A Radon-Transform-Based Image Noise Filter—with Applications to Multibeam Bathymetry

Knut Landmark, Anne H. Schistad Solberg, Fritz Albregtsen, Andreas Austeng, and Roy Edgar Hansen

Abstract

This paper describes a linear-image-transform-based algorithm for reducing stripe noise, track line artifacts, and motion-induced errors in remote sensing data. Developed for multibeam bathymetry (MB), the method has also been used for removing scalloping in synthetic aperture radar images. The proposed image transform is the composition of an invertible edge detection operator and a fast discrete Radon transform (DRT) due to Götz, Druckmüller, and Brady. The inverse DRT is computed by using an iterative method and exploiting an approximate inverse algorithm due to Press. The edge operator is implemented by circular convolution with a Laplacian point spread function modified to render the operator invertible. In the transformed image, linear discontinuities appear as high-intensity spots which may be reset to zero. In MB data, a second noise signature is linked to motion-induced errors. A Chebyshev approximation of the original image is subtracted before applying the transform, and added back to the denoised image; this is necessary to avoid boundary effects. It is possible to process data faster and suppress motion-induced noise further by filtering images in nonoverlapping blocks using a matrix representation for the inverse DRT. Processed test images from several MB data sets had less noise and distortion compared to those obtained with standard low-pass filters. Denoising also improved the accuracy in statistical classification of geomorphological type by 10–28% for two sets of invariant terrain features.

Index Terms

Discrete transforms, image denoising, image restoration, iterative methods, remote sensing, sonar, terrain mapping.

DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

I. Introduction

Stripe noise, track line artifacts, antenna pattern effects or motion-induced errors degrade many types of remote sensing imagery. For example, satellite images from scanning mirror systems are often affected by periodic stripes in the across-track direction [1], [2]. The primary causes of these artifacts are offsets in detector response between the forward and reverse scans, unequal response of adjacent detectors, or random fluctuations in sensor response. Detector-to-detector variations also occur in pushbroom-type systems, where a linear array is oriented perpendicular to the flight direction and records one line of pixels simultaneously. With this configuration, stripe artefacts appear in the along-track direction and are not periodic [3]. Any kind of raster data obtained from a moving platform, where the pixels are recorded line by line, geolocated and then collated together, are prone to have similar errors to some degree, particularly if the image contains data from several adjacent survey track lines.

The multibeam echo sounder (MBES) provides a good example. By electronic beamforming of acoustical arrays, an MBES measures the echo travel time and seabed reflectivity in narrow beams fanning out perpendicular to the survey vessel's track [4, Ch. 8]. Subsequent data processing produces a high-resolution digital terrain model (DTM) of the seabed and a sidescan image mosaic of seabed scattering strength. MBES technology has had a significant impact on the marine sciences; in particular, it yields much information about seabed geological processes (erosion and deposition) by mapping morphological features such as sedimentary bedforms [5]. In addition to sidescan imagery, important tools in this respect are attribute images derived from the DTM, particularly gradient, curvature or shaded relief [6]. Not only are such images powerful interpretative tools, there is also promise that terrain attributes can be used in automated statistical classification of terrain type [7] and, ultimately, sediment type [8], [9].

The origin of this work was an attempt to use terrain attributes to improve the accuracy of acoustical classification of sediments, and the realization that, unfortunately, many attributes are sensitive to slight errors or artifacts in the data. One type of noise is caused by motion sensor errors, e.g., time delays in the heave, roll, and pitch (HRP) measurements. As the vessel moves through swell, even slight errors in HRP compensation will cause artificial wavy patterns ("ribbing") in the bathymetry. Often the patterns are near-periodical and oriented perpendicular to the vessel track. A detailed investigation of this phenomenon can be found in [10]. Another source of error is the mismatch between adjacent survey lines. A vessel typically moves back and forth in a parallel "lawnmower pattern", and in open seas a single pass may take several hours, during which time the sea level changes. Even if a good tidal model is available, a residual mismatch between survey lines may persist. Sound speed profiles (SSPs) are obtained from conductivity, temperature and depth (CTD) probe casts at extended intervals (unless a towed moving vessel profiler is used). Using an inaccurate SSP for computing sound refraction leads to depth estimation errors, and such errors are most pronounced at the outer beams where adjacent swathes overlap [5]. These types of mismatch cause stripes in attribute images. In addition, many sonar systems estimate the bottom echo by peak amplitude detection in the near-nadir beams, and by interferometric methods (phase matching) at higher incidence angles. At the transition angle between the two modes, a systematic depth offset may occur, which also causes stripe noise. Finally, beam calibration errors cause stripes in sidescan data [11], a phenomenom which is analogous to the one observed in pushbroom optical systems.

Many of the methods for destriping remote sensing images have been developed for

DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

satellite systems, and all are not directly applicable to multibeam bathymetry. The spatial filters developed in [12]–[14] are 1-D and intended for images with horizontal scan-line noise. The statistical methods of equalization [15], [16], histogram matching [17], [18], and moment matching [19] adjust each detector output to match the statistics of some reference distribution, provided each detector can be identified with a line of pixels (horizontal or vertical) in the image. To remove irregular stripes, i.e., lines where only a subset of the pixels are noise-contaminated, and nonlinear stripes where the degradation parameters change across the scene, Shen et al. [20] has proposed a method where lines are split into segments that are processed separately with a statistical destriping algorithm. It may be that these methods can be adapted to sidescan imagery (before georectification), but the artifacts to be considered here are not caused by intrinsic variations in detector response. Likewise, the imaging modes or assumptions are different from ours in the recent statistical and variational approaches reported in [1]–[3], [21], and [22].

A more general approach is to use an edge detection algorithm to find the pixels affected by stripe noise and inpainting techniques to correct them. Tsai and Chen [23] used the Canny algorithm [24] and a spline-based inpainting scheme applicable to thin horizontal or vertical stripes. Image filtering in the Fourier domain is another option when the noise is periodic or at least directive [25, Ch. 4]. This was explored in [26] for side-scan sonar, and in, for example, [27] for satellite imagery. Fourier domain filtering does not require a statistical model of the image formation process, and can be applied to both georectified and raw images, but it may cause unwanted smoothing or distortion of real features. The same is true of filtering in the wavelet domain. To minimize the smoothing effect in images with horizontal stripes, Chen et al. [27] isolated the noise frequency by averaging the 1-D power spectra from each image column, as the noise frequency tends to be constant across columns. However, the noise of primary interest in this paper has a broad spectrum.

This paper presents an alternative image-transform-based denoising filter which is motivated by the problems affecting MBES bathymetry and derived attributes. The method has also proven effective for reducing scalloping in synthetic aperture radar (SAR) images, but we concentrate here on MBES data. The forward image transform is an edge detection operator composed with a discrete Radon transform (DRT). Noise suppression takes place in the DRT domain. There are no assumptions about periodicity or orientation, and artifacts need not be parallel. A linear artifact is focused to an isolated high-intensity spot in the Radon domain. For MBES data, it is also conjectured that another noise signature is linked with motion-induced errors in the sounding data. The proposed algorithm is applied to two MBES datasets from the North Sea, and compared qualitatively with Fourier domain filters and spatial low-pass filters. To quantify the effect of the denoising algorithm, we have considered the machine learning problem of classifying different terrain morphologies, and how denoising affects the classification accuracy. For this problem we employ two sets of invariant terrain features based on the empirical distribution of local binary patterns (LBPs) and intrinsic geometric features.

A major part of this paper concerns the forward and inverse two-step image transform. One candidate for the edge operator is the Laplace operator (Laplacian), which is linear, approximately rotation invariant, edge-sensitive, and noise-sensitive. The Laplacian is also a measure of curvature, which is a recommended feature for visualizing terrain [6]. However, the Laplacian is not one-to-one, and we therefore suggest a modification to obtain an invertible edge operator. Basing the algorithm on discrete approximations of the classical forward

and inverse Radon transform formulae did not yield satisfactory results in our experiments. Filtered backprojection [28], [29], i.e., the standard fast method for inverting the linear Radon transform, causes a degree of smoothing. Discrete approximations may involve interpolation, and Beylkin [30] moreover has shown that, in a discrete setting, inversion of the Radon transform can only be approximate when it is based on a discretization of Radon's classical inversion formula. Here we consider instead the DRT of Götz and Druckmüller [31] and Brady [32] (GDB), and compute the inverse using linear algebra. In this formulation, the forward and inverse Radon transforms are discrete linear operators, where no interpolation of data is involved. Press [33] found a fast iterative inverse algorithm that can reconstruct images with arbitrary precision from the forward GDB transform data. However, a numerical analysis of the Press inverse suggests that the algorithm converges only for images of small-to-moderate size. We therefore also consider two alternative approaches to invert the GDB transform, one of which is a modification of Press' method.

Thus, this paper has four subthemes: 1) inversion of the GDB (DRT) transform; 2) construction of a linear invertible edge detection operator; 3) noise signature reduction in the GDB transform domain; and finally, 4) the effect of noise reduction on the classification of geomorphology by invariant terrain features. Section II concerns the Radon transform and its inverse, in particular the GDB algorithm (see Section II-B) and the use of iterative inverse methods (see Section II-C). Section III describes the full denoising algorithm, including the invertible Laplace operator (see Section III-A), and the suppression of noise signatures in the DRT domain (see Section III-B). Section IV describes the formation of invariant, unbiased features for terrain classification. Section V presents the MBES data (see Section V-A), examples of denoised images (see Section V-B), and the results of the classification experiment (see Section V-C). The summary and discussion (see Section VI) includes considerations on parameter choices and generalizability, elaborates further on the potential of block processing, and mentions other possible applications.

II. The Radon transform

A. Classical Transform and PSF

The classical Radon transform of a function $f : \mathbb{R}^2 \to \mathbb{R}$ is formed by integrating f along straight lines [29]. If $s \in \mathbb{R}$ is a signed distance from the origin, and $\boldsymbol{\theta} = \begin{bmatrix}\cos\theta & \sin\theta\end{bmatrix}$ is a unit vector, then the straight line orthogonal to $\boldsymbol{\theta}$ and parameterized by (θ, s) is $L(\theta, s) = \{\, \boldsymbol{x} \in \mathbb{R}^2 \mid \boldsymbol{x} \cdot \boldsymbol{\theta} = s \,\}$. The Radon transform $\mathcal{R}f$ is a function on the cylinder $C = [0, 2\pi) \times \mathbb{R}$ defined by

$$(\mathcal{R}f)(\theta, s) = \int_{L(\theta,s)} f(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \tag{1}$$

The adjoint, or backprojection operator, applied to a function $g : C \to \mathbb{R}$, is defined by

$$(\mathcal{R}^+ g)(\boldsymbol{x}) = \int_0^{2\pi} g(\theta, \boldsymbol{x} \cdot \boldsymbol{\theta})\, \mathrm{d}\theta, \tag{2}$$

which is 2π times the average of all line integrals passing through $\boldsymbol{x}$. The standard filtered backprojection algorithm for reconstructing f from $\mathcal{R}f$ is based on the fact that

$$(\mathcal{R}^+ g) * f = \mathcal{R}^+(g * \mathcal{R}f), \tag{3}$$

where $*$ denotes convolution [29]. One finds a function g such that $V \equiv \mathcal{R}^+ g \approx \delta$ is the Dirac delta function. Then, by (3), the reconstructed function is $\mathcal{R}^+(g * \mathcal{R}f)$, the backprojection

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

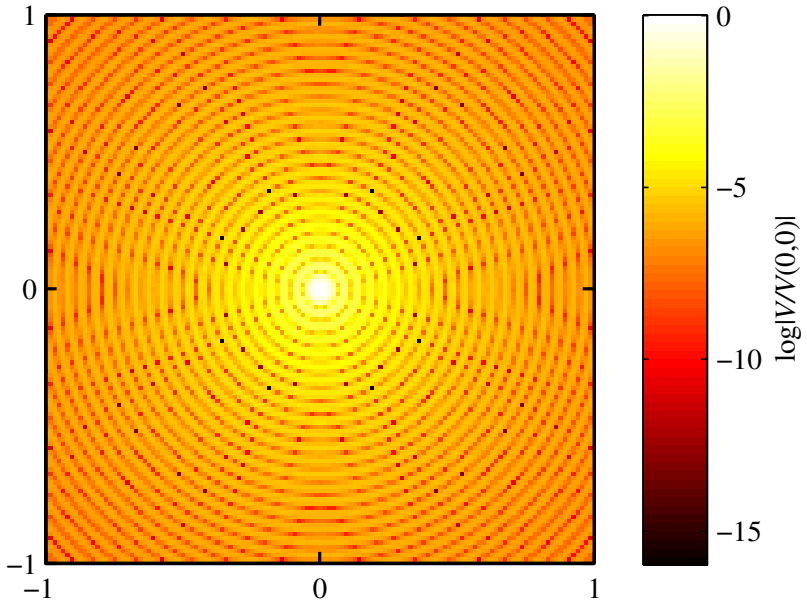


Fig. 1. PSF for filtered backprojection with a Shepp-Logan window with bandwidth $\Omega = 64$ cycles per unit length (128×128 pixels image). The number of projections is $p = 181$, which exactly meets the angular sampling criterion for this bandwidth [29].

of $\mathcal{R}f$ filtered by g; V is the point spread function (PSF), which determines how much the function f is blurred when passed through Radon transform and filtered backprojection.

For reconstructing bandlimited functions, as when f represents a digital image, g must be chosen such that the Fourier transform $\hat{V}(k) = \hat{\delta}(k) = 1/\sqrt{2\pi}$ for $|k| \leq \Omega$, i.e., the bandwidth of f. Several functions g that yield an appropriate window function for $\hat{V}(k)$ have been proposed and applied in practice [34]. One example is the Shepp-Logan window [35], for which the corresponding PSF $V(\boldsymbol{x})$ is plotted in Fig. 1 for the bandwidth $\Omega = 64$. A certain smoothing or distortion of the original function is implied by this PSF. Moreover, when applied to digital images (regular, discrete data), there will be an additional loss of accuracy from the fact that line integrals must be evaluated by interpolation between nearby pixels, or the (weighted) sum of all pixel values in a band containing the line.

