\beta/2, \beta/2]$ as the algorithm of *partial* $\mathcal{R}_{\mathrm{lp}}$.

With this in mind, we now briefly explain how to make slight modifications to the above scheme, better suited for the processing of CMP gathers. A simplified synthetic example of a typical CMP gather is shown in Fig. 1. Note that the function continues outside the maximum limits given by x and t, leading to a truncation of (3), (which can be seen e.g. as the circular artifacts in Fig. 1). Also note that there is no data in the region above a line t=kx, i.e. high offset x and small time intercept t, so to decrease the amount of computations we may ignore this piece. In the coordinates (s,y) this triangle is again a triangle, but with equation $s = k^2 y$. We set $\gamma = \arctan k^2$. Thus, we are in practice only

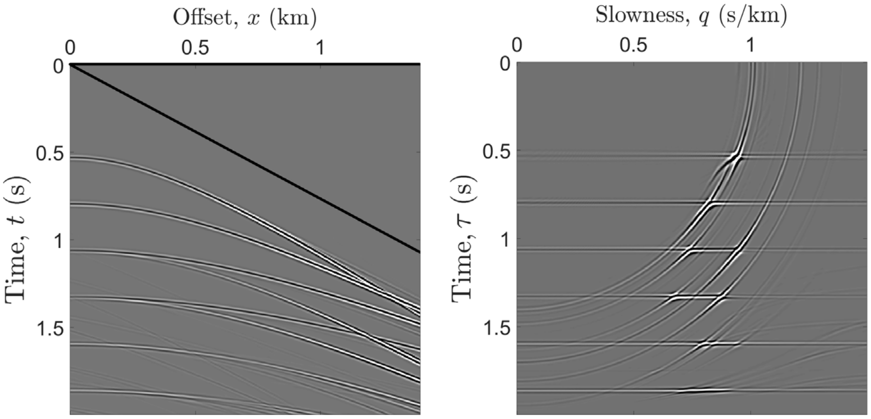


Fig. 1. CMP gather, cutting information for high offset and small time intercept (left). Hyperbolic Radon transform, the circles are artifacts from truncation (right).

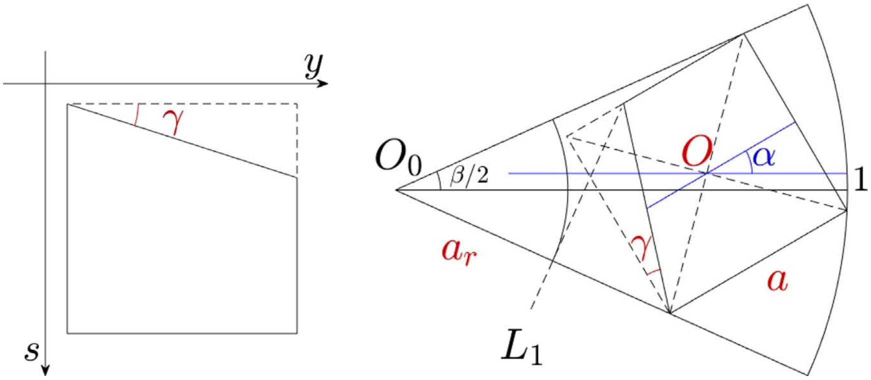


Fig. 2. Region of interest (trapezoid) for data evaluation (left). Scaling, rotation and translation for the log-polar setup (right).

interested in evaluating (4) for data $\widetilde{f}$ on a right trapezoid with the form illustrated in Fig. 2, left. Besides, one is usually also interested only in values of (τ, q) in a limited range $[\tau_{min}, \tau_{max}] \times [q_{min}, q_{max}]$.

In order for this to correspond to a symmetric interval of θ, we set $\beta = \arctan(q_{\max}^2) - \arctan(q_{\min}^2)$ and modify the relation between θ and q in (5) as follows

$$\theta = \alpha - \arctan(q^2),$$

where $\alpha = (\arctan(q_{\max}^2) + \arctan(q_{\min}^2))/2$. For a particular value of θ the output of the partial $\mathcal{R}_{\text{lp}}$ correspond to integrals over lines whose angle with respect to the vertical axis is θ. In order for these to correspond to desired values of q, one needs to rotate $\widetilde{f}$ so that the t axis makes an angle α with respect to the horizontal axis in Fig. 2, right. Moreover, due to the problems at the origin, $\widetilde{f}$ needs to be dilated and translated so that it fits within the circle sector of radius 1 and opening angle β, as in Fig. 2, right. This has the effect that the trapezoidal support is inscribed inside a square with side length a, located so that three of its corners lie on the border of the sector. It can be shown that

$$\begin{aligned} a &= \frac{\sin(\beta)}{\sqrt{\sin(2\alpha)\sin(\beta) + \cos(\beta)(\sin(2\alpha) + \sin(\beta)) + 1}}, \\ O &= O\left(\frac{a\sin\left(\alpha + \frac{\pi}{4}\right)\tan(\beta)}{\sqrt{2}}, \frac{a\cos\left(\alpha + \frac{\pi}{4}\right)\tan\left(\frac{\beta}{2}\right)}{\sqrt{2}}\right). \end{aligned}$$

The line L_1 passes through the fourth corner of the trapezoid and is orthogonal to the border of the sector; the distance from the origin O_0 to the line L_1 is indicated by a_r and indicates the first non-zero contribution to the partial $\mathcal{R}_{\text{lp}}$.

In summary, we are interested in the values of the log-polar Radon transform in the range $[-\frac{\beta}{2}, \frac{\beta}{2}] \times [\log(a_r), 0]$. With this setup the log-polar Radon transform can be computed in terms of the finite convolution

$$\begin{aligned} \mathcal{R}_{\text{lp}}\widetilde{f}(\theta, \rho) &= \cos(\theta)\int_{-\frac{\beta}{2}}^{\frac{\beta}{2}}\int_{\log(a_r)}^{0}\widetilde{f}(\theta', \rho')e^{\rho'}\zeta(\theta - \theta', \rho - \rho')d\rho' d\theta' \\ &= \cos(\theta)\mathcal{F}^{-1}(\mathcal{F}(\widetilde{f}(\theta, \rho)e^{\rho})(\widehat{\theta}, \widehat{\rho})\cdot\mathcal{F}\zeta(\widehat{\theta}, \widehat{\rho}))(\theta, \rho). \end{aligned} \quad (6)$$

Here, $\mathcal{F}$ denotes the two-dimensional Fourier transform. We use the notation $(\widehat{\theta}, \widehat{\rho})$ for the reciprocal variables of (θ, ρ). The function $\widehat{\zeta}(\widehat{\theta}, \widehat{\rho})$ can be accurately evaluated numerically (in a precomputing step) in contrast to $\zeta(\theta, \rho)$ which is defined in terms of distributions and is discontinuous along a curve, see (Andersson et al., 2016, Chapter 3) for a detailed description. To avoid wrapping effects, zero-padding is applied in the log-polar domain. The effects of the convolutions are schematically illustrated in Fig. 3. The trapezoid containing the support of the data is transformed to the shape indicated by the black points after a change to log-polar coordinates; the green lines show shifted versions of the function ζ; and the support after the log-polar Radon transform is applied is indicated by the thick black curves. By using this scheme we conclude that the rectangle $[-\beta, \beta] \times [\log(a_r), 0]$ is a good choice for enclosing the support of the functions, which is needed for the discrete evaluation of the integrals by means of convolutions in log-polar coordinates.

We now describe how $\mathcal{R}_{\text{lp}}$ can be used to recover $\mathcal{R}\widetilde{f}$ for a function $\widetilde{f}$ with support in the unit rectangle. The change of coordinates (s,y) for the log-polar setup is described by the transformation T,

$$\mathsf{T}\begin{pmatrix} t \\ x \end{pmatrix} = a\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}\begin{pmatrix} s - 0.5 \\ y - 0.5 \end{pmatrix} + \begin{pmatrix} O_1 \\ O_2 \end{pmatrix} \quad (7)$$

as well as the change of coordinates (τ, q) for the log-polar setup can be expressed by S, which can be found by scaling, rotation and translation procedures for the log-polar setup. Some tedious manipulations yield