B. GDB Discrete Transform

While filtered backprojection is a cornerstone of image reconstruction as in tomographic imaging, efficient evaluation of the forward Radon transform is important in its own right. For example, the classical Hough transform, used for straight line detection in image analysis [25], [36], may be implemented as a Radon transform when applied to binary images. A fast approximate method for computing line integrals was described independently in [31] and [32]. This algorithm does not involve weighted sums. Instead, each DRT element is the sum of pixel values along a certain graph which includes exactly one pixel from each column of the zero-padded image. Here $\boldsymbol{f}$ denotes the discrete set of pixel values $\boldsymbol{f} = \{ f_{i,j} \}$ indexed by row i and column j, where $f_{i,j} = 0$ for i, j outside the range $0 \leq i, j < N - 1$, and it is assumed that N is a power of 2. A graph $D_N(h, s)$ (using notation from [33]) is identified by its intercept h and rise s, meaning that it starts at $(i, j) = (h, 0)$ and ends at $(i, j) = (h + s, N - 1)$ (see Fig. 2). The construction of the graph recursive. In the first step, the image is divided into $N/2$ subimages, each two pixels wide, for which the lines $D_2(h, s)$

DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

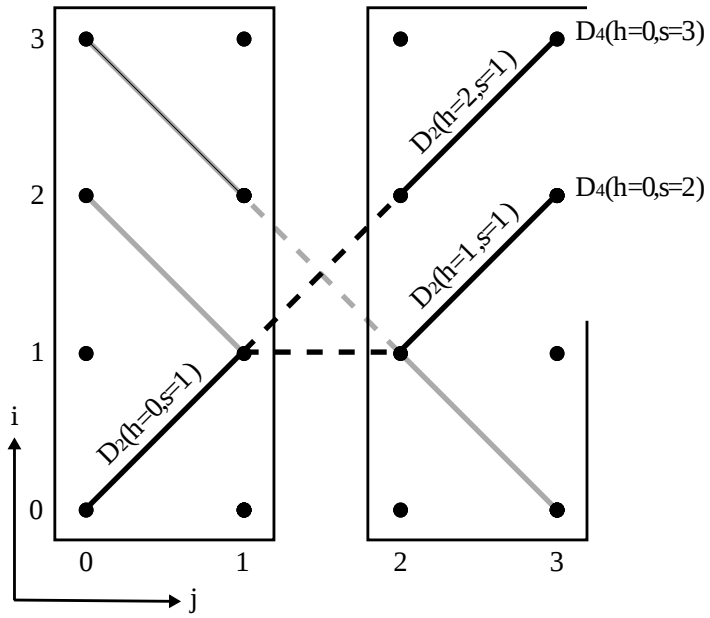


Fig. 2. Recursive construction of graphs in a 4×4 pixels image. The graphs are formed by joining graphs in the two 4×2 pixels subimages according to rule (4). The rise parameter s ranges from 0 to 1 (4×2 subimages) and 0 to 3 (4×4 image), yielding slopes from 0° to 45°. When the image is flipped upside down, the graphs are mapped to lines with slopes in the range -45° to 0° (faint lines).

are constructed. In the next step, graphs from pairs of adjacent subimages are combined, and so on, for $\log_2 N$ steps in all. Graphs from adjacent images are combined according to the following rule:

$$D_n(h, 2s) = D^{(L)}_{n/2}(h, s) \cup D^{(R)}_{n/2}(h + s, s) \tag{4a}$$

$$D_n(h, 2s+1) = D^{(L)}_{n/2}(h, s) \cup D^{(R)}_{n/2}(h + s + 1, s) \tag{4b}$$

where it is understood that the right-hand side of a union refers to the right (R) half of the image, and correspondingly for the left-hand (L) side. A graph in a subimage therefore forms part of two graphs in the twice-as-wide subimage (see Fig. 2).

Equation (4) is applied for $0 \le s < N$, i.e., for slopes from 0° to 45°. The other line integrals are performed in the same way after first transposing or flipping the original image. The full DRT is the disjoint union of four parts (quadrants);

$$(\mathcal{R}^1 f)(h, s) = \sum_{(i,j) \in D_N(h,s)} f_{i,j} \qquad (0^\circ \text{ to } 45^\circ) \tag{5a}$$

$$(\mathcal{R}^2 f)(h, s) = \sum_{(i,j) \in D_N(h,s)} f_{j,i} \qquad (45^\circ \text{ to } 90^\circ) \tag{5b}$$

$$(\mathcal{R}^3 f)(h, s) = \sum_{(i,j) \in D_N(h,s)} f_{j,N-1-i} \qquad (-90^\circ \text{ to } -45^\circ) \tag{5c}$$

$$(\mathcal{R}^4 f)(h, s) = \sum_{(i,j) \in D_N(h,s)} f_{N-1-i,j} \qquad (-45^\circ \text{ to } 0^\circ) \tag{5d}$$

The number of non-zero elements in each quadrant is at most $N^2 + N(N-1)/2$, and the minimum value for h is $-N+1$, for which the associated graph intersects the image only at a corner pixel.

C. Iterative Inverses

The GDB transform (DRT) $\mathcal{R}$ is a linear operator; each output element is a sum of image pixel values. In principle $\mathcal{R}$ may therefore be represented by a $P \times Q$ matrix $\boldsymbol{R}$, with $P = 6N^2 - 2N$ and $Q = N^2$. The matrix $\boldsymbol{R}$ is only moderately sparse, and a left-inverse of $\boldsymbol{R}$ is not sparse. The inverse problem is to solve the linear system

$$\boldsymbol{R}\boldsymbol{f} = \boldsymbol{d} \tag{6}$$

with respect to the image $\boldsymbol{f} \in \mathbb{R}^Q$, when the data vector $\boldsymbol{d} \in \mathbb{R}^P$, containing the line sums, is known. (Where appropriate we will use a vector to store a two-dimensional image, by concatenating the columns into a single column of length N^2.)

1) Normal Equations: The GDB transform has a corresponding fast recursive backprojection algorithm, which, analogously to (2), sums the values of all line integrals passing through a given pixel [32]. As shown in Appendix B, the matrix representation of the GDB backprojection algorithm is precisely the transpose $\boldsymbol{R}^T$. Therefore one may consider the normal equations, i.e.,

$$\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{f} = \boldsymbol{R}^T\boldsymbol{d}, \tag{7}$$

also when it is infeasible to use the explicit matrix representations. Several iterative algorithms that solve the normal equations (7) are available, e.g. the LSQR [37] and LSMR [38] algorithms.

In the proposed denoising method (see Section III-B), the transform $\boldsymbol{d}_0 = \boldsymbol{R}\boldsymbol{f}_0$ of the original image $\boldsymbol{f}_0$ is modified by a filter in the DRT domain to obtain some perturbation $\boldsymbol{d} = \boldsymbol{d}_0 + \Delta\boldsymbol{d}_0$. The vector $\boldsymbol{d}$ is generally not contained in the range $\mathcal{I}(\boldsymbol{R})$ of $\boldsymbol{R}$, which implies that (6) does not have a solution. We have the following range-null space decomposition of $\mathbb{R}^P$ (see Appendix B),

$$\mathbb{R}^P = \mathcal{I}(\boldsymbol{R}) \oplus \mathcal{N}(\boldsymbol{R}^T) \tag{8}$$

where $\oplus$ denotes the direct sum, and the range $\mathcal{I}(\boldsymbol{R})$ and the null space $\mathcal{N}(\boldsymbol{R}^T)$ are orthogonal subspaces of $\mathbb{R}^P$. A solution to (7) minimizes the residual norm $\|\boldsymbol{d} - \boldsymbol{R}\boldsymbol{f}\|$ [39, Ch. 6.9], and $\boldsymbol{d} - \boldsymbol{R}\boldsymbol{f}$ lies in the null space $\mathcal{N}(\boldsymbol{R}^T)$ of the backprojection operator.

Hence, (7) yields an optimal solution in the sense that $\|\boldsymbol{d} - \boldsymbol{R}\boldsymbol{f}\|$ is minimized. However, one should bear in mind that there is no definite best choice for $\Delta\boldsymbol{d}_0$. Rather, what matters is the quality of the restored image and the computational cost. For these reasons, we consider two alternative approaches that in experiments yielded better image reconstructions at less computational cost, and which converge faster when (6) does have a solution.

2) Press Inverse: Press [33] devised an approximate inverse for the GDB transform, based on the backprojection algorithm. The approximate inverse algorithm is a linear operator $\mathcal{B}$ that is applied recursively on downsampled versions of $\mathcal{R}^i\boldsymbol{f}$ $(i = 1, \ldots 4)$, to produce a smoothed image $\boldsymbol{f}_s$. The residual $\mathcal{R}\boldsymbol{f} - \mathcal{R}\boldsymbol{f}_s$ is backprojected and high-pass filtered to obtain an image correction $\delta\boldsymbol{f}$ which is added to the smoothed image $\boldsymbol{f}_s$. Since $\mathcal{B}$ is linear, it has a matrix representation $\boldsymbol{B}$.

Suppose (6) has a solution, i.e., $\boldsymbol{d} \in \mathcal{I}(\boldsymbol{R})$. We can show that $\boldsymbol{R}$ is injective (one-to-one), therefore, the solution must be unique (see Appendix A). Define $\boldsymbol{E} = \boldsymbol{1}_Q - \boldsymbol{B}\boldsymbol{R}$, and let $\rho(\boldsymbol{E})$

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

(a) Original image (b) Approximate inverse (c) Three iterations (iterative inverse)

Fig. 3. Press approximate inverse (b), equation (9a), and the Press iterative inverse (c), equation (9b) with $k = 2$, applied to the GDB transform of the test image in (a) (512×512 pixels). The original image (a) has gray level intensities in the range 0–1; the approximate inverse has intensities in approximately the range 0–1.4, but has been rescaled so that white corresponds to maximum intensity in all three images. The RMS difference between (a) and (b), and (a) and (c), can be read off the graph labeled "Press (camera)" in Fig. 4 below.

denote the spectral radius of $\boldsymbol{E}$ (the largest eigenvalue magnitude). The following recurrence relation developed in [33] converges to the solution of (6)

$$\boldsymbol{f}_0 = \boldsymbol{B}\boldsymbol{d}, \tag{9a}$$

$$\boldsymbol{f}_{k+1} \equiv \left(\sum_{j=0}^{k+1} \boldsymbol{E}^j \right) \boldsymbol{B}\boldsymbol{d} = \boldsymbol{f}_k + \boldsymbol{B}\left(\boldsymbol{d} - \boldsymbol{R}\boldsymbol{f}_k\right) \tag{9b}$$

provided $\rho(\boldsymbol{E}) < 1$. This condition implies that the $N^2 \times N^2$ matrix $\boldsymbol{B}\boldsymbol{R}$ is non-singular [40, Ch. 7.10]. Fig. 3 shows the Press approximate (9a) and iterative (9b) inverses applied to the GDB transform of a standard test image. The root-mean-square (RMS) error after three iterations is about 1%.

3) GMRES: However, the analysis in Appendix C suggests that, for $N \geq 512$, a small fraction of the eigenvalues of $\boldsymbol{E}$ have magnitudes larger than unity. This implies that the convergence of (9) halts after a certain number of iterations and the residual error begins to grow as some eigenimages get amplified. Nevertheless, the initial rate of convergence is typically very good, and we would therefore like to make use of the approximate inverse $\boldsymbol{B}$.

Applying the binomial formula to the Neumann series appearing in (9b) gives

$$\boldsymbol{f}_k = \sum_{j=0}^{k} \sum_{i=0}^{j} (-1)^i \binom{j}{i} \boldsymbol{A}^i \boldsymbol{b}, \tag{10}$$

where we have defined $\boldsymbol{A} \equiv \boldsymbol{B}\boldsymbol{R}$ and $\boldsymbol{b} \equiv \boldsymbol{B}\boldsymbol{d}$. Consequently $\boldsymbol{f}_k$ lies in the Krylov subspace [41]

$$\mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{b}) \equiv \operatorname{span}\{\, \boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^k \boldsymbol{b} \,\}. \tag{11}$$

The generalized minimum residual (GMRES) algorithm [42], [43] produces a sequence of iterates $\boldsymbol{x}_k$, where $\boldsymbol{x}_k$ is the solution to the least squares problem

$$\min_{\boldsymbol{x} \in \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{b})} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|, \; k = 1, 2, \ldots \tag{12}$$

DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

in the Euclidean norm. Hence, GMRES gives approximate solutions to the $N^2 \times N^2$ linear system obtained by left-multiplying (6) by $\boldsymbol{B}$,

$$\boldsymbol{BRf} = \boldsymbol{Bd}. \tag{13}$$

GMRES, like (9), involves only matrix-vector multiplications, which are implemented by the recursive algorithms $\mathcal{R}$ and $\mathcal{B}$.

The exact solution to (13), if $\boldsymbol{A}$ is nonsingular, lies in the Krylov space $\mathcal{K}_d(\boldsymbol{A}, \boldsymbol{b})$, where d is the degree of the minimal polynomial of $\boldsymbol{A}$ [41]. Note that we have only established that $\boldsymbol{A}$ is nonsingular for $N < 512$. However, $\rho(\boldsymbol{E}) < 1$ is not a necessary condition for $\boldsymbol{A}^{-1}$ to exist, it is only required that $\lambda = 1$ is not an eigenvalue of $\boldsymbol{E}$. Moreover, the necessary condition that $\boldsymbol{R}$ (and $\boldsymbol{R}^T$) is injective has been established. If $\rho(\boldsymbol{E}) < 1$, then (9) and (12) converge to the same solution, as is also evident from (32) (Appendix C).

4) Stopping Criterion: For the kth iterate x_k in (12), the relative residual norm is defined as $r_k = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\| \,/\, \|\boldsymbol{b}\|$. From (9b),

$$\frac{\left\|\boldsymbol{f}_{k+1} - \boldsymbol{f}_k\right\|}{\|\boldsymbol{Bd}\|} = \frac{\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{f}_k\|}{\|\boldsymbol{b}\|} = r_k \tag{14}$$

For both GMRES and the Press iterative inverse the iteration is stopped when $r_k < \tau$, for some tolerance τ. In denoising applications with image size $N = 1024$, it turned out that a tolerance of $\tau \sim 0.1$ was usually enough to achieve the desired result. This corresponded to three to four iterations.

5) Comparison: To demonstrate the difference between the three methods, we applied them to the GDB transform of two test images, the cameraman image from Fig. 3 and an eigenimage of $\boldsymbol{E}$ with eigenvalue $\lambda > 1$. The eigenimage was found by Arnoldi iteration (see Appendix C). Both images have 512×512 pixels and intensity values in the range 0–1. The inverse problem has 262144 unknowns, and the dimension of $\boldsymbol{R}$ is 1571840×262144.

Fig. 4 shows the RMS value of $\boldsymbol{f} - \boldsymbol{f}_k$ [Press inverse (9)] and $\boldsymbol{f} - \boldsymbol{x}_k$ [GMRES inverse (12)] as a function of k, where $\boldsymbol{f}$ denotes the original image. The RMS error of the approximate inverse ($k = 0$) is about 10 % for both images. Fig. 4 shows that both methods apparently converge to the cameraman image when $k \leq 20$. The GMRES inverse converges to the eigenimage as well, but the Press inverse does not. Fig. 5 shows the RMS error for all three methods as a function of computing time (cameraman image only). During the initial iterations, the Press and GMRES inverses converge almost equally fast, but the amplification problem that destroys convergence for the Press inverse is now evident. The GMRES inverse approaches $\boldsymbol{f}$ to high accuracy, while the normal equation solver LSQR has slower convergence. The results are similar when plotting the residual matrix norm $\|\boldsymbol{f} - \boldsymbol{f}_k\|_2$. The preliminary conclusion from these tests is that the GMRES inverse is to be preferred because of its fast convergence when the inverse problem has a solution.

III. Denoising algorithm

The main idea is to transform the image (DTM) by an edge operator (filter) $\mathcal{L}$ followed by the DRT $\mathcal{R}$. The edge operator accentuates fine-scale intensity changes; the DRT concentrates accentuated linear features. The algorithm should cause as little distortion of real features as possible. Image reconstruction is done by applying the inverse operator $\mathcal{L}^{-1}\mathcal{R}^{-1}$. $\mathcal{R}^{-1}$ is implemented using one of the iterative methods described in Section II-C, such that $\mathcal{R}^{-1}\mathcal{R} \approx \mathbf{1}_Q$ to arbitrary desired accuracy. However, edge operators are in general not one-to-one

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

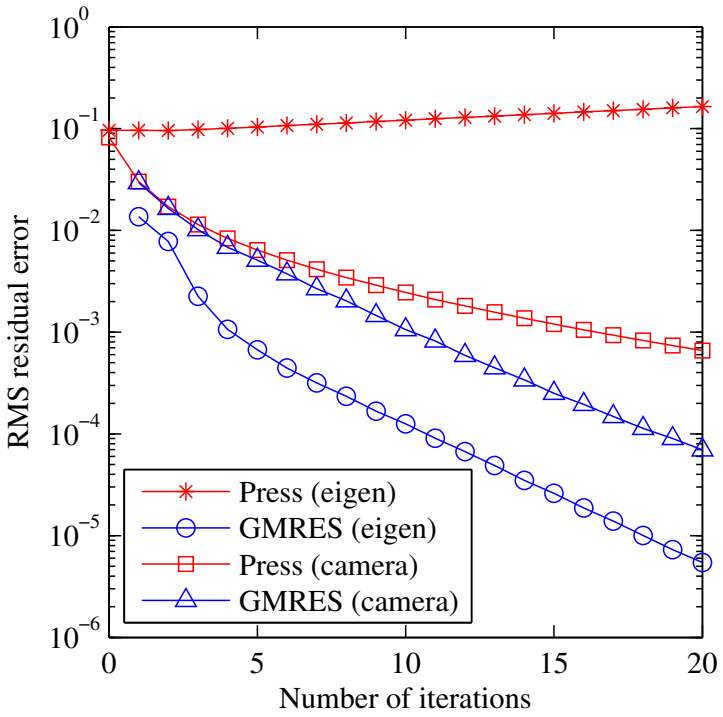


Fig. 4. RMS difference between original image and inverse iterate of the GDB transform, using (9) and (12). Results for two images are shown: the cameraman image in Fig. 3a, and an eigenimage of $\boldsymbol{E}$ with eigenvalue magnitude $|\lambda| > 1$.

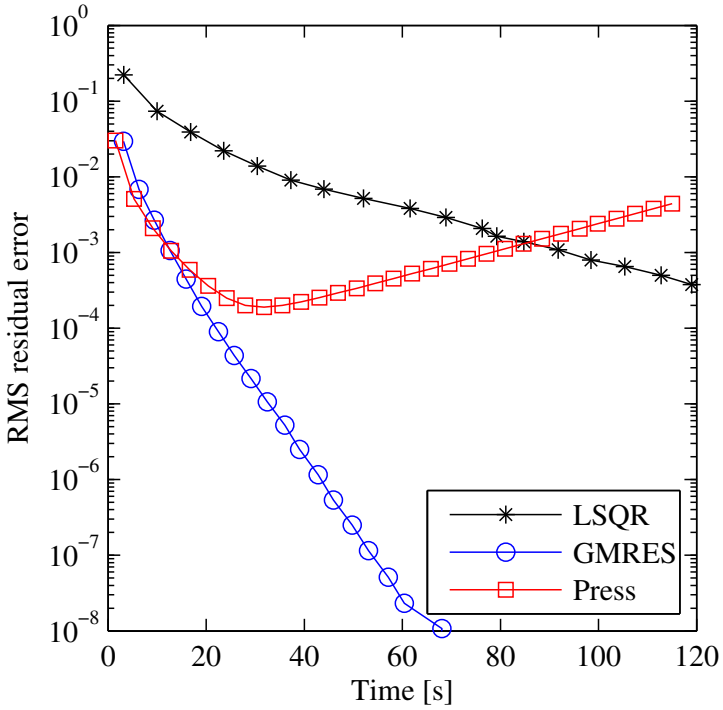


Fig. 5. RMS difference between original image (cameraman image in Fig. 3a) and the inverse iterate of the GDB transform, plotted as a function of computing time (single-core processing using Matlab on a standard desktop computer). Three inverse methods are compared: LSQR (normal equation solver), GMRES (12), and the Press inverse (9) based on the Neumann series.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

mappings [44, Ch. 12.3] and hence not invertible. Therefore we first consider a suitable invertible edge operator.

A. Invertible Edge Detection Operator

For the reasons stated in Section I, our preferred operator is the Laplacian. The non-invertibility of the Laplacian is perhaps best examined in the frequency domain. A digital image $\boldsymbol{f} = [f(u,v)]_{u,v=1}^{N}$ with $N \times N$ pixels, N even, may be constructed by regularly sampling the periodic, band-limited function given by the Fourier series

$$f : \mathbb{R}^2 \to \mathbb{R}, \quad f(x,y) = \sum_{m=-N/2}^{N/2-1} \sum_{n=-N/2}^{N/2-1} c_{mn} e^{2\pi i(nx+my)}, \tag{15}$$

where f may be viewed as a function on the torus $\mathbb{T}^2 = \mathbb{R}^2/\mathbb{Z}^2$, obtained geometrically by gluing together the two pairs of opposing edges of the unit square $[0,1] \times [0,1]$. By letting the first row in the image ($m = 1$) correspond to $y = 0$, the coefficients c_{mn} in (15) are related to the 2D discrete Fourier transform $\mathcal{F}$ by

$$c_{mn} = \frac{1}{N^2} \mathcal{F}[\boldsymbol{f}](m + N \bmod N, n + N \bmod N), \tag{16}$$

with $\mathcal{F}[\boldsymbol{f}](m,n) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} f(u,v) e^{-2\pi i(mu+nv)/N}$. A function

$$f \in L^2(\mathbb{T}^2) = \{\, g : \mathbb{T}^2 \to \mathbb{R} \mid \int_{\mathbb{T}^2} |g(\boldsymbol{x})|^2 \, \mathrm{d}\boldsymbol{x} < \infty \,\}$$

may be identified with its set of Fourier coefficients. For twice-differentiable functions in $L^2(\mathbb{T}^2)$, the continuous Laplace operator $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ may be defined by

$$\Delta : c_{mn} \longmapsto -4\pi^2(m^2+n^2)c_{mn}, \ (m,n) \in \mathbb{Z}^2. \tag{17}$$

In terms of the wave vector $\boldsymbol{k} = 2\pi(m,n)$, the transfer function is $\Delta(\boldsymbol{k}) = -\boldsymbol{k} \cdot \boldsymbol{k}$, which is a manifestation of the rotation invariance of the continuous Laplace operator. For the restricted set of functions

$$L_0^2(\mathbb{T}^2) = \left\{ \, f \in L^2(\mathbb{T}^2) \;\middle|\; \int_{\mathbb{T}^2} f \, \mathrm{d}x = 0 \, \right\}, \tag{18}$$

for which $c_{00} \equiv 0$, the operator $D : L_0^2(\mathbb{T}^2) \to L_0^2(\mathbb{T}^2)$ defined by

$$D : c_{mn} \longmapsto \begin{cases} \frac{c_{mn}}{-4\pi^2(m^2+n^2)} & \text{for } (m,n) \in \mathbb{Z}^2 \setminus \{0\} \\ 0 & \text{for } m = n = 0. \end{cases} \tag{19}$$

is a right-inverse so that $\Delta\, Df = f$ [45, Ch. 4.4]. This suggests forming an invertible Laplace-like operator for images via (15)–(17) by perturbing the factor (m^2+n^2) in (17) away from zero, i.e.,

$$-4\pi(m^2+n^2) \mapsto -4\pi^2(m^2+n^2+\epsilon^2), \ 0 < \epsilon^2 \ll 1.$$

However, while this approach works well for some images, computing the Laplacian in the frequency domain may fail because the function (15) need not be differentiable on the boundary of the unit square, whence image distortions may be incurred (one may consider, for example, the image formed by regularly sampling the function $g(x,y) = xy \exp(x^2+y^2)$ on the unit square $[0,1] \times [0,1]$).

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

Instead, a more robust approach is to perturb the PSF of a discrete approximation Δ_d to the Laplacian in the spatial domain. We start with the operator implemented by circular (because of the periodicity condition) convolution with the PSF

$$\boldsymbol{h} = \frac{1}{2}\begin{bmatrix} 1/2 & 1 & 1/2 \\ 1 & -6 & 1 \\ 1/2 & 1 & 1/2 \end{bmatrix}. \tag{20}$$

The transfer function is

$$\mathcal{F}[\boldsymbol{h}] = 4\left[\cos^2(\hat{k}_x/2)\cos^2(\hat{k}_y/2) - 1\right] \tag{21}$$

$$\approx -\hat{k}^2 + \frac{3}{32}\hat{k}^4 - \frac{1}{96}\cos(4\phi)\hat{k}^4 + \mathcal{O}(\hat{k}^6), \tag{22}$$

where $\hat{\boldsymbol{k}} = 2\pi(m,n)/N$ is a scaled wave vector with polar coordinates (k,ϕ) [44, Ch. 12.5]. This operator is isotropic at long wavelengths. We have $\mathcal{F}[\Delta_d \boldsymbol{f}] = \mathcal{F}[\boldsymbol{h}]\mathcal{F}[\boldsymbol{f}]$, and direct deconvolution fails because of a single zero (at zero frequency) in the transfer function $\mathcal{F}[\boldsymbol{h}]$. (Here and below it is understood that the optical transfer function $\mathcal{F}[\boldsymbol{h}]$ has the same dimension as $\mathcal{F}[\boldsymbol{f}]$, obtained after zero-padding $\boldsymbol{h}$ and circularly shifting the DFT.) This problem can be amended by adding a small random component to the PSF $\boldsymbol{h}$ (not to the filtered image). We let $\boldsymbol{h}_M^0$ denote the matrix (20) symmetrically zero-padded to size $M \times M$, and let $\boldsymbol{\eta}_M(\epsilon)$ be an $M \times M$ matrix where the elements η_{ij} are random numbers such that $|\eta_{ij}| \leq \epsilon \ll 1$. The edge operator $\mathcal{L}$ is then implemented by circular convolution with the modified PSF

$$\boldsymbol{h} \to \boldsymbol{h}_M^0 + \boldsymbol{\eta}_M(\epsilon). \tag{23}$$

There is no guarantee that any randomly picked matrix on the form (23) has a zero-free transfer function, so one should check (for any dimension N only one such matrix need be found). This modified PSF does the same job as the Laplace operator as long as $\epsilon \ll 1$. There is no compromise or approximation involved in (23). It simply means replacing one operator with another which fulfills the same purpose while also being invertible. In this paper, we have used $M = 7$ and $\epsilon = 10^{-3}$.

B. Noise Suppression Filter

The GDB transform can be visualized by appropriately flipping, vertically shifting, and stitching together the four quadrants $\mathcal{R}^1\boldsymbol{f}, \ldots, \mathcal{R}^4\boldsymbol{f}$ into a Möbius band, as shown in Fig. 6 [33]. It is a Möbius band because the left and right edges, corresponding to vertical projections ($\pm 90°$), can be identified after flipping one of them upside down. The horizontal axis, the projection angle, maps to the rise parameter s in the GDB transform. A change in the vertical direction corresponds to a change in intercept parameter h. It is to this representation of the transformed data that we apply the noise filter.

Fig. 6a shows the GDB transform of a Laplacian image of a DTM with parallel horizontal tracks (it is actually the transposition of a DTM with vertical tracks). The track line discontinuities appear as a string of high intensity spots at the $0°$ projection angle. The other conspicuous feature is the concentration of noise at $\pm 90°$, i.e., in the direction perpendicular to the tracks. In addition, there is a more diffuse salt-and-pepper-like pattern across the whole image. We think the noise amplification at $\pm 90°$ is a manifestation of the motion-induced noise, which tends to be oriented perpendicular to the tracks.

DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

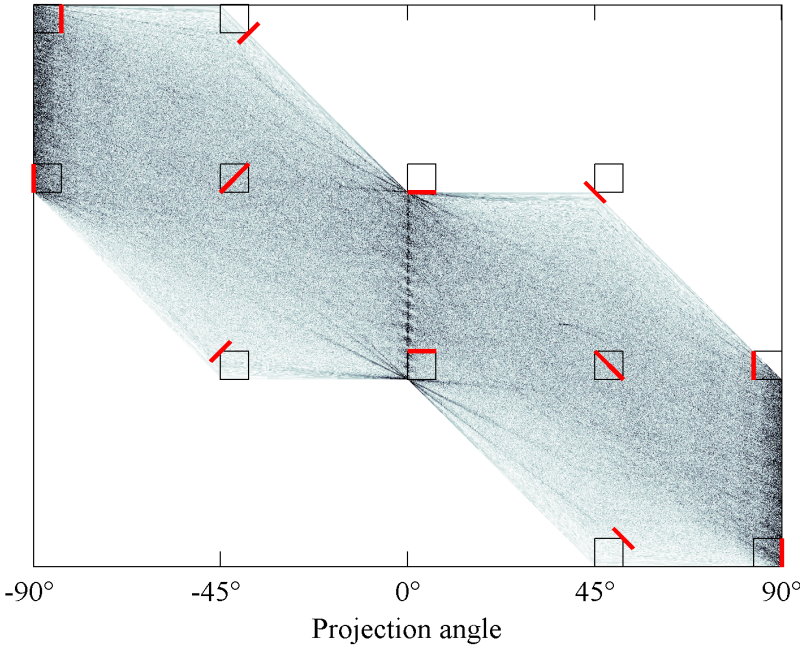
Fig. 6b shows the effect of applying a 3×3 median filter, a standard choice for salt-and-pepper noise removal. The track line spots persist, and must be removed in a separate step. Fig. 7 shows the 0° column of the DRT where the track line discontinuities appear as distinct spikes in the data. When the track lines are roughly parallel as they often are in remote sensing situations, the spikes will always lie in a narrow vertical band at the angle given by the orientation of the tracks. The spikes can then be eliminated by zeroing all values in the narrow vertical band, or for example, by using a 1-D polynomial approximation for each column in the band (see Fig. 7).

Note that, when the edge operator is applied to a DTM (image) by circular convolution with (23), high pixel intensities may occur along the four edges of the resulting image. This is because the DTM may not be continuous there, in the sense of being a function on $\mathbb{T}^2$ as in (15). The edges of the image therefore get transformed to high intensity spots in the DRT domain. These spots are not artifacts in the data, and should not be affected by the DRT domain filter. To avoid boundary effects in the reconstructed DTM, the edge filter should be applied to a detrended DTM. A low-frequency approximation should first be subtracted, and subsequently added back to the final reconstructed DTM. This step removes discontinuities at the edges. As will be shown in Section V-B, standard low-pass filters tend to preserve some of the noise/artifacts that we want to remove, and this must be avoided. In their place we use a Chebyshev polynomial approximation that achieves the desired effect. The algorithm, denoted $\mathcal{T}(p, D)$, is characterized by the polynomial degree p and a downsampling factor D (see Appendix D). The complete denoising algorithm may now be summarized as follows:

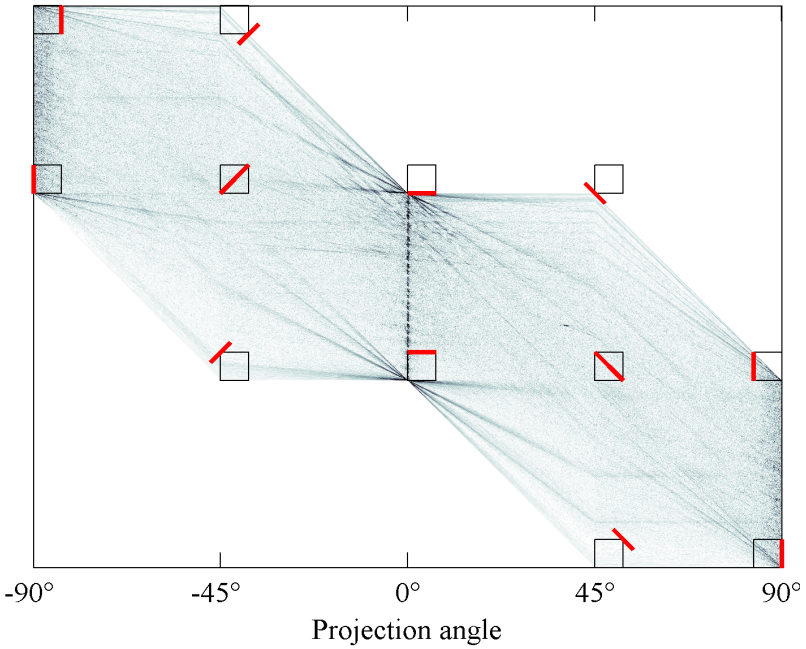
procedure $\boldsymbol{f} = \text{DENOISE}(\boldsymbol{f}_0)$
  $\boldsymbol{a} \leftarrow \mathcal{T}(p, D)\boldsymbol{f}_0$
  $\boldsymbol{f} \leftarrow \boldsymbol{f}_0 - \boldsymbol{a}$
  $\boldsymbol{f} \leftarrow \mathcal{L}\boldsymbol{f} = \boldsymbol{h}(M, \epsilon) * \boldsymbol{f}$ ▷ circular convolution
  $\boldsymbol{d} \leftarrow \mathcal{R}\boldsymbol{f}$
  $\boldsymbol{d} \leftarrow \text{medfilt}(\boldsymbol{d})$ ▷ 3×3 median filter
  $\boldsymbol{d} \leftarrow$ suppress peaks at track line angle
  $\boldsymbol{f} \leftarrow \mathcal{R}^{-1}\boldsymbol{d}$
  $\boldsymbol{f} \leftarrow \mathcal{L}^{-1}\boldsymbol{f} = \mathcal{F}^{-1}\left[\mathcal{F}\left[\boldsymbol{f}\right] / \mathcal{F}\left[\boldsymbol{h}\right]\right]$
  return $\boldsymbol{f} + \boldsymbol{a}$
end procedure

First, the trend $\mathcal{T}(p, D)\boldsymbol{f}_0$ is computed and stored in memory and subtracted from the image (DTM) f_0. The modified Laplace operator is applied to the detrended image, and the PSF is stored for later use in the inverse (deconvolution) step. Next, the forward GDB DRT is applied to the Laplacian image, and the four quadrants in (5) are merged into a single matrix as illustrated in Fig. 6. A 3×3 median filter is applied to this matrix. The median filter may not be relevant in other applications; we have used the procedure, excluding the median filter, to remove scalloping (parallel stripes) in synthetic aperture radar (SAR) images; see further remarks in Section VI. The peaks in the DRT corresponding to linear artefacts are suppressed, e.g., by zeroing the columns corresponding to the orientation of the lines. The inverse DRT is next computed using one of the iterative methods described in Section II-C. The inverse Laplacian image is computed by direct deconvolution with the PSF $\boldsymbol{h}(M, \epsilon)$ (23). Finally, the trend is added back to obtain the restored image (DTM).

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380



(a) DRT



(b) 3×3 median filter of DRT

Fig. 6. Typical example of the GDB transform of an edge-filtered bathymetric DTM with motion-induced noise and track line artifacts is shown in (a). (b) shows the effect of applying a 3×3 median filter in the DRT domain, i.e., to the image (a). A unit (one pixel) increase along the vertical axis corresponds to a unit change in the intercept parameter h. However, the parameter h is not constant along horizontal lines in the figure, it is only constant within each of the four quadrants. This is because the quadrant transforms $\mathcal{R}^1 f, \ldots, \mathcal{R}^4 f$ have been translated vertically so that they fit together in a Möbius band [33]. The color map is symmetric about zero, so that white corresponds to zero and black corresponds to high positive or negative values. The DTM has 1024×1024 pixels and horizontal line artifacts (tracks). The small squares symbolize the image (DTM) extent, and the red (short, thick) lines indicate the projection that corresponds to the associated pixel in the GDB domain. For example, the topmost non-zero value at 0° corresponds to the bottom horizontal line, while the top left value corresponds to the rightmost column in the image.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

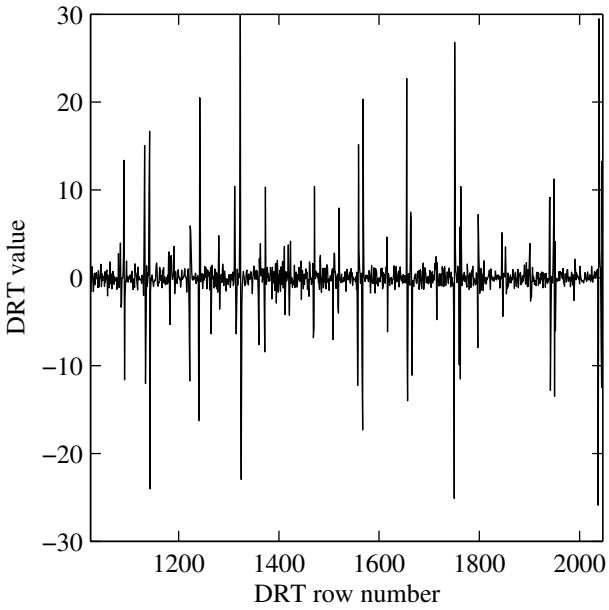


Fig. 7. GDB transform (non-zero part) at 0° projection angle, showing track line discontinuities as a series of spikes.

IV. Unbiased terrain features

As noted above, local geometric attributes, such as gradient or curvature, may be used to visualize landforms and geomorphology. Furthermore, images may be segmented (or classified) according to variations in morphological patterns caused by differing erosional and depositional processes. The rate at which local patterns are correctly classified will be a relative measure of image quality. Preferably, local patterns should be represented by intrinsic properties that do not depend on image orientation, intensity (or depth/elevation), or large-scale variations such as approximately constant slopes. We also require that features should not depend on externally imposed parameters, such as the direction of a derivative, the spatial relationship of pixel pairs, or the elevation angle of artificial illumination. The operator will, effectively, only determine the spatial resolution, i.e, the size of the pixel blocks used to identify local patterns.

Features with such properties may be obtained from the Laplacian Δf, which is approximately rotation invariant and insensitive to depth and constant slopes. The Laplacian is a measure of curvature, and in some situations additional information on patterns can be obtained by computing the Gaussian curvature K and the mean curvature H. K and H are intrinsic geometric properties of more general differentiable surfaces embedded in three-dimensional space [46]. For our purpose, a DTM may be considered as a regular set of samples from the graph of a function $f(x, y)$ of rectangular coordinates. For the special case of a graph surface, the formulae for K and H are

$$K = \frac{f_{xx} f_{yy} - f_{xy}^2}{\left(1 + f_x^2 + f_y^2\right)^2} \tag{24}$$

$$H = \frac{f_{xx}\left(1 + f_y^2\right) - 2 f_{xy} f_x f_y + f_{yy}\left(1 + f_x^2\right)}{2\left(1 + f_x^2 + f_y^2\right)^{3/2}}, \tag{25}$$

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

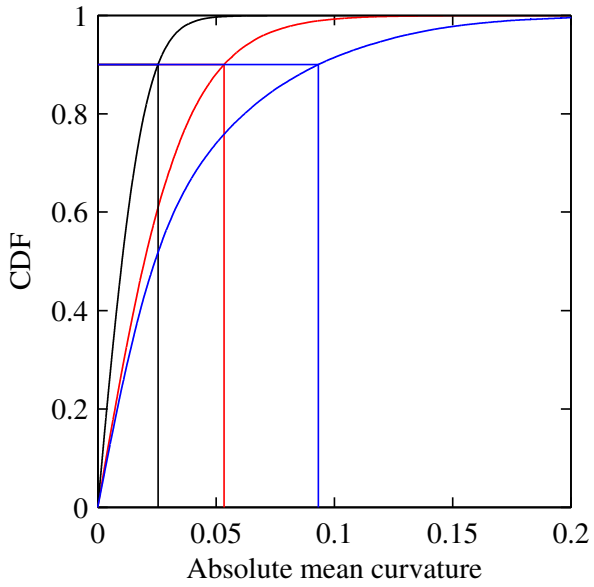


Fig. 8. ECDFs for the quantity $|H|$ (in a rescaled image), for three areas with different geomorphology (see Section V-C). Given the local CDF for some geometrical property, a Q-dimensional feature vector may be formed by taking the inverse CDF at Q values $0 < q_1 < \ldots < q_Q < 1$ (only $q = 0.9$ shown here).

where f_x denotes $\partial f/\partial x$ etc. [47]. The curvatures K and H have some useful local and global properties. For example, for a domain $V \subset \mathbb{R}^2$, the functional $\int_V H\sqrt{1+|\nabla f|^2}\,\mathrm{d}x\,\mathrm{d}y$ is equivalent to the minimum curvature functional of thin-plate spline interpolation [47], and measures how much the area of the surface $f(V)$ changes in response to a unit length deformation in the surface normal direction. K and H are related to the more familiar principal curvatures, i.e., κ_1 and κ_2, by $K = \kappa_1\kappa_2$ and $H = (\kappa_1 + \kappa_2)/2$ (at any point $p \in \Sigma$, i.e., a differentiable surface, κ_1 and κ_2 are the maximum and minimum curvatures of the family of 1D curves formed by the intersection of Σ with all planes containing the surface normal $\boldsymbol{N}_p$ at p).

An image block of size $S \times S$ pixels may be characterized by the distribution of a particular geometric property. Here we have used the empirical cumulative distribution function (ECDF), as shown in Fig. 8. Specifying Q values $0 < q_1 < \ldots < q_Q < 1$ (e.g., the 10-quantiles) yields a Q-dimensional feature vector via the inverse ECDF. One may also concatenate the feature vectors of several geometric properties, e.g. K and H or H and Δf.

For comparison and as an alternative to the ECDF, we have applied the local binary pattern (LBP) operator, a widely used texture operator that also yields distribution-based feature vectors. LBPs were introduced by Ojala, Pietikäinen and Harwood [48], building on the work on texture spectra by Wang and He [49]. The original LBP is formed by thresholding the pixel values of an 8-neighborhood with respect to the center pixel, producing a string of eight zeros and ones. The string is interpreted as a binary number in the range 0–255. The distribution (histogram) of LBPs in an image block of size $S \times S$ pixels may be used as a measure of the texture in that block [50]; the number of features equals the number of bins in the histogram. Since this gives very high-dimensional feature vectors (256 components), it is common to consider only the so-called uniform patterns corresponding to the binary strings with at most two transitions between 0 and 1 (e.g. 00111000 has two transitions and is uniform, while 01001000 has four transitions). There are 58 such patterns [51].

DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

By definition, the LBP is invariant with respect to uniform (or locally uniform) changes in image intensity, but it is not rotation invariant. However, rotating an image at angles $2\pi/P$ (where $P = 8$ for an 8-neighborhood) corresponds to a circular bit shift in the patterns. Therefore, one obtains rotation-invariant distributions by identifying all patterns that are equal modulo a circular bit shift. For 8-neighborhoods, this results in 36 distinct features [52]. For the subset of uniform patterns, the corresponding number is nine.