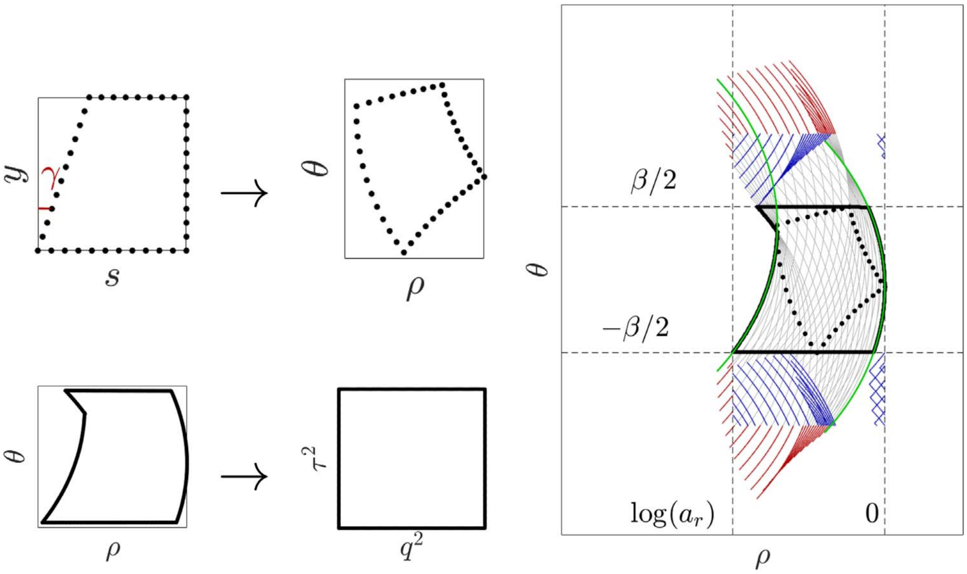


Fig. 3. Conversion to the log-polar domain and effects of computing convolutions. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

$$\mathsf{S}\begin{pmatrix}\tau^2\\ q^2\end{pmatrix} = \begin{pmatrix}\left(\left(a(\tau^2-\frac{1}{2})\cos(\alpha)+a\frac{\sin(\alpha)}{2}+O_1\right.\right.\\ \left.\left.+\left(a(\tau^2-\frac{1}{2})\sin(\alpha)-a\frac{\cos(\alpha)}{2}+O_2\right)\phi\right)\\ \phi\end{pmatrix}$$

with $\phi = \tan(\alpha - \arctan(q^2))$.

Moreover, we introduce two transformations for switching to log-polar coordinates according to relations (5):

$$\mathsf{P}_1\begin{pmatrix}t\\ x\end{pmatrix} = \begin{pmatrix}\log(\sqrt{t^2+x^2})\\ \arctan(\frac{x}{t})\end{pmatrix} \quad \mathsf{P}_2\begin{pmatrix}\tau^2\\ q^2\end{pmatrix} = \begin{pmatrix}\log(\tau^2\cos(-\arctan(q^2)))\\ -\arctan(q^2)\end{pmatrix} \tag{8}$$

To the end, by introducing linear operators

$$T\tilde{f} = \tilde{f}(\mathsf{T}^{-1}\mathsf{P}_1^{-1}\cdot) \quad Sg = g(\mathsf{S}^{-1}\mathsf{S}_2^{-1}\cdot)$$

the Radon transform over straight lines and its adjoint operator can be computed (up to a scaling factor) by

$$\begin{aligned}\mathcal{R}\tilde{f}(\tau, q) &= S^{-1}\mathcal{R}_{\rm lp}(T\tilde{f})(\tau, q),\\ \mathcal{R}^*g(t, x) &= T^{-1}\mathcal{R}^*_{\rm lp}(Sg)(t, x).\end{aligned}$$

2.2. Hyperbolic coordinates

Let f be a CMP gather measured on the rectangle

$$\{(t, x)\colon 0 \le t \le T,\ 0 \le x \le X\},$$

which we treat as a function on all of $\mathbb{R}^2$ which is 0 outside this rectangle. Note that

$$\begin{aligned}\mathcal{R}_{\rm h}(f)(\tau, q) &= XT\int_0^1\int_0^1 f(Tt, Xx)\delta(tT - \sqrt{\tau^2+q^2x^2X^2})dxdt\\ &= X\int_0^1\int_0^1 f(Tt, Xx)\delta\left(t - \sqrt{(\tau/T)^2+q^2x^2\frac{X^2}{T^2}}\right)dxdt\\ &= X\mathcal{R}_{\rm h}(f(T\cdot, X\cdot))\left(\frac{\tau}{T}, \frac{qX}{T}\right),\end{aligned} \tag{9}$$

which allows us to assume that f is given on the rectangle $[0, 1] \times [0, 1]$ to begin with. Upon corresponding rescaling of $\tau{:=}\frac{\tau}{T}$ and $q{:=}\frac{qX}{T}$, we are interested in evaluating $\mathcal{R}_{\rm h}f$ on the rectangle

$$\{(\tau, q)\colon \tau_{min} \le \tau \le 1,\ q_{\min} \le q \le q_{\max}\},$$

where τ_{min} corresponds to the arrival of the first event in the rescaled coordinates.

Now we recall the expression (6) representing the finite convolution for computing the log-polar Radon transform. It can be rapidly evaluated in terms of FFT if the log-polar samples (θ, ρ) are given on an equally spaced grid. Since data is assumed to be sampled in the (t,x) domain, a resampling is needed. We propose to do this using *cardinal B-spline interpolation* (De Boor, 1978; Unser, 1999), since this type of interpolation is particularly well suited for GPU implementations (cf. Ruijters et al., 2008). This technique is related to that used for fast unequally-spaced Fourier transforms (USFFT) (Beylkin, 1995; Dutt and Rokhlin, 1993), in the way that the interpolation is conducted by smearing data in one of the domains, and the compensating for that effect is done in the reciprocal domain.

In (6) we have to compute $\mathcal{F}(\tilde{f}(\theta, \rho)e^{\rho})(\hat{\theta}, \hat{\rho})$ which we can write as

$$\mathcal{F}(\tilde{f}(\theta, \rho)e^{\rho})(\hat{\theta}, \hat{\rho}) = \frac{\mathcal{F}((\tilde{f}e^{\rho})^*B_3)(\hat{\theta}, \hat{\rho})}{\mathcal{F}B_3(\hat{\theta}, \hat{\rho})}, \tag{10}$$

where B_3 is the cubic (cardinal) B-spline. Here we only consider frequencies $(\hat{\theta}, \hat{\rho})$ in a rectangle L, where $|\mathcal{F}B_3(\hat{\theta}, \hat{\rho})|$ does not become too small.

By using the coordinate transformations (2), (7) and (8) let

$$\begin{pmatrix}\varphi(t, x)\\ \eta(t, x)\end{pmatrix} = \mathsf{P}_1\mathsf{T}\begin{pmatrix}t^2\\ x^2\end{pmatrix}$$

be log-polar coordinates that correspond to the coordinates (t^2, x^2) in the time-offset domain. In these coordinates (10) takes the form

$$\mathcal{F}(f(\theta, \rho)e^{\rho})(\hat{\theta}, \hat{\rho}) = \frac{\mathcal{F}\left(\int\int\frac{f(t, x)}{2x}e^{\eta}\mathbf{J}(t, x)B_3(\theta-\varphi, \rho-\eta)dtdx\right)(\hat{\theta}, \hat{\rho})}{\mathcal{F}B_3(\hat{\theta}, \hat{\rho})} \tag{11}$$

where the division by $2x$ is related to the transformation (2). However, the Jacobian determinant $\mathbf{J}(t, x) = \left|\frac{\partial(\varphi, \eta)}{\partial(t, x)}\right|$ is a easily seen to consist of smooth bounded functions multiplied with $2x$ (coming from the derivative of x^2), which cancels out this seeming singularity at x=0. Subsequently the integrals and the Fourier transforms above can be well approximated using the trapezoidal rule and FFT for approximative evaluation of $\mathcal{F}$. If (t_j, x_k) are regular sampling points in the time-offset domain, we introduce the approximation to (11) by

$$g(\widehat{\theta}, \widehat{\rho}) = c\frac{\mathcal{F}\left(\sum_{j,k} \frac{f(t_j, x_k)}{2x_k} e^{\eta(t_j,x_k)} \mathbf{J}(t_j, x_k) B_3(\theta - \varphi(t_j, x_k), \rho - \eta(t_j, x_k))\right)(\widehat{\theta}, \widehat{\rho})}{\mathcal{F}B_3(\widehat{\theta}, \widehat{\rho})},$$

where c is a constant related to the sampling intervals. This approximation is then accurate for values of $(\widehat{\theta}, \widehat{\rho})$ in the rectangle L mentioned above.

As outlined in the previous section, this allows us to efficiently compute approximations of (6) on a regular lattice in the log-polar coordinate system (θ, ρ) via the formula

$$\mathcal{R}_{\mathrm{lp}}\widetilde{f}(\theta, \rho) \approx \cos(\theta)\mathcal{F}^{-1}(g(\widehat{\theta}, \widehat{\rho})\cdot\mathcal{F}\zeta(\widehat{\theta}, \widehat{\rho}))(\theta, \rho). \tag{12}$$

The final interpolation from the log-polar (θ, ρ) lattice to the Radon (τ, q) lattice can be done by using cubic B-splines and a slight modification of (12). Here, we again employ the transformations (7)–(8). In this case, let

$$\begin{pmatrix}\varphi(\tau, q)\\ \eta(\tau, q)\end{pmatrix} = \mathsf{P}_2\mathsf{S}\begin{pmatrix}\tau^2\\ q^2\end{pmatrix}$$

be the log-polar coordinates that correspond to the coordinates (τ^2, q^2) in the Radon domain. The interpolation from the log-polar (θ, ρ) lattice to the Radon (τ, q) lattice can then be done by using (a discrete version of)

$$\mathcal{R}_{\mathrm{h}}f(\tau, q) = \cos(\varphi)\int_{-\frac{\beta}{2}}^{\frac{\beta}{2}}\int_{\log(a_r)}^{0}\left(\mathcal{F}^{-1}\left(\frac{\chi_L g(\widehat{\theta}, \widehat{\rho})\mathcal{F}\zeta(\widehat{\theta}, \widehat{\rho})}{\mathcal{F}B_3(\widehat{\theta}, \widehat{\rho})}\right)(\theta, \rho)\right)B_3 \times(\varphi - \theta, \eta - \rho)d\rho d\theta, \tag{13}$$

where χ_L denotes the characteristic function of the set L.

Numerical evaluation of the approximations (11) and (13) appear to be well-suited for parallel computations, particularly on GPUs. For FFT we make use of the high-performance cuFFT library, efficient GPU kernels can be constructed for the smearing operations and for the vector multiplications. The discrete version of the operator $\mathcal{R}_{\mathrm{h}}$, as explained in the previous sections, will be denoted by R_{h}.

3. Reconstruction techniques

The adjoint operator for the hyperbolic Radon transform R_{h}^* is defined by using the inner product equality

$$\langle R_{\mathrm{h}}f, g\rangle = \langle f, R_{\mathrm{h}}^*g\rangle, \tag{14}$$

for arbitrary f and g. The operator is easy to construct by using the approach with switching to log-polar coordinates, essentially by reversing the order of the operations. With the adjoint operations at hand, one can consider iterative methods for representing f by sparse sums of hyperbolic wave events, and related interpolation and reconstruction techniques. A popular such method is based on the soft thresholding method for obtaining sparse representations proposed in Daubechies et al. (2004). In this setting it means to consider the minimization of

$$\| R_{\mathrm{h}}^*g - f \|_2^2 + \mu \| g \|_1, \tag{15}$$

for some choice of sparsity parameter μ.

By a simple modification of Theorem 3.1 in Daubechies et al. (2004), this minimization problem is solved by the iterations

$$g^n = \mathbf{S}_{c^2\mu}(g^{n-1} + c^2R_{\mathrm{h}}(f - R_{\mathrm{h}}^*g^{n-1})), \quad n = 1, 2, \ldots, \tag{16}$$

where g^0 is arbitrary, c is a positive constant such that $c \| R_{\mathrm{h}} \| < 1$, and $\mathbf{S}_\mu$ is a soft-thresholding function defined as

$$\mathbf{S}_\mu(v) = \begin{cases} v + \frac{\mu}{2}, & \text{if } v \le -\frac{\mu}{2},\\ 0, & \text{if } |v| < \frac{\mu}{2},\\ v - \frac{\mu}{2}, & \text{if } v \ge \frac{\mu}{2}.\end{cases}$$

To perform interpolation in the case of missing data, let S be a subset of the (t_j, x_k) grid where we do have measurements of f. We are then interested in minimizing

$$\sum_{(t_j,x_k)\in S}(R_{\mathrm{h}}^*g - f)^2(t_j, x_k) + \mu \| g \|_1,$$

which, defining f to be 0 where data is missing, is solved by the iteration

$$g^n = \mathbf{S}_{c^2\mu}(g^{n-1} + c^2R_{\mathrm{h}}(f - \chi_S R_{\mathrm{h}}^*g^{n-1})), \quad n = 1, 2, \ldots.$$

Here χ_S is the characteristic function of S. Again, this scheme is efficiently evaluated using the fast implementation of R_{h} explained in the previous section.

4. Discretization

In this section we derive guidelines for how to choose discretization parameters. For simplicity, we assume to work with regular sampling in the time-offset, and in the Radon domain; but the log-polar-based method can be easily generalized for unequally-spaced grids in these two domains.

In order to apply FFTs, samples in log-polar coordinates $(\theta, \rho) \in [-\frac{\beta}{2}, \frac{\beta}{2}] \times [\log a_r, 0]$ must be chosen on an equally spaced grid. By using coordinate conversions for the log-polar setup given by

$$\begin{pmatrix}\varphi(t, x)\\ \eta(t, x)\end{pmatrix} = \mathsf{P}_1\mathsf{T}\begin{pmatrix}t^2\\ x^2\end{pmatrix}.$$

In order to maintain accurate interpolation, we choose the sample spacing in θ and ρ with respect to the largest distance between sample points in the φ and η variables, i.e.,

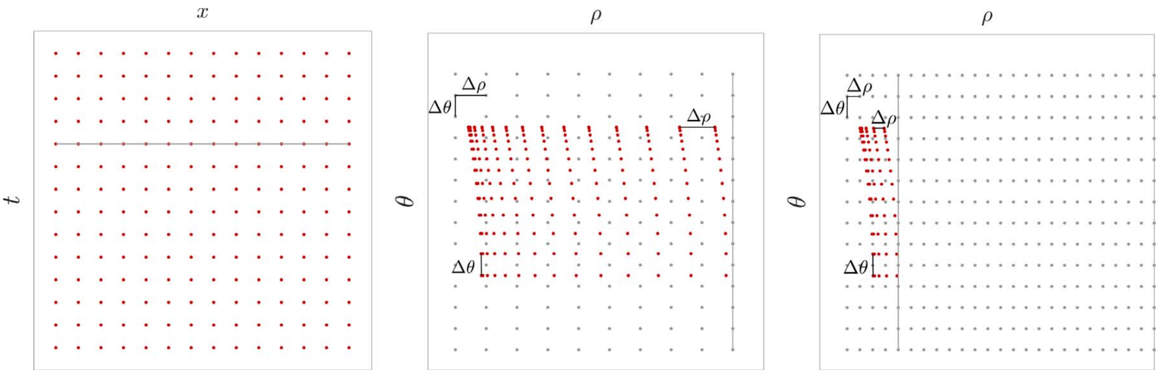
$$\begin{aligned}\Delta\theta &\ge \max_{t_j,x_k}(|\varphi(t_j, x_k) - \varphi(t_j + \Delta t, x_k)|, |\varphi(t_j, x_k) - \varphi(t_j, x_k + \Delta x)|),\\ \Delta\rho &\ge \max_{t_j,x_k}(|\eta(t_j, x_k) - \eta(t_j + \Delta t, x_k)|, |\eta(t_j, x_k) - \eta(t_j, x_k + \Delta x)|).\end{aligned} \tag{17}$$

This choice will determine the log-polar frequency range that can be covered, which in turn determines the resolution in the (τ, q) (Radon) domain. The quadratic behavior in the time sampling can be fairly well described in terms of the log-polar sampling, as long as time range is not too large. In the case of large time ranges, it can be beneficial to split the time-offset and Radon domains in parts and consider the log-polar Radon transform for each of these parts, in order to avoid too large differences in sample densities.

For instance, for small values of τ the grid for the Radon domain becomes more dense and samples $\Delta\theta, \Delta\rho$ should be chosen to be smaller. Suppose that we have already rescaled f according to (9), and note that the function is 0 until the arrival of the first event at τ_{min}. We may then split the integral in the following way

$$\begin{aligned}\mathcal{R}_{\mathrm{h}}f(\tau, q) &= \int_{\tau_{min}}^1\int_0^1 f(t, x)\delta(t - \sqrt{\tau^2 + q^2x^2})dxdt\\ &= \int_{\tau_{min}}^a\int_0^1 f(t, x)\delta(t - \sqrt{\tau^2 + q^2x^2})dxdt\\ &\quad + \int_a^1\int_0^1 f(t, x)\delta(t - \sqrt{\tau^2 + q^2x^2})dxdt\end{aligned} \tag{18}$$

for some a between τ_{min} and 1. For numerical evaluation of the first integral by using the log-polar based method samples in θ, ρ determined according to (17) become more dense, see Fig. 4 for a schematic description. The red dots in Fig. 4b indicate log-polar samples after conversion to discrete coordinates in the (t,x) domain illustrated in Fig. 4a. Equally spaced samples in the log-polar domain (gray dots) are chosen with respect to maximal distances between points (17). Fig. 4c demonstrates samples in the log-polar domain corresponding to small values of t (located above the gray line in Fig. 4a). The splitting procedure is not computationally intensive and can be applied several times to achieve accuracy for small values of τ.



(a) Grid (t, x) with splitting in t coordinate

(b) Grid (t, x) in log-polar coordinates

(c) Part in log-polar coordinates for small values of t

Fig. 4. Grids for conversion between time-offset and log-polar coordinates. Splitting in t variable, (a) Grid (t,x) with splitting in t coordinate, (b) Grid (t,x) in log-polar coordinates, (c) Part in log-polar coordinates for small values of t.