The block size S must be large enough to ensure reliable histogram estimation; here we have used $S = 16$, the smallest size considered in [50], and $S = 32$. To compute the LBPs we have used the VLFeat library [53], which was modified to handle rotation-invariant patterns. In the VLFeat LBP algorithm, each pixel-level pattern contributes to the histograms of the four nearest blocks via bilinear interpolation.

V. Tests and results

A. Multibeam Bathymetry

The denoising algorithm was tested on MBES data from three areas in the northern North Sea (A, B, and C: Fig. 9). The data were acquired by the Norwegian Defence Research Establishment (FFI, Kjeller/Horten, Norway), on the research vessel M/S HU Sverdrup II, in 2005 (C), 2006 (B), and 2008 (A), respectively. Area A was surveyed with a Kongsberg Maritime (KM, Horten, Norway) EM 710 echo sounder, while areas B and C were surveyed with a KM EM 1002. The EM 1002 has a center frequency of 95 kHz; the array is semi-cylindrical and emits a fan-shaped beam with 2.0° along-track $3\,\mathrm{dB}$ beamwidth, and it forms 111 beams with 2.0° beamwidth across track. The EM 710 operates at 70–100 kHz, and has orthogonally oriented rectangular transmit and receive arrays. The EM 710 transmits in three sectors simultaneously with distinct frequencies in inner and outer sectors, and forms up to 400 $1.0^\circ \times 0.5^\circ$ beams on reception (0.5° transmit beamwidth). The EM 710 system may generate two separate swathes per ping (at slightly different forward tilt angles), yielding a higher density of soundings than the older EM 1002 system.

The survey area stretches from the eastern margin of the North Sea plateau (water depth circa 100 m) and into the Norwegian Trench (water depth circa 300 m, see Fig. 9). The three areas A, B, and C have different sedimentary and morphological characteristics, and the data have different noise characteristics. Area A (EM 710) lies on the North Sea plateau, where the seabed is sandy. Area B covers part of the western slope of the Norwegian Trench. This area has an irregular pattern of elongated pockmarks that may be a result of fluid seepage and water current erosion [54], [55]. Area C is flat with soft, silty sediments, but there are many circular pockmarks caused by vertically migrating shallow gas. In general, noise and artifacts will be most visible in flat areas such as C, and less visible in areas with greater depth variation, such as B, which has a depth range of about 90 m.

For each area (A, B, and C), DTMs with resolutions in the range of 1.5 m to 8 m were constructed by block averaging the MBES soundings. Empty cells were filled in by taking the mean depth of the four nearest cells. All the processing results presented in the next section were obtained on grid models with 1024×1024 pixels. The EM 1002 data set had, before we received it, been put through a standard post-processing chain that included corrections for tidal variations, manual editing, and removal of outliers. The EM 1002 data have motion-induced noise which is most pronounced at the outer beams, possibly a result of imperfect sound refraction corrections; this also causes visible discontinuities between adjacent track

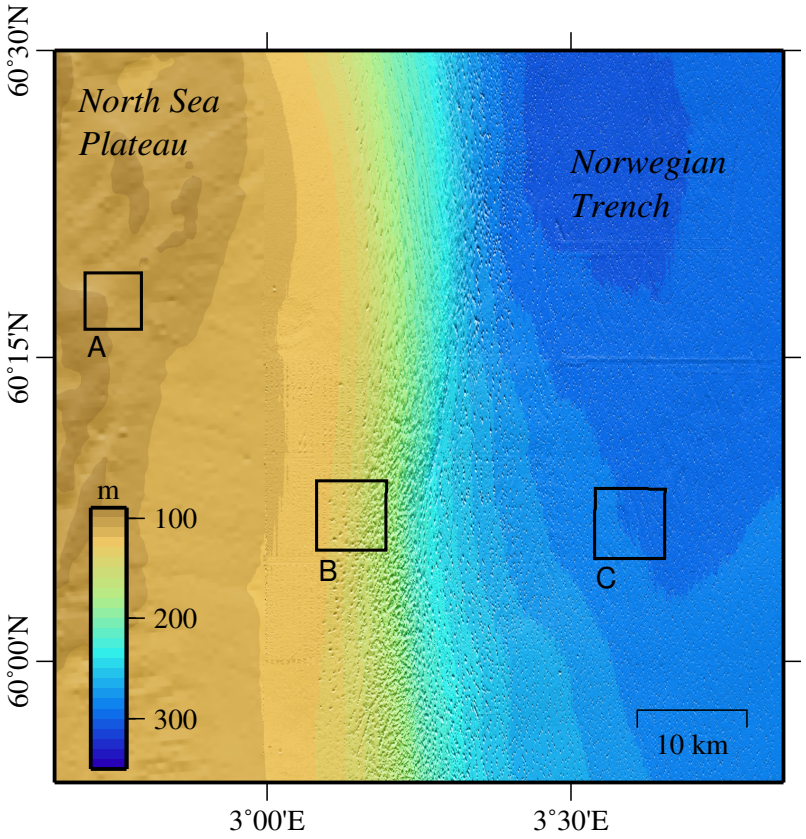


Fig. 9. Overview and bathymetry of the survey area in the northern North Sea. The denoising algorithm was tested in the areas labeled A, B, and C. Area A was surveyed with a Kongsberg Maritime (KM) EM 710 MBES, while areas B and C were surveyed with a KM EM 1002 MBES.

lines. The EM 710 data were raw, unprocessed data, where, prior to construction of DTMs we applied only a crude thresholding to exclude large outliers. The lack of tidal correction causes marked discontinuities between track lines. There is also a low-frequency swell-induced noise that appears as wavy patterns "propagating" in the direction of the ship tracks. While MBES data normally have post-processing corrections, it is useful to test a new algorithm on a challenging data set.

B. Denoising results

The DTMs for areas A, B, and C were DRT-filtered with the following settings: For areas A and B, the trend surface $\mathcal{T}f$ was computed using the Chebyshev polynomial approximation algorithm of Appendix D with the parameter values $p = 12$ and $D = 4$; for area C the parameter values were $p = 6$ and $D = 4$. Note that the approximation algorithm is based on least squares fitting, which mitigates the phenomenon of Runge (oscillations near the boundary) [56]. The modified Laplacian was computed using the PSF (23) with size $M = 7$ and perturbation level $\epsilon = 10^{-3}$. A 3×3 median filter was applied to the DRT of the Laplacian. In all three areas, the track lines are oriented approximately north-south; the track line discontinuities therefore appear as high-intensity spots at $\pm 90^\circ$ in the DRT domain, or, equivalently, at 0° in the DRT of the transposed image (as in Fig. 6). To suppress linear artefacts, each DRT column was zeroed in approximately the range $\langle -1^\circ, 1^\circ \rangle$. (One might also use some interpolation scheme to reset the zeroed values, but this was not done in the

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

present work and we do not think it is important when the DRT is applied to the Laplacian of a detrended surface.) The inverse DRT was computed using the GMRES algorithm with stopping criterion $r_k = 10^{-6}$ or maximum six iterations, whichever is reached first. We found, however, that the less tight criterion of $r_k = 10^{-2}$ also yielded good results, good enough as perceived by the human eye. The rate of convergence will depend on the nature of the noise and data; we found that the algorithm converged faster when inverting the SAR images (briefly mentioned in Section VI) compared with the MBES terrain models. As discussed below (Section VI), except for the polynomial degree p the results shown here were not very sensitive to the parameters, and consequently, little parameter tuning was required. How to determine the best value for p a priori is an open question.

For comparison, various lowpass filters were applied to the DTMs or directly to attribute images, some examples of which are shown in the following. Lowpass filters were implemented as convolutional averaging filters with constant or Gaussian weights and filter diameters between 3 and 13 pixels (for Gaussian filters, the diameter was defined as 6 standard deviations). In addition, a Butterworth filter was implemented by multiplying the centered normalized spectrum of $\mathcal{F}[f - \mathcal{T}f]$ or $\mathcal{F}[\Delta f]$ (see Fig. 10b) with

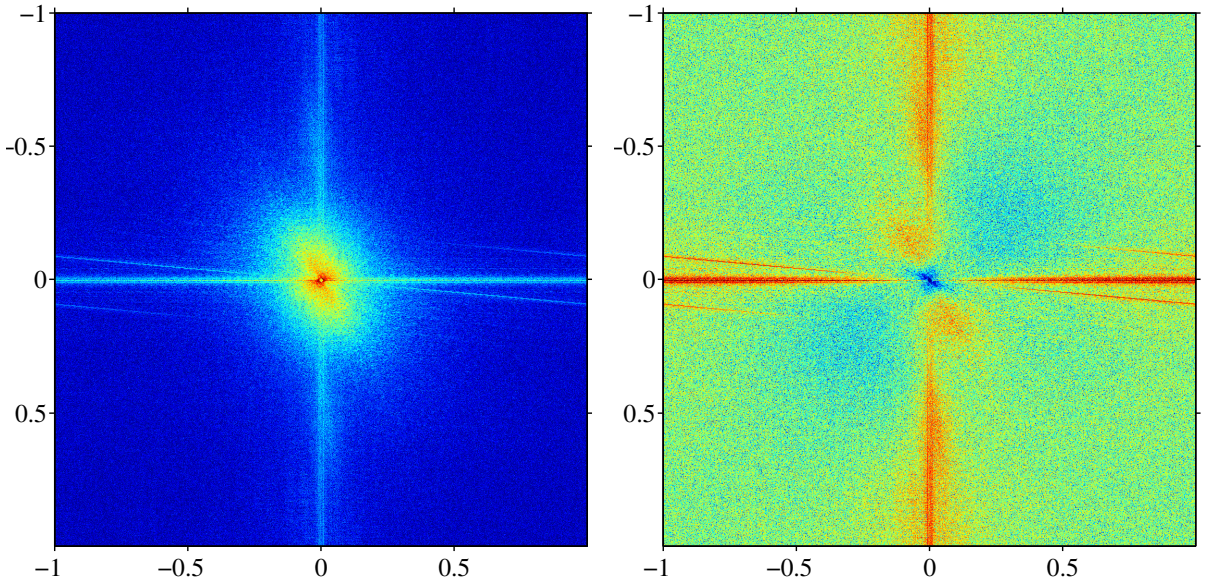
$$G(\nu_1, \nu_2) = \frac{1}{(1 + \sqrt{\nu_1^2 + \nu_2^2}/D_0)^{2n}}, \tag{26}$$

where n and D_0 are adjustable parameters.

In general, we found that low-pass filters were ineffective against stripe noise caused by track line discontinuities. Depending on the size of the convolution kernel, low-pass filters can significantly reduce motion-induced noise, but only at the cost of severe smoothing of real features. The DRT filter is targeted at linear artifacts and produced comparatively good attenuation of stripe noise. The DRT filter also has a smoothing effect which reduced motion-induced noise with apparently less distortion of real features. However, severe motion-induced noise, seen in particular on the outer beams in some of the EM 1002 data, is difficult to eliminate using either low-pass filters or the DRT filter.

A partial solution to this problem was to apply a DRT filter locally by subdividing the 1024×1024 pixels DTMs into nonoverlapping blocks of 32×32, 64×64 or 128×128 pixels. This DRT filter was designed to attenuate linear artifacts in narrow sectors close to perpendicular to the track lines. Applying the filter in non-overlapping blocks ensures fast processing. Moreover, at least for block size $N \leq 64$, it is possible to use a matrix representation for the DRT inverse (see further remarks in Appendix E). It does not seem obvious in advance that disjoint block processing should work satisfactorily, i.e., without producing block artifacts in the final composite image. Indeed, local DRT filters applied directly to gradient images did produce block artifacts. In the present results, the DRT was applied to the Laplacian images, and the denoised gradient images were derived from the final, restored DTMs. These gradient images have no visible block artifacts.

Fig. 10 shows the power spectrum of the detrended original DTM $f - \mathcal{T}f$ for area A (see Fig. 10a), and the power spectrum of the Laplacian image $\Delta(f - \mathcal{T}f)$ (see Fig. 10b). These spectra reflect that the noise patterns are directed and broadbanded. For both spectra it is difficult to design a filter that eliminates the artifacts without distorting the real bathymetric features. Fig. 11 shows the frequency response of the Butterworth filter (26) with parameters $n = 2$ and $D_0 = 400$. This filter was applied to the Laplacian image of area A (cf. Fig. 10b) to produce the lowpass-filtered Laplacian image shown in Fig. 12d.



(a) DTM (detrended) (b) Laplacian of detrended DTM

Fig. 10. Centered Fourier log-power spectra $2\log|\mathcal{F}[\cdot]|$ (normalized frequency) of detrended DTM for area A (a) and corresponding Laplacian image (b).

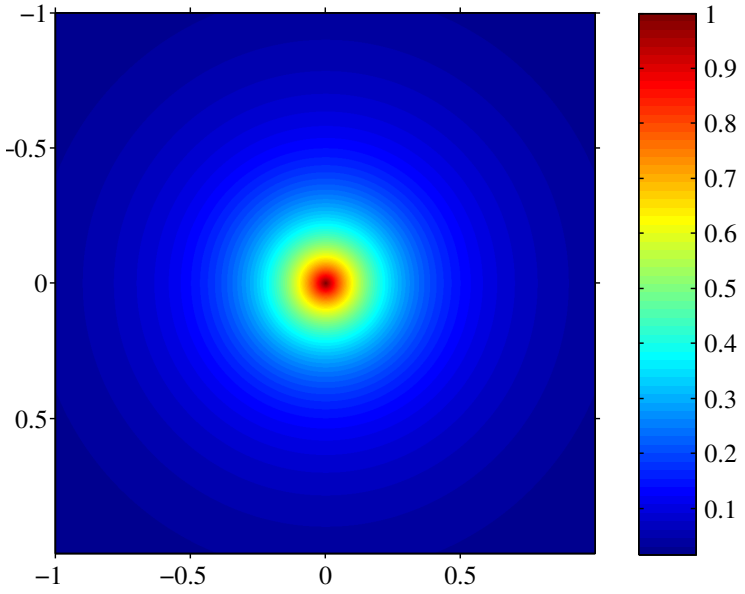


Fig. 11. Centered frequency response, in normalized frequency, of Butterworth low-pass filter (26) with $n = 2$ and $D_0 = 400$. This filter was used to produce the image in Fig. 12d below.