5. Accuracy tests

The proposed method uses cubic B-splines for data interpolation and therefore does not have provable error estimates. Here we aim at seismic applications in which case the data are band-limited (in particular, band-pass filtering is applied at the preprocessing step) and a limited accuracy level of several digits is sufficient in most applications. Thus in this section we study accuracy levels when processing synthetic and real CMP gathers which have different frequency content.

To begin with, we provide some numerical examples illustrating the error (versus a slow direct sum) of the log-polar-based method and its dependence on the frequency content of the data, the sampling rate, and the number of splittings applied in accordance with formula (18). For the sake of quality comparisons, we perform similar tests as the ones presented by Hu et al., 2013 for the fast hyperbolic Radon transform based on fast butterfly algorithms. The method of (Hu et al., 2013) is available in the open source software package Madagascar (Fomel et al., 2013). We analyze the dependence of accuracy on the following main factors:

- Sampling rates,
- Interpolation orders,
- Number of splittings by formula (18).

To study the sampling rates, we consider two synthetic CMP gathers generated using Ricker source wavelets of peak frequencies 10 and 40 Hz, respectively, see Figs. 5(a), (b). The corresponding frequency content is illustrated in Figs. 5(d), (e). Moreover, the hyperbolic Radon transform of the data sets is illustrated in Figs. 5(g), (h).

For the first test, a synthetic CMP gather is generated using a Ricker wavelet of 10 Hz, and sampled using N=2014. This data set is then resampled to sizes $N = 2^9, 2^{10}, 2^{11}, 2^{12}$. This test is thus made to illustrate the behavior of different sampling rates, and we should expect increased accuracy with increased values of N. For the second test, we generate a synthetic CMP gather for $(2^{12} \times 2^{12})$. Parts of this gather are then to conduct tests of smaller data sizes. Specifically, the part $\mathrm{f}_{1..N,1..N}$ is selected from the gather. In this test, the sampling density remains the same, and we would thus expect to see a similar accuracy level for the different values of N.

Up to this point we used cardinal cubic B-splines (B_3) for interpolation. To study the accuracy dependence on the interpolation order, we present tests for fourth-order (B_4) and fifth-order (B_5) B-splines. Moreover, we also present tests where either zero, one, or two splittings are used.

The CMP gathers are defined for $(t, x) \in [0, 4] \times [0, 5]$, and the hyperbolic Radon transform is computed for $(\tau, q) \in [0, 4] \times [0.1, 0.7]$. We check accuracy of the log-polar-based method as well as the method based on the butterfly algorithm. The fast butterfly algorithm has several parameters for controlling efficiency and accuracy, for details we refer to the pages 5, 6 in Hu et al. (2013). The parameter M (N in the paper) is of the order of the maximum value of the phase function $|\Phi(\mathbf{x}, \mathbf{k})|$ used for the approximation; and the parameters $q_{k_1}, q_{k_2}, q_{x_1}, q_{x_2}$ control the number of Chebyshev points. According to the results presented in Hu et al. (2013), the set of parameters $(q_i = 9, M = 64)$ shows an accuracy level of about $O(10^{-3})$ for typical images of size 1000×1000. We perform tests for $N_t = N_x = N_\tau = N_q = N$ where N was chosen as different powers of 2, and for obtaining an approximate accuracy level of $O(10^{-3})$ we used $(q_i = 9, M = N/16)$ in accordance with the tests conducted in Hu et al. (2013). As a reference method, we use a standard C implementation of the direct summation given by (3). Here cubic interpolation is used for interpolation in time. Normalized errors compared to direct summation over hyperbolas for the log-polar-based and for the fast butterfly algorithm are demonstrated in Fig. 6(a) and Fig. 6(b), respectively. The figures show that both methods have the same order of accuracy (normalized l^2 errors are inscribed into figures). The errors for the log-polar-based method are mostly observed in the region of small time intercept τ and high values of slowness parameter q. These accuracy problems can be reduced by additional splittings of the integral for the hyperbolic Radon transform, similar to the one suggested in the expression (18).

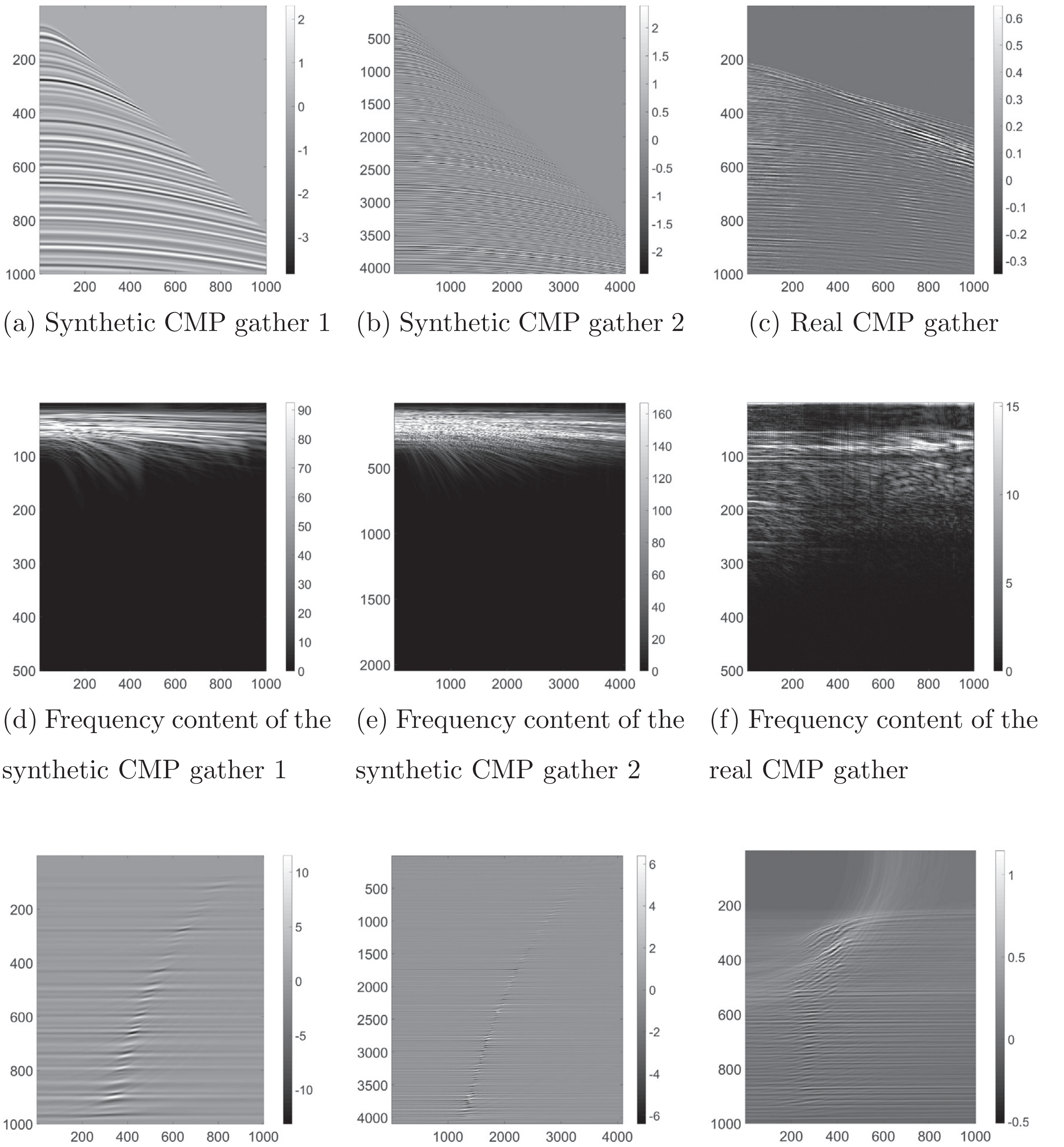
Test results are shown in Table 1 and 2. The first table shows relative accuracy levels for data sets constructed with making use of the wavelet of peak frequency 10 Hz. The table shows improvement of accuracy with increasing data sizes. It also confirms that higher order interpolators sufficiently improve accuracy and make it close to the accuracy of single precision arithmetics. Another conclusion from the table is that one splitting is enough to avoid large accuracy problems near t=0.

The second table in turn shows relative accuracy levels for data sets constructed by using parts of the CMP gather with sizes $(2^{12} \times 2^{12})$ which in turn was generated by the wavelet of peak frequency 40 Hz. As expected, the table shows the same order of accuracy with increasing data sizes, except the case N=512 where aliasing effects occur. Both tables show favorable accuracy of the log-polar-based method with B_3 spline interpolation.

We also check accuracy for the real CMP gather shown in Fig. 5(c). It has higher frequency components (Fig. 5(f)), and its hyperbolic Radon transform has more complicated structure, see Fig. 5(i). Table 3 shows the log-polar-based method has a favorable accuracy level.