Fig. 12 shows the effect of the DRT filter and lowpass filters on the Laplacian $\Delta(f - \mathcal{T}f)$ for area A. In the original image (see Fig. 12a), the morphological features are poorly visible. The track line discontinuities cause parallel vertical stripes. The DRT-filtered version (see Fig. 12b) has no stripes and better defined morphology. As demonstrated in Section V-C, moreover, noise reduction is essential to achieve accurate discrimination between different geomorphological classes using second-derivative-based attributes. The low-pass-filtered images in Fig. 12c (7×7 boxcar filter applied to $f-\mathcal{T}f$) and 12d (frequency-domain Butterworth filter applied to $\Delta[f - \mathcal{T}f]$) retain the vertical stripes. In particular, it does not help to filter

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

the attribute image $\Delta(f - \mathcal{T}f)$ directly, nor is the effect on stripes sensitive to the filter size. It may also be seen that lowpass filters cause greater smoothing of terrain features compared to the DRT filter.

Fig. 13 shows gray level maps of bathymetry (gray-level-coded water depth) in area A before and after the DRT filter was applied. Gray level (color) maps are usually not the best way to visualize morphological details imaged by instruments like the MBES nor are they as sensitive as attribute images to the forms of noise studied here. The maps were included to show that the DRT filter did not alter the depth of real topographical features. As described in Section III-B, to ensure this, it is necessary to subtract a trend surface from the DTM before applying the Laplacian operator, to avoid having high values for the Laplacian along the boundary. If a low-frequency distortion appears in the final restored DTM, one should try and alter the degree p of the approximation operator $\mathcal{T}(p, D)$ (see Appendix D). Bear in mind that the data in this area had not been through a post-processing chain; in particular, there is no correction for tidal variations. This causes relatively large discontinuities that are not completely eliminated. The effect of the DRT filter is best seen in the attribute images.

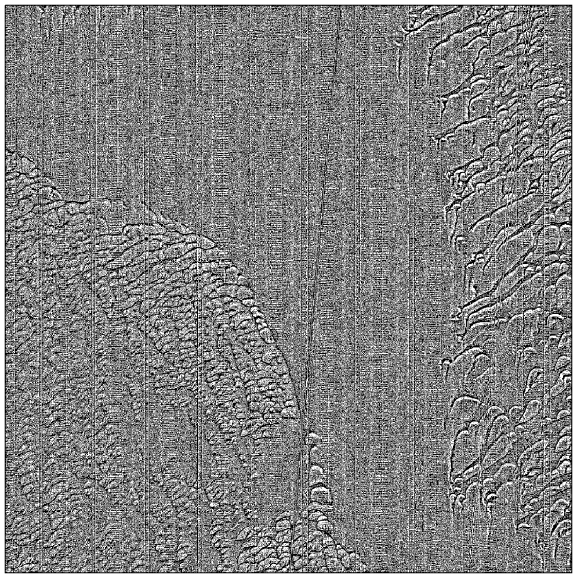
With the given orientation of the track lines, the stripe noise is most conspicuous when taking the horizontal gradient (see Fig. 14). Again, by comparing Fig. 14b and Fig. 14c, it is seen that the DRT filter is more effective than the lowpass filter in removing stripes, whereas the latter yields a less sharp view of morphology. The straight line that runs at a steep angle across the images is a real feature, probably a pipeline. The partial attenuation of this feature is an unwanted side effect.

Fig. 15 shows an example of motion-induced noise from area C, in vertical gradient images, and the effect of block processing with the DRT filter (128×128 pixels blocks). These images show many pockmarks (craters) on an almost flat seabed. When experimenting with the standard Radon transform and filtered backprojection, we found that many of the small pockmark features were distorted by the filter, even with a high sampling rate in the angular domain. This side effect was eliminated when using the GDB transform and iterative inverse. Low-pass filters, in this example, a 5×5 pixels boxcar filter, can remove much of the motion-induced noise (see Fig. 15b), but a better result is obtained with a DRT filter (see Fig. 15c). In this example, the filter worked by zeroing the DRT in two narrow sectors symmetric about the horizontal direction.

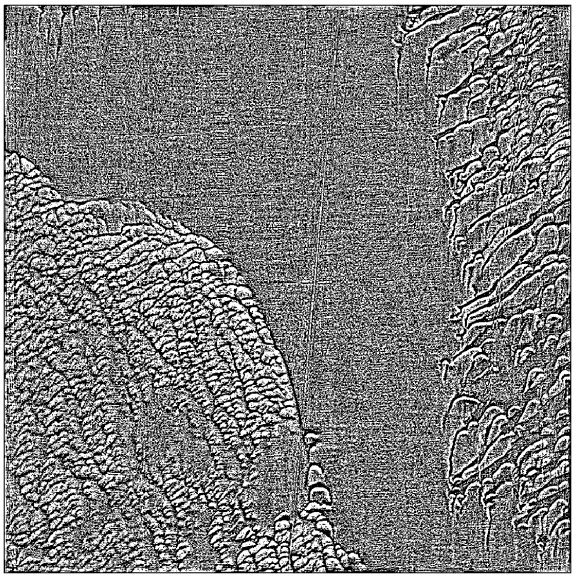
A yet more difficult example, from area B, is shown in Fig. 16. The interference of motion-induced noise patterns, between the outer beams of adjacent track lines, creates an effect which is visible in both the horizontal (see Fig. 16a and 16b) and vertical (see Fig. 16c and 16d) gradient images. The DRT filter applied directly to area B did attenuate the vertically oriented noise patterns but not completely (see Fig. 16b). Moreover, low-pass filters were not effective in this respect. Block processing with a DRT filter, as in Fig. 15c, with block size 32×32, does markedly reduce the noise level. This result (see Fig. 16d) was achieved without any discernible smoothing or distortion of the underlying geomorphological patterns showing the ocean current-modified pockmarks of area B.

As the true terrain form and the amount of noise are not known, direct evaluation of the signal-to-noise ratio (SNR) and the level of image distortion is not possible. However, the inverse coefficient of variation (ICV) may be used as an approximate measure of SNR and hence of image quality [21], [22], [57]. The ICV for an image region W is defined as

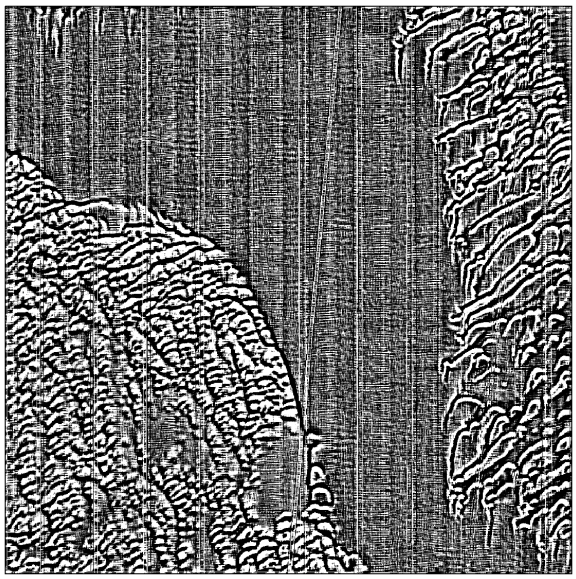
$$\text{ICV} = \frac{R_a}{R_{sd}}, \tag{27}$$



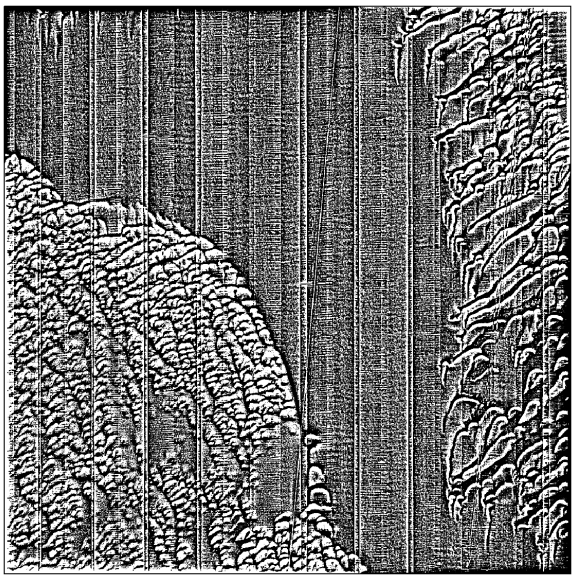
(a) Laplacian of original DTM (Δf)



(b) Laplacian of DRT-filtered DTM



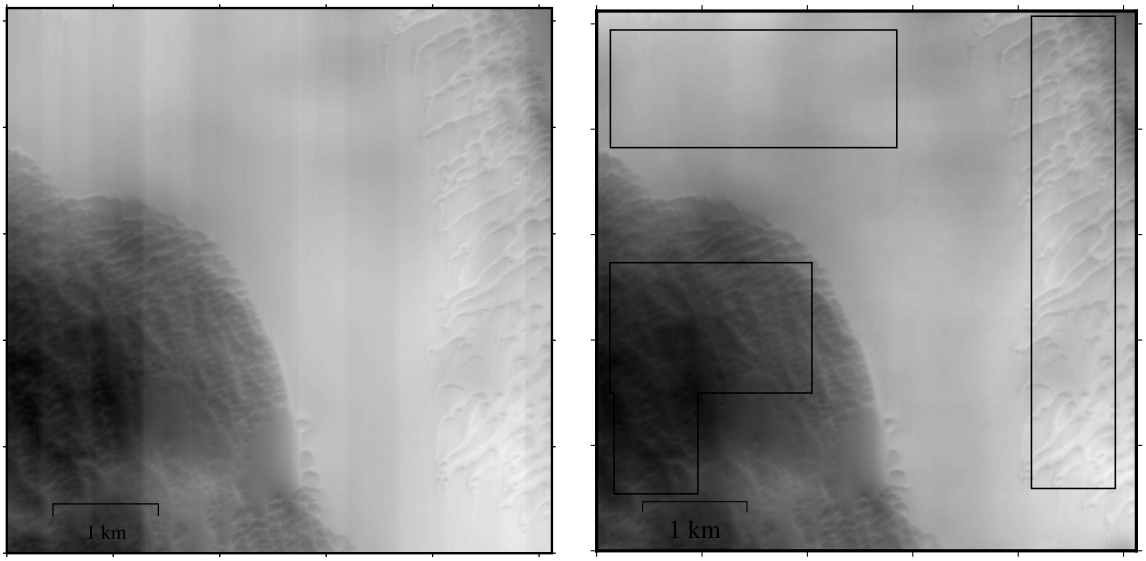
(c) Laplacian of lowpass-filtered DTM



(d) Lowpass-filtered Laplacian of original DTM

Fig. 12. The Laplacian of the DTM for area A. The color (gray) scale is identical in all four images. Low-pass filters are not effective for removing the vertical track line discontinuities; Fig. (c) was obtained after smoothing the original DTM with a 7×7 pixels boxcar averaging filter, while Fig. (d) was obtained by smoothing the Laplacian of the original DTM with a Butterworth filter.

where R_a is the mean pixel value in W and R_{sd} is the standard deviation of the pixels in W. In the works [21], [22], and [57], the ICV is computed in two or four small (10×10 pixels), homogeneous, noisy image regions, and R_{sd} is assumed to be mainly determined by noise. Here, we have computed the ICV for the whole DTM, in 8×8 pixels blocks, so that the result can be displayed as an image (see Fig. 17) showing the linear discontinuities as regions with low ICV (SNR) (to save space we only include the result for area A). Another measure employed in the above-cited works is the mean relative deviation (MRD), which is computed in noise-free regions. However, as the present images are affected by (motion-



(a) Original DTM (b) DRT-filtered DTM

Fig. 13. Gray level maps of bathymetry in area A (depth range of 94–110 m). The rectangles in Fig. 13b mark the areas representing the three classes of seafloor morphology in the classification tests (see Section V-C). Gray level (color) maps alone do normally not yield a good visual impression of morphology and noise (although moderately useful in this particular case). Here, they demonstrate that major features and absolute depth values are preserved after taking the inverse Radon transform and inverse Laplace operator.

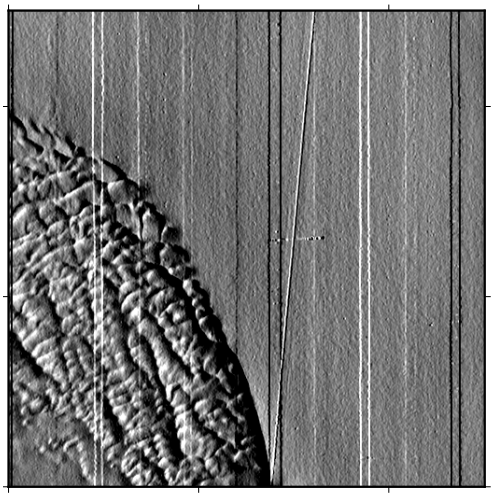
induced) noise effectively everywhere, the MRD is not considered here. The ICV may, by definition, be high for blurred (or smoothed) images, which conflicts with the perceived sense of poor image quality. On the other hand, we emphasize that the DRT filter better preserves the image sharpness compared to low-pass filters. We have therefore also attempted to quantify the effect of denoising in terms of the practical measure of terrain classification error.

C. Effect on Classification

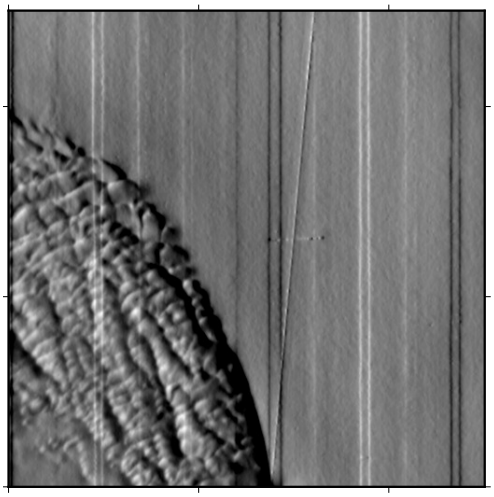
The classification experiment was restricted to a single area (A) to ensure that extraneous factors, such as differences in the noise characteristics or instrument resolution, did not improve the classification accuracy artificially. Three main geomorphological classes were identified, labeled, and defined by the training areas enclosed by rectangles in Fig. 13b. Each training area contains about 500 non-overlapping subcells with 16×16 pixels, for which the local ECDFs and rotation-invariant LBP histograms were computed for the quantities K (Gaussian curvature), H (mean curvature), Δf (Laplacian), and $f - \mathcal{T} f$ (depth residual). With respect to ECDFs, each cell is characterized by the set of 10-quantiles, which gives 9 features (or 18 if two geometric quantities are combined, etc.). For comparison, we have also tested the full set of uniform LBPs, which are not rotation invariant. The experiment was also run on 32×32 blocks, which yields more robust estimation of local distributions at the cost of lower spatial resolution and less training samples.