6. Performance tests

In this section we will check the performance of the log-polar-based



(a) Synthetic CMP gather 1 (b) Synthetic CMP gather 2 (c) Real CMP gather

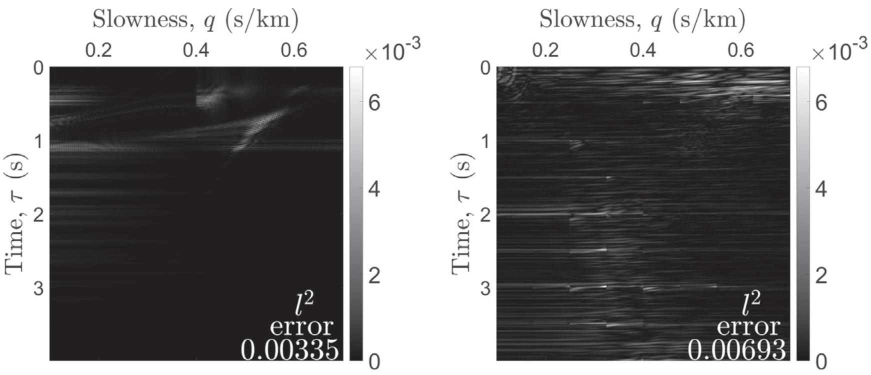
(d) Frequency content of the synthetic CMP gather 1 (e) Frequency content of the synthetic CMP gather 2 (f) Frequency content of the real CMP gather

(g) Hyperbolic Radon transform of the synthetic CMP gather 1 (h) Hyperbolic Radon transform of the synthetic CMP gather 2 (i) Hyperbolic Radon transform of the real CMP gather

Fig. 5. Data for accuracy tests.

method and compare it to the performance of the fast butterfly algorithm. The tables in the previous section show that the log-polar-based method using the cubic interpolation and one splitting in variables (according to (18)) behave similarly in terms of accuracy as the fast butterfly algorithm with q_i=9 Chebyshev points and $M/16$ as the maximum value of the phase function. Thus, we compare the computational times for these setups. For testing we used two implementations of the butterfly algorithm. First is the Madagascar implementation (Hu et al., 2013) which has convenient interface and works reliably for large 2D data sets. As pointed out during the review, there exist newer versions of the butterfly algorithm than the ones presented in Hu et al. (2013) and Poulson et al. (2014). Some drawbacks of this initial formulation were discussed in Li et al. (2015b). Better versions of the algorithm were proposed, i.e. the *butterfly factorization* in Li et al. (2015a), and the *interpolative butterfly factorization* in Li and Yang (2016). Matlab implementation of the latter algorithm was publicly available for testing. This implementation is substantially faster but unfortunately we were able to run tests for the hyperbolic Radon transform only for the data of sizes up to (512×512) because it gets quite memory demanding.

Table 4 demonstrates computational time results for the fast butterfly algorithm; for the CPU and the GPU versions of the log-polar-based method; and for the direct summation over hyperbolas. Direct summation over hyperbolas so as the CPU version of the log-polar-based method were implemented with Intel MKL and 8 OpenMP threads. The table confirms the complexity of the proposed methods



(a) Normalized error, Log-polar-based method

(b) Normalized error, Fast butterfly algorithm ($M = 64$, $q_i = 9$)

Fig. 6. Normalized error structure of computing the hyperbolic Radon transform for the log-polar-based method and for the fast butterfly algorithm.

Table 1
Relative errors of applying the hyperbolic Radon transform for different data sizes ($N \times N$) by using fast butterfly algorithm and the log-polar-based method for different setups. Test 1 - wavelet of peak frequency 10 Hz.

Method	N			
	512	1024	2048	4096
Fast butterfly	3.9e−02	7.1e−03	4.2e−04	7.0e−05
B_3 - no split	1.2e−01	2.3e−02	2.0e−03	8.1e−04
B_3 -1 split	8.8e−02	3.5e−03	3.6e−04	4.1e−05
B_3 -2 splits	6.8e−02	1.8e−03	2.6e−04	3.8e−05
B_4 - no split	1.3e−01	2.0e−02	2.1e−03	7.9e−04
B_4 - 1 split	4.7e−02	9.3e−04	8.6e−05	3.2e−05
B_4 - 2 splits	2.8e−02	5.4e−04	6.4e−05	2.4e−05
B_5 - no split	9.5e−02	2.1e−02	1.3e−03	7.9e−04
B_5 - 1 split	2.8e−02	4.6e−04	5.7e−05	2.1e−05
B_5 - 2 splits	1.8e−03	2.7e−04	5.1e−05	1.9e−05

Table 2
Relative errors of applying the hyperbolic Radon transform for different data sizes ($N \times N$) by using fast butterfly algorithm and log-polar-based method with different parameters. Test 2 - wavelet of peak frequency 40 Hz.

Method	N			
	512	1024	2048	4096
Fast butterfly	1.2e−02	8.3e−03	7.7e−03	7.3e−03
B_3 - no split	6.2e−02	3.2e−02	3.3e−02	2.8e−02
B_3 - 1 split	6.5e−03	5.3e−03	5.1e−03	5.4e−03
B_3 -2 splits	5.1e−03	4.4e−03	4.4e−03	5.0e−03
B_4 - no split	5.8e−02	3.5e−02	3.2e−02	3.6e−02
B_4 - 1 split	4.6e−03	1.3e−03	9.6e−04	9.3e−04
B_4 - 2 splits	4.4e−03	7.4e−04	7.3e−04	7.9e−04
B_5 - no split	4.5e−02	3.1e−02	3.3e−02	2.7e−02
B_5 - 1 split	3.4e−03	6.1e−04	7.2e−04	6.5e−04
B_5 - 2 splits	2.2e−03	4.8e−04	4.5e−04	4.4e−04

Table 3
Relative errors. Real CMP.

	No split	1 split	2 splits
B_3	5.7e−02	8.1e−03	5.4e−03
B_4	3.1e−02	6.6e−03	4.6e−03
B_5	2.3e−02	5.3e−03	4.4e−03
Fast butterfly: 1.2e−02			

and shows that a substantial performance gain is obtained by using GPUs. It is common in GPU computing that time to copy data between host and device memory constitute an essential part of the total computational costs (for our tests it takes ≈30% of total time). This time can be neglected in the case of using iterative schemes since it is then possible to keep all data in the GPU memory. For the tests performed, we have used a standard desktop with an Intel Core i7-3820 processor and NVIDIA GeForce GTX 970 video card with PCI Express x16 graphic interface. We also show the operator application time after the interpolative butterfly factorization (Li and Yang, 2016) but only for sizes (512 × 512) since the transform can not be evaluated for greater sizes due to memory demands.

All computations for the tests described above were performed in single precision. It is interesting to check GPU performance also for double precision. The processing power of GeForce GTX 970 in single precision is more than 30 times higher than the power for double precision, hence one can expect sufficient increase of total computational times in this case. According to Table 5 the algorithm slowdown for double precision in comparison to single precision is approximately equal to 4. This slowdown is explained by memory bandwidth limitations; see Kirk and Wen-Mei (2016), Ch. 5.1, 20.3 for details. FFT and interpolations constitute the most essential part of the total computational costs. These operations actively utilize GPU memory leading to a slowdown in arithmetics. Furthermore, time-consuming CPU-GPU data transfers for single precision are 2 times lower than the times for double precision. Tesla K20m is more adapted for computations in double precision than GeForce GTX 970 since processing power in single precision for K20m is only 3 times higher than for double precision. The obtained implementation of the log-polar-based method in double precision for this GPU is 2 times slower than the implementation in single precision (see Table 5).

In Fig. 7 we show the output of the adjoint operator for the hyperbolic Radon transform, as well as ℓ^1 regularized reconstruction given by (15). Since the adjoint transform is computed by reversing the order of operations (corresponding adjoint ones) for the forward transform, it does not arise problems with accuracy of the inner product test (14). The proposed algorithm passes the test with a relative error $O(10^{-5})$ for single precision, and with $O(10^{-12})$ for double precision.

Table 6 shows computational times using a GPU implementation of the proposed log-polar-based hyperbolic Radon transform, and 64 iterations of the iterative scheme (16). The table also contains times for single iteration of the forward and adjoint operators. Implementation of the adjoint operator is 10–15% slower than the implementation of the forward operator due to different GPU memory latency: number of cache hits for the forward operator is higher than for the adjoint operator. Performance indices of this kind depend on the GPU architecture. One can also see that in comparison to the GPU results in Table 4, the times for the forward operator, as well as for the adjoint operator, are lower due to a limited number of host-device data

Table 4
Computational time (in sec) for the hyperbolic Radon transform via direct summation over hyperbolas, fast butterfly algorithm and via the log-polar-based method (CPU and GPU), speed-up compared to the direct summation.

N	Direct sums CPU	Fast butterfly, Madagascar, CPU		Fast butterfly, int.fact., CPU[a]		Log-polar CPU		Log-polar GPU, GTX 970	
	Time	Time	Speed-up	Time	Speed-up	Time	Speed-up	Time	Speed-up
2^9	4.8e + 00	1.1e + 00	4.3	1.6e–01	30	3.3e–02	145.5	2.6e–03	1828.1
2^{10}	4.0e + 01	4.5e + 00	9.0	–	–	1.2e–01	344.4	9.6e–03	4220.2
2^{11}	3.2e + 02	1.8e + 01	17.8	–	–	4.7e–01	682.4	3.5e–02	9018.3
2^{12}	2.5e + 03	7.3e + 01	33.7	–	–	2.0e + 00	1257.7	1.4e–01	17,036.3

[a] Time of the MATLAB implementation of the fast butterfly algorithm via interpolative factorization; tests for sizes more than (512×512) are not presented due to memory problems.

Table 5
Computational time (in sec) of the log-polar-based method on different GPUs for single and double precision.

N	GeForce GTX 970			Tesla K20m		
	Single	Double	Slowdown	Single	Double	Slowdown
2^9	2.6e–03	1.0e–02	3.8	3.8e–03	7.6e–03	2.0
2^{10}	9.6e–03	3.8e–02	3.9	1.3e–02	2.7e–02	2.1
2^{11}	3.5e–02	1.4e–01	4.0	5.0e–02	9.9e–02	2.0
2^{12}	1.4e–01	5.7e–01	4.1	2.2e–01	4.2e–01	1.9

Table 6
Computational time (in s) for 64 soft-thresholding iterations.

N	Total time	Average time per iteration	
		Forward operator	Adjoint operator
2^9	8.1e + 00	1.9e–03	2.3e–03
2^{10}	3.1e + 01	7.4e–03	8.7e–03
2^{11}	1.2e + 02	2.9e–02	3.2e–02
2^{12}	4.6e + 02	1.1e–01	1.3e–01

transfers. Host-device data transfers take 20–30% of the total computational time; for instance, taking difference between times for the forward operator for $N = 2^{12}$ from Table 4 and Table 6 we have $0.14\ s - 0.11\ s = 0.03\ s$ or 27% which is mostly caused by data transfers.

7. Applications

In this section we mention some applications of the fast hyperbolic Radon transform. These are fairly standard, but the examples could be of practical interest due to the substantial computational speedup of the proposed implementation of the hyperbolic Radon transform.

7.1. Multiple attenuation

A well-known method for the attenuation of multiple reflections in CMP gathers is based on conducting the attenuation in a Radon domain. Here, multiples and primaries can be separated due to their differences in moveout. We have tested method described in Chapter 3 for the synthetic CMP gather in Fig. 8(a). Fig. 8(c) illustrates the Radon data, and note that the primaries and multiples are difficult to separate. The corresponding result after using 30 soft-thresholding iterations is illustrated in Fig. 8(d). The black line indicates the border between primaries and multiples, and Fig. 8(b) shows the reconstructed primaries.

7.2. Interpolation

Here we show some examples where we use soft-thresholding for conducting interpolation in cases of missing traces in the sampling setups. The CMP gather in Fig. 9(a) contains 50% randomly missing traces. Fig. 9(b) shows reconstruction results after 30 soft-thresholding iterations. To control the obtaining results, we also consider synthetic CMP gathers with 90% of missing traces, see Figs. 10a,c. In spite of the low amount of given data, it is still possible to reconstruct the structure of the waves (Fig. 10(b)). Moreover, varying the parameter of soft-thresholding (μ, see Chapter 3), one can improve the reconstruction quality. Here, the increase of the parameter μ leads to a better accuracy for low-amplitude events; conversely, high-amplitude events can be reconstructed with smaller values of μ. In Fig. 10(d) we show a result of the reconstruction with soft-thresholding iterations where the parameter μ was increased by 10 times compared to the one used for reconstructions in Figs. 9(b) and Fig. 10(b).

7.3. 2D field data

As an example of real data processing, we consider a CMP gather from the Canterbury data set (Lu et al., 2003), see Fig. 11(a). Attenuation of the multiples was carried out after applying the reconstruction method from Chapter 3 with 40 soft-thresholding iterations and the related muting procedure (Fig. 11(d)). The part of

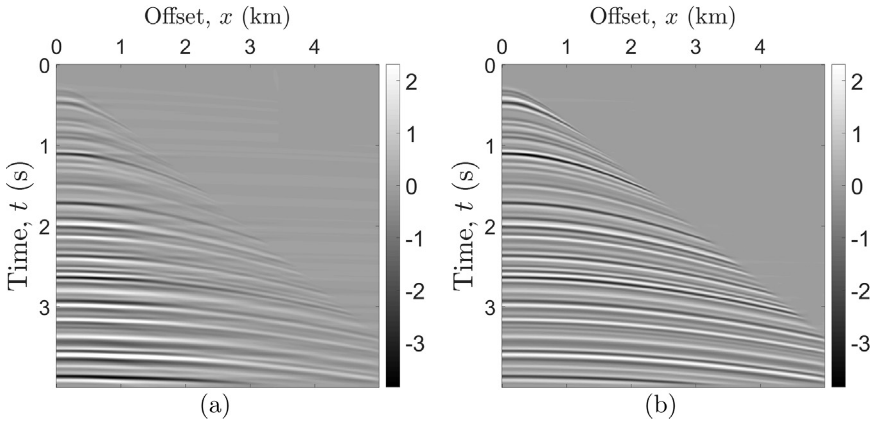
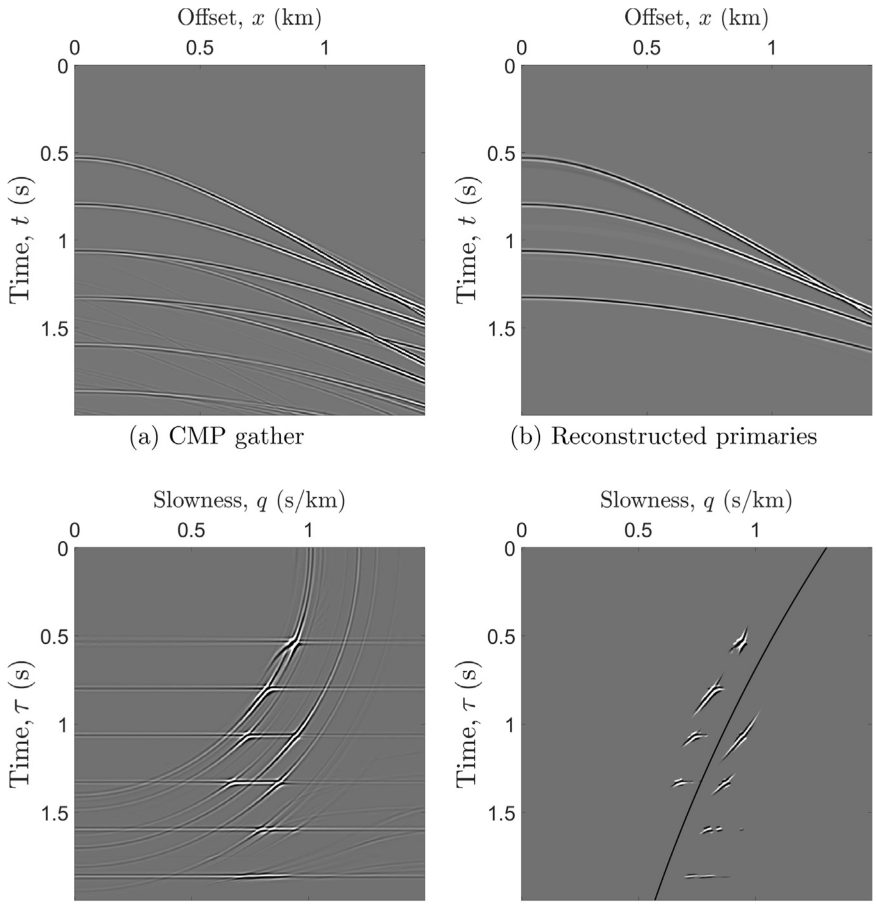


Fig. 7. Output of the adjoint operator for the hyperbolic RT (a), and the result from using 30 soft-thresholding iterations from (16) (b).



(a) CMP gather (b) Reconstructed primaries

(c) Hyperbolic Radon transform (d) Sparse representation

Fig. 8. Multiple attenuation with 30 soft-thresholding iterations.

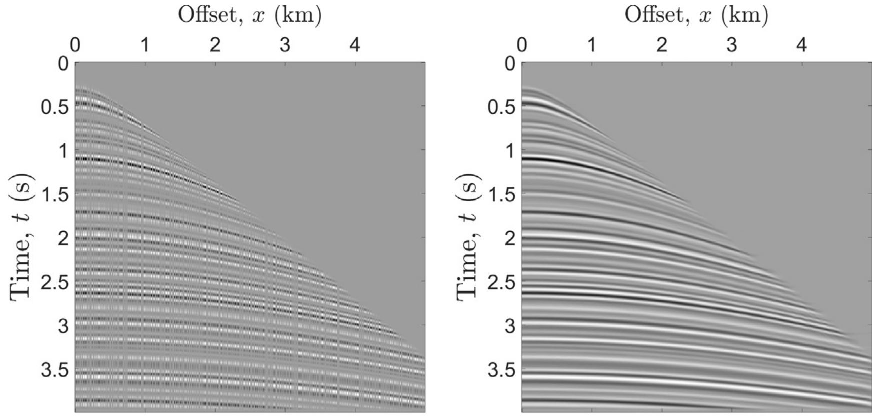
the Radon image corresponding to multiples was taken back to the time-offset domain (Fig. 11(c)) and subtracted from the initial CMP gather (Fig. 11(b)).