The classification results are presented here as learning curves (see Fig. 18), where the estimated classification error obtained with the Bayesian normal density classifier (quadratic discriminant function) is plotted versus the number of features D used in the classifier. A range of other classifiers were also tested, but none performed significantly better than the normal density classifier. The forward feature selection method was used to select subsets

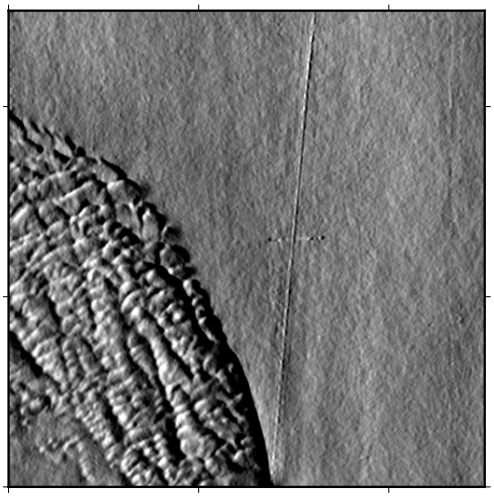
Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380



(a) Horizontal gradient of original DTM ($\nabla_x f$)

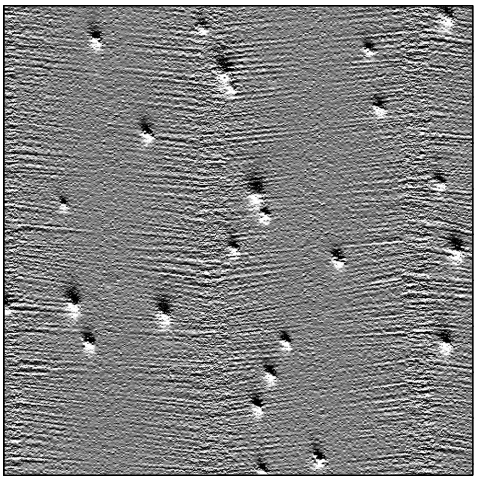


(b) Horizontal gradient of lowpass-filtered DTM

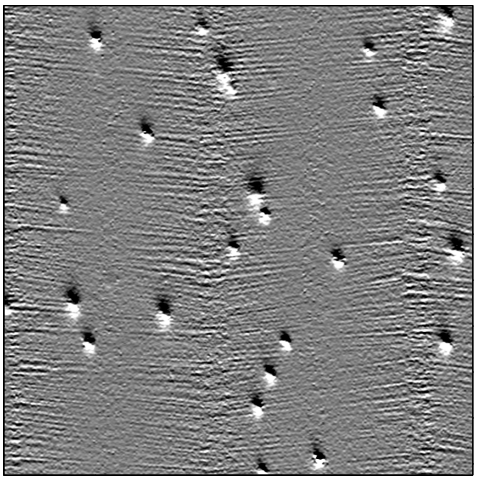


(c) Horizontal gradient of DRT-filtered DTM

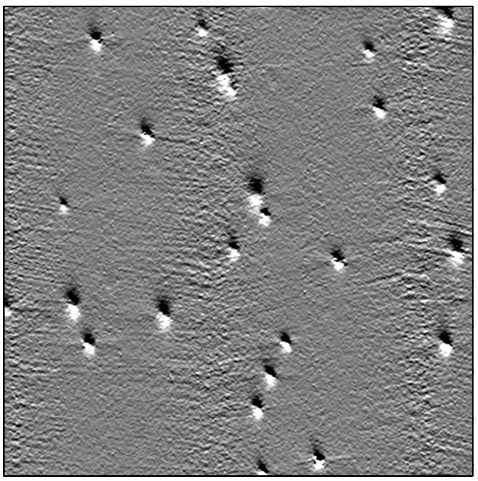
Fig. 14. Horizontal gradients (detail from area A) show vertical linear artifacts due to track line discontinuities and lack of tidal corrections. Fig. (b) was obtained after filtering the original 1024×1024 pixels DTM with a Butterworth lowpass filter with parameters $n = 2$ and $D_0 = 400$ [see equation (26)]. The single oblique line seen is most likely a real feature (e.g., a pipeline).



(a) Vertical gradient of original DTM ($\nabla_y f$)

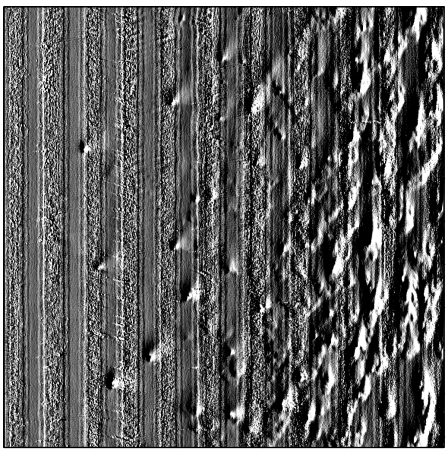


(b) Vertical gradient of lowpass-filtered DTM

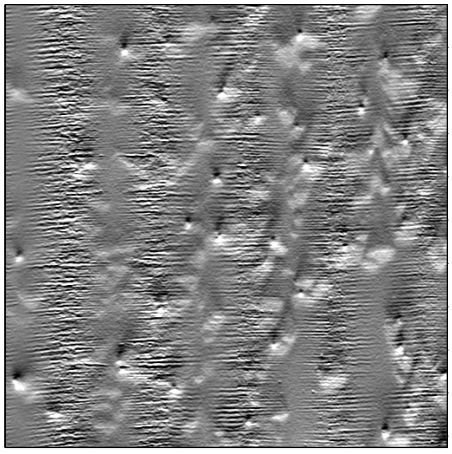


(c) Vertical gradient of block-DRT-filtered DTM

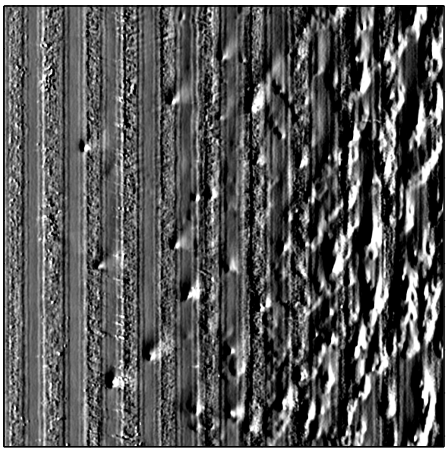
Fig. 15. Vertical gradients in a flat area with pockmarks (detail from area C) show motion-induced artefacts forming a "fishbone pattern", where the overlapping areas between adjacent swathes (vertically oriented) correspond to the spine of the fish. Fig. (b) was obtained by filtering the original 1024×1024 pixels DTM with a 5×5 pixels boxcar averaging filter. Block processing the DTM with a DRT filter produced Fig. (c) (32×32 pixels block size).



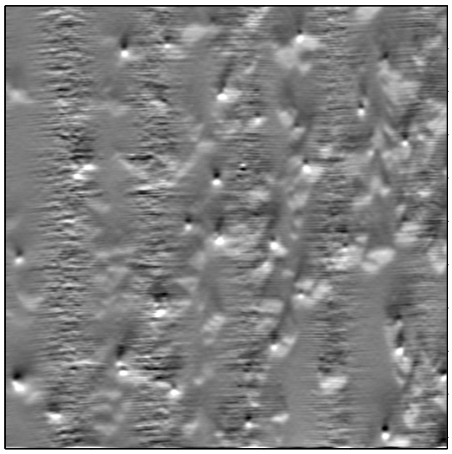
(a) Horizontal gradient of original DTM



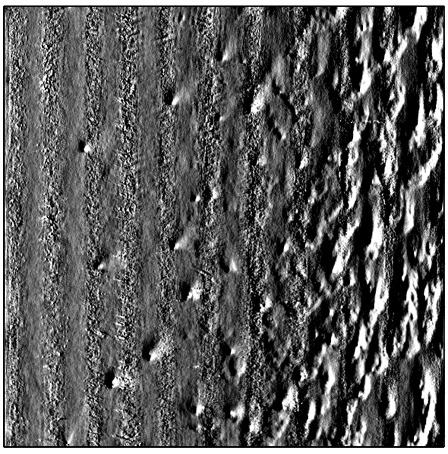
(b) Vertical gradient,original DTM (detail)



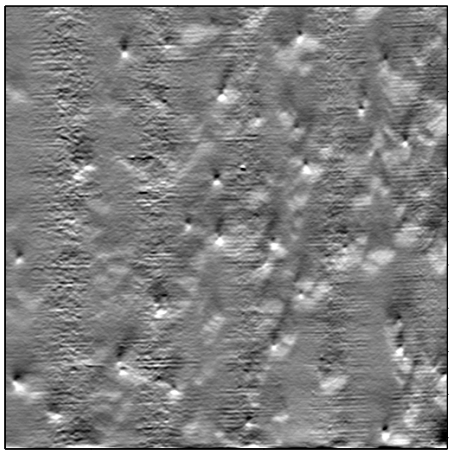
(c) Horizontal gradient, LP-filtered DTM



(d) Vertical gradient, LP-filtered DTM

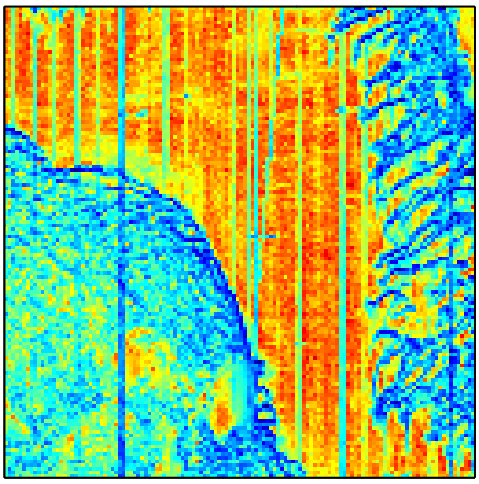


(e) Horizontal gradient, DRT-filtered DTM

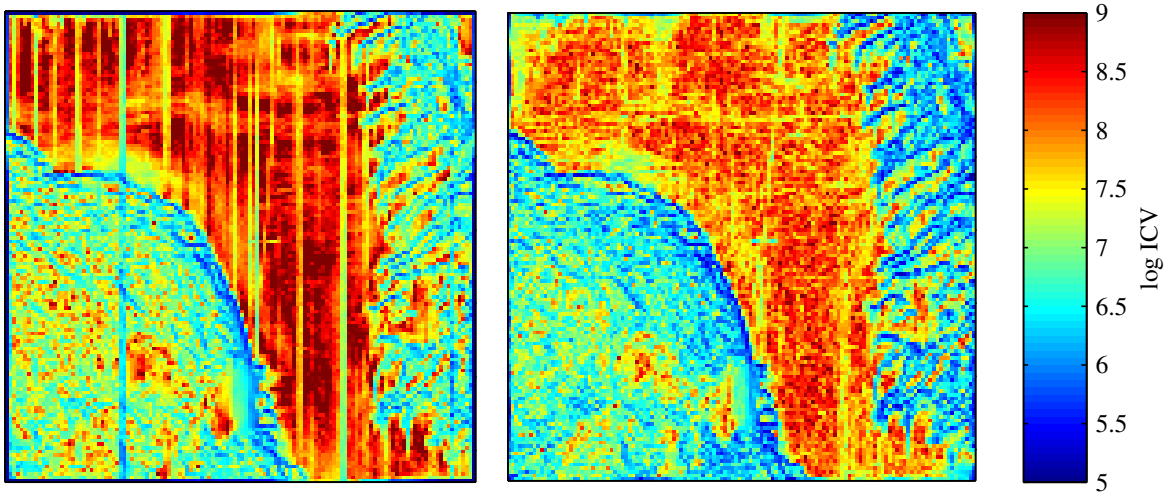


(f) Vertical gradient, DRT-filtered DTM

Fig. 16. Depth estimation errors and motion-induced noise cause vertical linear artifacts in the horizontal gradient image of area B (a). The DRT filter, applied to the 1024×1024 pixels DTM, dampens much of the stripe noise (e), whereas a lowpass filter (5×5 boxcar) is ineffective (c). However, the severe motion-induced noise, shown in Fig. (b) by taking the vertical gradient in a close-up corresponding to the lower right quadrant of (a), is difficult to eliminate without blurring the image (d). A partial solution, which reduces noise without blurring, is to apply a DRT-filter in (nonoverlapping) blocks (here 32×32 pixels) (f); the DRT is applied to Δf (Laplacian of original DTM) and dampens nearly-horizontal linear structures. Here, one color map has been used for the left column panels and another for the right column panels, because the general scale of magnitudes is different for horizontal and vertical gradients.



(a) Original DTM



(b) Low-pass-filtered DTM

(c) DRT-filtered DTM

Fig. 17. Inverse coefficient of variation (ICV) computed for area A (cf. Fig. 12 and 13) in 8×8 pixels blocks. The ICV is a measure of the SNR, assuming that the variance of the pixels is due to noise. In (a) and (b) the linear discontinuities in the DTM are seen as regions with low ICV (SNR). In other regions, smoothing (b) produces high ICV values by definition of the ICV. However, in reality, pixel variance is due to both real terrain variations as well as noise, and smoothing affects real features.

of D features. To avoid a potential selection bias caused by performing feature selection and error estimation on the same data set (see, e.g., [58], [59]), we have used the holdout approach. The complete dataset was randomly split into a training dataset and a test dataset, with 75 % of the samples assigned for training. Feature selection was subsequently done using 10-fold cross-validation on the training set only, using the normal density classification accuracy as selection criterion. Following feature selection, the trained classifier was applied to the test data set to get the error estimate. To reduce the variance of the error estimate, the whole process was repeated 20 times (random splits) and the average error computed.