8. Discussion

8.1. Non-hyperbolic moveouts

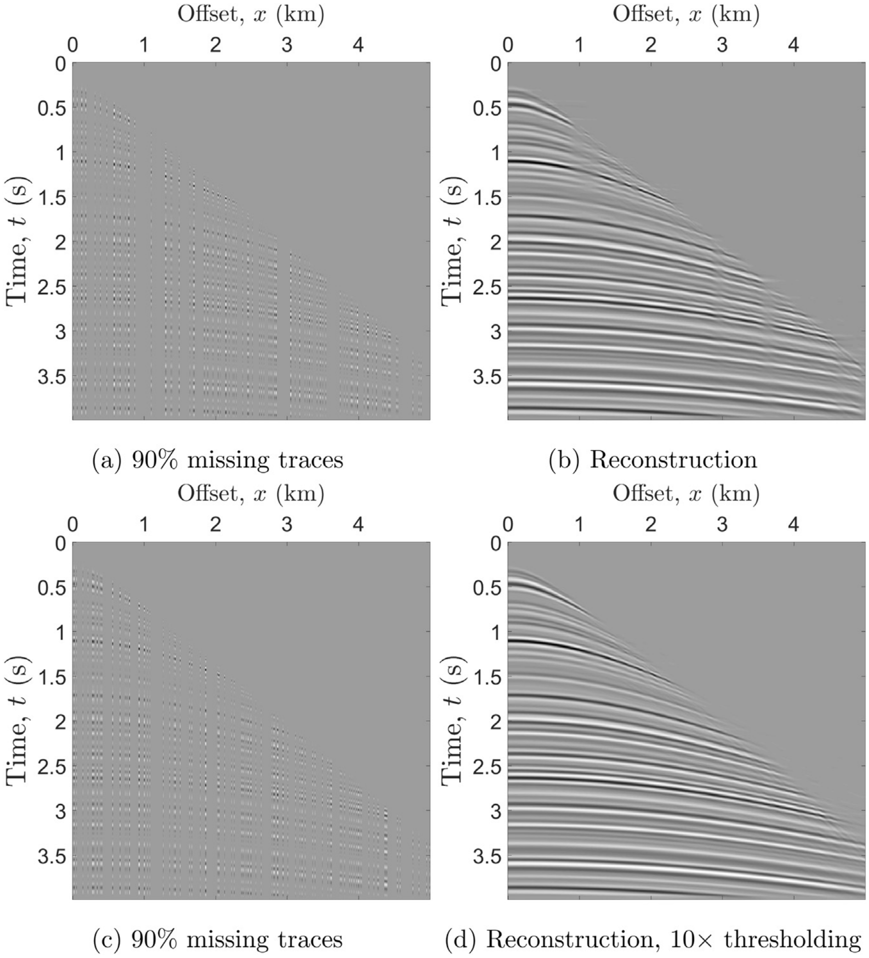
In this work we have dealt with the hyperbolic moveout $t(x) = \sqrt{\tau^2 + q^2x^2}$. The proposed method can be trivially adapted for the parabolic Radon transform where the moveout is given by $t(x) = \tau + qx^2$ (can be used for small offsets or after the normal moveout correction). Several other moveout types are used in seismic processing as listed in Fomel and Stovas (2010) . Let us mention a few examples of these non-hyperbolic moveouts: $t(x)^2 = q_1 + q_2x^2 + q_3x^4$ and $t(x)^2 = q_1 + q_2x^2 + q_3x^4/(1 + q_4x^2)$ (associated with seismic anisotropy of the model); $t(x) = q_1 + \sqrt{q_2 + q_3x^2}$ (associated with vertical heterogeneity of the model).

All of these moveouts describe monotonous dependence of time on the offset. Thus we can use a change of coordinates similar to formulas (2) and (3) in order to rewrite the corresponding Radon transform into the form appropriate for the log-polar change of variables. It should be



(a) 50% missing traces (b) Reconstruction

Fig. 9. Interpolation into missing traces with the soft-thresholding algorithm.



(a) 90% missing traces

(b) Reconstruction

(c) 90% missing traces

(d) Reconstruction, 10× thresholding

Fig. 10. Interpolation into missing traces with the soft-thresholding algorithm.

noted that the sampling rate should be chosen properly for each case.

8.2. 3D data processing

The proposed method can be extended for processing of three-dimensional data sets. The hyperbolas in the three-dimensional case are described by three parameters, in contrast to the two-dimensional case where only one parameter is needed. This means that if all three curvature parameters are to be present, the hyperbolic Radon transform would map three-dimensional data set onto four-dimensional data sets, and this is often not useful in practice. Therefore, it is common to consider cases where only two parameters are used. Let us begin by studying

$$\mathcal{R}_{\mathrm{h}}(\tau, q_1, q_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\sqrt{\tau^2 + q_1^2 x_1^2 + q_2 x_2^2}, x_1, x_2)dx_1 dx_2.$$

As an intermediate step, introduce

$$\mathcal{R}_s f(\tau, q_1, x_2) = \int_{-\infty}^{\infty} f(\sqrt{\tau^2 + q_1^2 x_1^2}, x_1, x_2)dx_1.$$

For each fixed x_2 the operator $\mathcal{R}_s$ can be rapidly evaluated by the log-polar-based method. With this operation, the 3D cube containing the variables (t, x_1, x_2) is transformed to a 3d cube described by the parameters (τ, q_1, y). Next, let us consider the hyperbolic Radon transform of the function $\mathcal{R}_s f$ with respect to variables τ, x_2. It follows that

$$\begin{aligned}
&\int_{-\infty}^{\infty} \mathcal{R}_s f(\sqrt{\tau^2 + q_2^2 x_2^2}, q_1, x_2)dx_2 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(\sqrt{(\sqrt{\tau^2 + q_2^2 x_2^2})^2 + q_1^2 x_1^2}, x_1, x_2\right)dx_1 dx_2 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\sqrt{\tau^2 + q_2 x_2^2 + q_1^2 x_1^2}, x_1, x_2)dx_1 \\
&= \mathcal{R}_{\mathrm{h}} f(\tau, q_1, q_2).
\end{aligned}$$

The computation of the three-dimensional transform can thus be made in two steps, where each step computes the two-dimensional hyperbolic Radon transform with regard to (t, x_1) and (τ, x_2) variables, respectively.

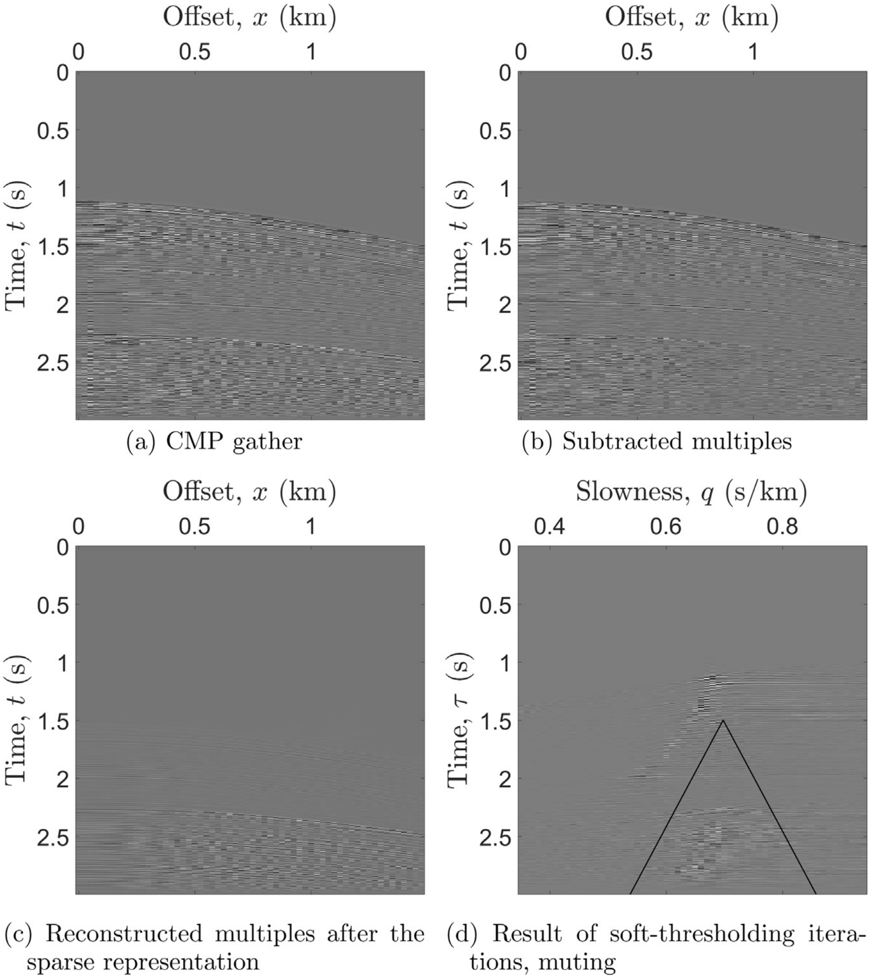
As an alternative, the hyperbolic Radon transform for a three-dimensional CMP gather after anisotropic moveout correction is also considered in Hu et al. (2015), Formula 7. It is given by

$$\mathcal{R}_{\mathrm{h}} f(\tau, p, q) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\sqrt{\tau^2 + p(x_1^2 - x_2^2) + 2q x_1 x_2}, x_1, x_2)dx_1 dx_2.$$

This problem can be addressed by making a change of coordinates between (x_1, x_2) and $(x_1^2 - x_2^2, x_1 x_2)$. In new variables the transform can be dealt with as a combination of two-dimensional hyperbolic Radon transforms.

8.3. Memory problems

Seismic data is characterized by large data sizes, therefore most of algorithm implementations on GPU need communication to transfer data between host and device memory. The current implementation is supposed to process a set of two-dimensional data slices simultaneously. Computation of the three-dimensional hyperbolic Radon



(a) CMP gather

(b) Subtracted multiples

(c) Reconstructed multiples after the sparse representation

(d) Result of soft-thresholding iterations, muting

Fig. 11. Multiple attenuation for 2D field data.

transform can be also performed as a combination of two-dimensional data slices, see Section 8.2. If the number of slices is too big, then the whole data set will not fit into the GPU memory and it is necessary to divide the set by parts and process each part independently. To optimize performance one can use CUDA Streams to overlap memory copy with computations.

Another question concerns the case if a two-dimensional slice does not fit into the GPU memory. Here we refer to the example of splitting (18) and recall that a similar procedure can be performed for other variables. Processing each term in the sum independently allows to decrease the amount of needed GPU memory and process large two-dimensional data.

9. Conclusions

A fast log-polar-based method for the evaluation of the hyperbolic Radon transform has been presented. According to the tests performed, the method demonstrates reasonable accuracy and favorable computational costs compared to other methods. The accuracy of the method can be increased when considering higher order interpolation kernels for coordinate conversions between time-offset, Radon, and log-polar domains. Numerical tests show that the GPU implementation is more than 10,000 faster for large data sets in comparison to a direct implementation in standard C of sums over hyperbolas, and a substantial speedup is also obtained compared to alternative fast methods. A discussion on how to use the method for processing three-dimensional data sets when computing the hyperbolic Radon transform for two slowness parameters is also presented.

Acknowledgements

This work was supported by the Crafoord Foundation (2014-0633) and the Swedish Research Council (2011-5589 and 2015-03780)

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2017.04.013.

References

Andersson, F., 2005. Fast inversion of the Radon transform using log-polar coordinates and partial back-projections. SIAM J. Appl. Math. 65 (3), 818–837.

Andersson, F., Carlsson, M., Nikitin, V.V., 2016. Fast algorithms and efficient GPU implementations for the Radon transform and the back-projection operator represented as convolution operators. SIAM J. Imaging Sci. 9 (2), 637–664.

Averbuch, A., Coifman, R., Donoho, D., Israeli, M., Walden, J., 2001. Fast slant stack: a notion of radon transform for data in a cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible. SIAM J. Sci. Comput..

Averbuch, A., Coifman, R.R., Donoho, D.L., Elad, M., Israeli, M., 2006. Fast and accurate polar Fourier transform. Appl. Comput. Harmon. Anal. 21 (2), 145–167.

Basu, S., Bresler, Y., 2000. $O(N^2 \log N)$ filtered backprojection reconstruction algorithm for tomography. IEEE Trans. Image Process. 9 (10), 1760–1773.

Beylkin, G., 1984. The inversion problem and applications of the generalized Radon transform. Commun. Pure Appl. Math. 37 (5), 579–599.

Beylkin, G., 1995. On the fast Fourier transform of functions with singularities. Appl. Comput. Harmon. Anal. 2 (4), 363–381.

Candès, E., Demanet, L., Ying, L., 2009. A fast butterfly algorithm for the computation of Fourier integral operators. Multiscale Model. Simul. 7 (4), 1727–1750.

Clayton, R.W., McMechan, G.A., 1981. Inversion of refraction data by wave field continuation. Geophysics 46 (6), 860–868.

Daubechies, I., Defrise, M., De Mol, C., 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. 57 (11), 1413–1457.

De Boor, C., 1978. A Practical Guide to Splines. Springer-Verlag, New York.

Dutt, A., Rokhlin, V., 1993. Fast Fourier transforms for nonequispaced data. SIAM J. Sci. Comput. 14 (6), 1368–1393.

Fessler, J.A., Sutton, B.P., 2003. Nonuniform fast fourier transforms using min-max interpolation. IEEE Trans. Signal Process. 51 (2), 560–574.

Fomel, S., Stovas, A., 2010. Generalized nonhyperbolic moveout approximation. Geophysics 75 (2), (U9–U18).

Fomel, S., Sava, P., Vlad, I., Liu, Y., Bashkardin, V., 2013. Madagascar: open-source software project for multidimensional data analysis and reproducible computational experiments. J. Open Res. Softw. 1 (1), e8.

George, A., Bresler, Y., 2007. Fast tomographic reconstruction via rotation-based hierarchical backprojection. SIAM J. Appl. Math. 68 (2), 574–597.

Greenhalgh, S., Mason, I., Mosher, C., Lucas, E., 1990. Seismic wavefield separation by multicomponent tau-p polarisation filtering. Tectonophysics 173 (1), 53–61.

Hampson, D., 1986. Inverse velocity stacking for multiple elimination. Can. J. Explor. Geophys. 22 (1), 44–55.

Hu, J., Fomel, S., Demanet, L., Ying, L., 2013. A fast butterfly algorithm for generalized Radon transforms. Geophysics 78 (4), (U41–U51).

Hu, J., Fomel, S., Ying, L., 2015. A fast algorithm for 3d azimuthally anisotropic velocity scan. Geophys. Prospect. 63 (2), 368–377.

Jiang, X., Zheng, F., Jia, H., Lin, J., Yang, H., 2016. Time-domain hyperbolic Radon transform for separation of PP and P-SV wavefields. Stud. Geophys. Et. Geod. 60, 91–111.

Kirk, D.B., Wen-Mei, W.H., 2016. Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, Boston.

Li, Y., Yang, H., Martin, E.R., Ho, K.L., Ying, L., 2015a. Butterfly factorization. Multiscale Model. Simul. 13 (2), 714–732.

Li, Y., Yang, H., Ying, L., 2015b. A multiscale butterfly algorithm for multidimensional fourier integral operators. Multiscale Model. Simul. 13 (2), 614–631.

Li, Y., Yang, H., 2016. Interpolative butterfly factorization. arXiv preprint arXiv:1605.03616.

Lu, H., Fulthorpe, C.S., Mann, P., 2003. Three-dimensional architecture of shelf-building sediment drifts in the offshore canterbury basin, New Zealand. Mar. Geol. 193 (1), 19–47.

O'Neil, M., Rokhlin, V., 2007. A New Class of Analysis-Based Fast Transforms. Tech. Rep. Yale University.

Poulson, J., Demanet, L., Maxwell, N., Ying, L., 2014. A parallel butterfly algorithm. SIAM J. Sci. Comput. 36 (1), C49–C65.

Ruijters, D., ter Haar Romeny, B.M., Suetens, P., 2008. Efficient GPU-based texture interpolation using uniform B-splines. J. Graph. GPU Game Tools 13 (4), 61–69.

Sacchi, M.D., Ulrych, T.J., 1995. High-resolution velocity gathers and offset space reconstruction. Geophysics 60 (4), 1169–1177.

Schonewille, M., Duijndam, A., 2001. Parabolic Radon transform, sampling and efficiency. Geophysics 66 (2), 667–678.

Trad, D., 2003. Interpolation and multiple attenuation with migration operators. Geophysics 68 (6), 2043–2054.

Turner, G., 1990. Aliasing in the tau-p transform and the removal of spatially aliased coherent noise. Geophysics 55 (11), 1496–1503.

Unser, M., 1999. Splines: a perfect fit for signal and image processing. IEEE Signal Process. Mag. 16 (6), 22–38.

Yu, Z., Ferguson, J., McMechan, G., Anno, P., 2007. Wavelet-Radon domain dealiasing and interpolation of seismic data. Geophysics 72 (2), V41–V49.