The maximum estimated classification accuracy for various types of features is listed in Table I. For DRT-filtered data, good accuracy (92–95 %) was attained for $S = 32$ using K, H, or Δf, while moderately good accuracy (83–89 %) was attained for $S = 16$. The best results were obtained using uniform LBPs applied to Δf (Fig. 18d). However, one

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

TABLE I
OVERALL CLASSIFICATION ACCURACY (%)

Features	Raw image		Filtered image	
	16×16	32×32	16×16	32×32
ECDF H and Δf	72.9	82.4	83.4	93.7
LBP (rot. inv.) Δf	55.3	63.3	85.4	91.5
LBP (uniform) Δf	61.6	71.4	88.6	95.2
ECDF $f - \mathcal{T}f$	74.8	75.9	74.2	73.9

should bear in mind that despite the fact that Δf is (approximately) rotation invariant, the distribution of uniform LBPs is not. Fig. 18d also demonstrates the adverse effect of having a low ratio between the number of training samples and the number of features (for $S = 32$). It is evident that noise filtering is necessary to obtain good classification results when using noise-sensitive attributes such as curvature; the improvement is as high as 30 % in some instances (see Fig. 18b). On the other hand, less reliable results may be expected when using the local distributions of the depth residual $f - \mathcal{T}f$ (Fig. 18c) primarily because the result will depend on the particulars of the trend surface $\mathcal{T}f$.

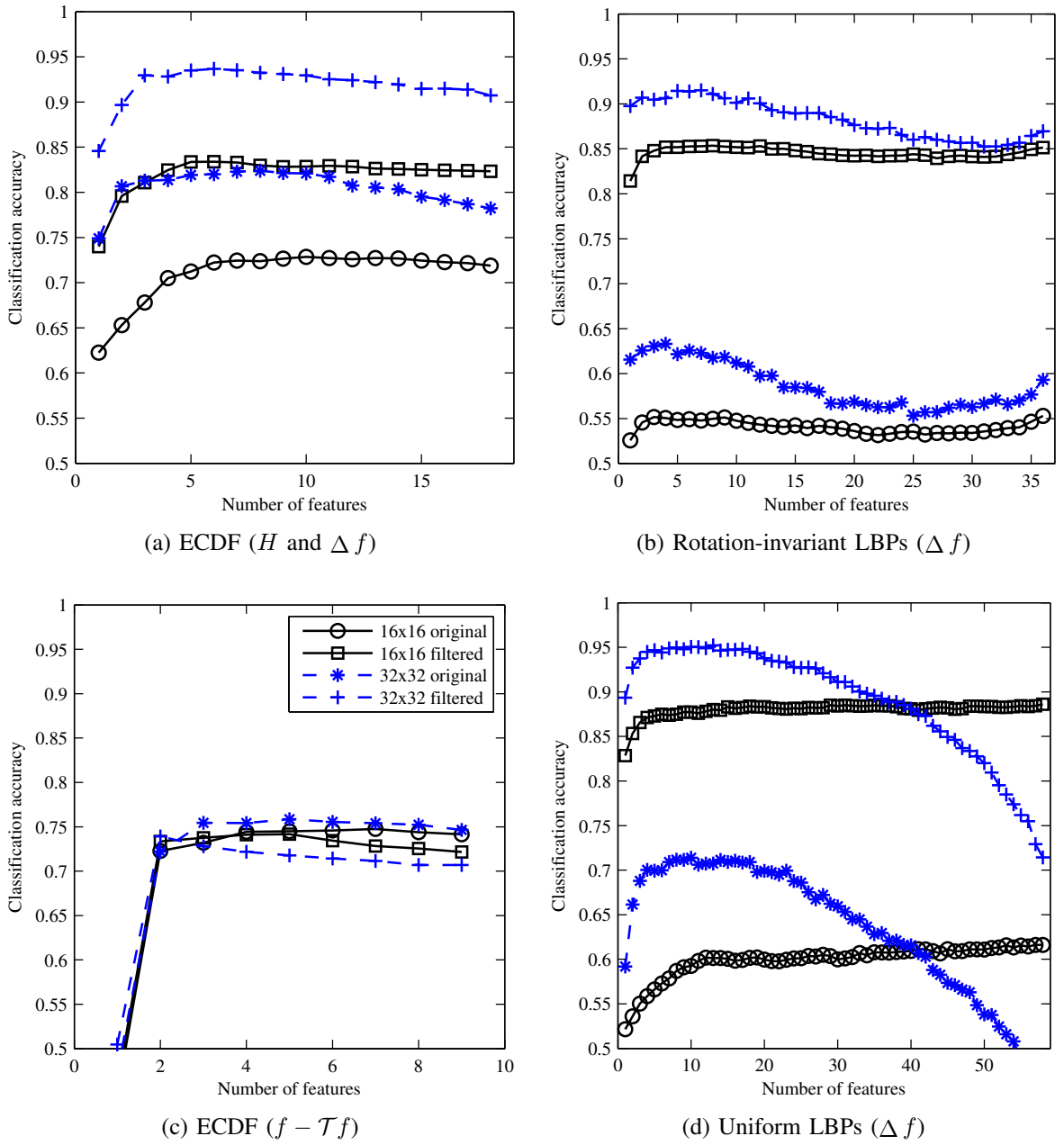
Appendix F contains a more detailed accuracy assessment, including confusion matrices, for classification with invariant geometrical properties and rotation-invariant LBPs, using the optimum number of features.

VI. SUMMARY AND DISCUSSION

The principal aim of this work has been to reduce stripe noise and motion-induced noise in remote sensing data, particularly in MBES bathymetry as a step towards more robust classification of seabed type. This paper has four subthemes: inversion of the GDB (DRT) transform, construction of a linear invertible edge detection operator, noise signature reduction in the GDB transform domain, and finally, the effect of noise reduction on the classification of geomorphology by invariant terrain features.

As an image-transform-based method, the proposed algorithm shares some of the strengths and weaknesses of Fourier domain filtering, but the noise characteristics are very different in the DRT domain. The two first steps of the algorithm, edge detection plus forward DRT, closely parallels a standard method for straight line detection in image analysis, namely to apply a gradient operator, thresholding, and the Hough transform [25] (see, e.g., [60] concerning the relationship between the Hough and Radon transforms). A straight line will correspond to a high intensity spot in the DRT domain irrespective of orientation or other image features. When lines are parallel, as in the present data and much other remote sensing imagery, the spots will align in a single vertical column in the global DRT plot (Fig. 6), but this is merely convenient, not essential. In general one could apply a computational method for finding the high intensity peaks, analogous to the `houghpeaks` function in the Matlab®Image Processing Toolbox [61]. The corresponding stripes may subsequently be suppressed by zeroing the DRT in a small window centered on each peak. The weak point in this approach is the risk of suppressing real linear features in the image, as exemplified by the supposed pipeline in Fig. 12. The proposed denoising algorithm works as a smoothing filter which, applied to the present MBES datasets, suppressed noise better than standard

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380



(a) ECDF (H and Δf)

(b) Rotation-invariant LBPs (Δf)

(c) ECDF ($f - \mathcal{T} f$)

(d) Uniform LBPs (Δf)

Fig. 18. Learning curves for the terrain classification experiment (three geomorphological classes), before and after the DRT noise filter was applied. The legend in Fig. (c) applies to all four panels. The classification is based on local distributions (ECDFs and LBP histograms) of geometric quantities, in 16×16 and 32×32 pixels blocks.

low-pass filters and also caused less distortion of topographic features. We also found that it was simpler to design a noise filter in the DRT domain than in the Fourier domain (see Section III-B).

Compared with conventional filters in the spatial or Fourier domains, the full DRT denoising algorithm (see Section III-B) is complicated, with more adjustable parameters: the polynomial degree p and downsampling factor D of the approximation algorithm (see Appendix D), the ϵ-factor and dimension M of the modified Laplacian (23), and a stopping criterion for the iterative DRT inverse (residual error threshold or fixed number of iterations). In addition there is some freedom of choice in how to suppress the high-intensity peaks in the DRT domain.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

However, little fine-tuning was needed to obtain the results presented here; only the degree p needed to be adjusted in some trials to avoid a low-frequency distortion in the restored DTM. The degree p should anyhow not be too large, so as to avoid fitting the approximation to the noise. A factor $D > 1$ both counters this tendency and reduces computer memory requirements. We think, moreover, that the fixed values of $\epsilon = 10^{-3}$ and $M = 7$ will work in other applications as well.

In addition to image size, the stopping criterion is the critical parameter with respect to computational cost. The tolerance r_k should not be smaller than necessary; $r_k \sim 10^{-2}$ or 10^{-3} produced good results for the present data. The DRT of smaller image blocks can be inverted noniteratively, i.e., by computing a pseudoinverse matrix explicitely. This may speed up execution when an image can be processed in disjoint blocks as in the two examples above (see Fig. 15c and 16d). A short discussion of these aspects can be found in Appendix E. Other iterative inverse methods that were not explored in the present work include the class of algebraic reconstruction techniques (ART), a common alternative to filtered backprojection in, e.g., medical imaging [62]. Another possibility we have not tried is to replace the approximate inverse operator $\boldsymbol{B}$ in (13) with filtered backprojection, and then apply GMRES to obtain an iterative refinement.

The classification experiment demonstrates that segmentation or classification based on local distributions of simple, invariant, second-derivative-based quantities can be an effective means for discriminating between geomorphological types. The mean and Gaussian curvatures have previously proven useful, e.g., for terrain recognition in airplane navigation, partly due to their invariant properties [63]. Curvature is, however, sensitive to noise; the Laplacian is after all used precisely as a noise detector in the DRT filter algorithm. Therefore, it is important to filter the data before classification, as Fig. 18 and Table I show. Only small differences were found when performing classification based on K, H, Δf, or a combination of the three. However, the easily interpreted quantities K and H may be useful in some situations. Indeed, it may be possible to construct features for discriminating between specific morphological types. For example, an even sandy seabed (sloping or not) has curvature $K \approx H \approx 0$, but if water currents deform the seabed by forming parallel ripples, then $K \approx 0$ but $|H| \neq 0$.

While the experimental results shown in this paper are for MBES bathymetry only, we have also used the DRT filter to remove scalloping in SAR images of the Earth's surface. Scalloping is an amplitude modulation that causes parallel stripes in burst mode SAR images [64, Ch. 5.3], particularly wide-swath ScanSAR images of ocean scenes. The procedure was the same as the DENOISE algorithm of Section III-B, minus the DRT domain median filter. (Recall that the median filter was only intended to remove motion-induced noise in MBES data.) The SAR tests suggest that the proposed method may be useful in other remote sensing applications. It is also possible to reverse the DRT domain filter so as to accentuate linear features that are oriented in a particular angular sector, i.e. by zeroing the DRT domain outside the sector before applying the inverse transform. When the DRT filter is applied locally, in sliding windows, this method can be useful for detecting thin structures with a predominant orientation, such as seismic fault lines, fractures, or perhaps blood vessels. In such applications we would use the matrix representation of the DRT inverse as described above.

Another possible development is the use of invertible gradient operators, which may be constructed in the same way as the modified Laplacian, i.e by using convolution with (23) but based instead on the PSFs of, e.g., the Sobel operators. Finally, we note that, in some applications, inversion of the GDB transform can be sped up, and memory consumption

lowered, by using the fact that each quadrant map $\mathcal{R}^i$ is injective, implying that each of the four quadrants of the GDB transform, $\mathcal{R}^1\boldsymbol{f}, \ldots, \mathcal{R}^4\boldsymbol{f}$, contains all the information needed to reconstruct $\boldsymbol{f}$ (see Appendix A).

APPENDIX A
INJECTIVITY OF $\mathcal{R}$

The GDB transform $\mathcal{R}$ is injective (one-to-one) if and only if $\mathcal{R}\boldsymbol{f} = \mathbf{0}$ implies that $\boldsymbol{f} = \mathbf{0}$. Suppose that $\boldsymbol{f} \in \mathbb{R}^{2n\times 2n}$, where $n = 2^k$ and $k \in \{0, 1, 2, \ldots\}$. The matrix (image) $\boldsymbol{f}$ is divided into a left half $\boldsymbol{f}^{(L)}$ and a right half $\boldsymbol{f}^{(R)}$, each of size $2n \times n$. We let x_{hs} and y_{hs} denote, in $\boldsymbol{f}^{(L)}$ and $\boldsymbol{f}^{(R)}$ respectively, the sum of pixel values on the graph $D_n(h, s)$, i.e.

$$x_{hs} = \sum_{(i,j)\in D_n(h,s)} f_{ij}^{(L)} \tag{28a}$$

$$y_{hs} = \sum_{(i,j)\in D_n(h,s)} f_{ij}^{(R)}, \tag{28b}$$

where $-2n < h < 2n$ and $0 \le s < n$. Thus $x_{00} + y_{00}$ is the sum of the bottom row, $x_{0,n-1} + y_{n,n-1}$ is the sum of the anti-diagonal, and so on. The zero-padded matrix is

$$\boldsymbol{f}_0 = \left[\begin{array}{ccc|ccc} \multicolumn{3}{c|}{\mathbf{0}_{(2n-1)\times n}} & \multicolumn{3}{c}{\mathbf{0}_{(2n-1)\times n}} \\ f_{11} & \cdots & f_{1n} & f_{1,n+1} & \cdots & f_{1,2n} \\ \vdots & \overset{x_{n,n-1}}{\nearrow} & \vdots & \vdots & \overset{y_{n,n-1}}{\nearrow} & \vdots \\ f_{n1} & & f_{nn} & f_{n,n+1} & & f_{n,2n} \\ f_{n+1,1} & & f_{n+1,n} & f_{n+1,n+1} & & f_{n+1,2n} \\ \vdots & \overset{x_{0,n-1}}{\nearrow} & \vdots & \vdots & \overset{y_{0,n-1}}{\nearrow} & \vdots \\ f_{2n,1} & \overset{x_{00}}{\longrightarrow} & f_{2n,n} & f_{2n,n+1} & \overset{y_{00}}{\longrightarrow} & f_{2n,2n} \\ \multicolumn{3}{c|}{\mathbf{0}_{(2n-1)\times n}} & \multicolumn{3}{c}{\mathbf{0}_{(2n-1)\times n}} \end{array}\right].$$

By construction (4), $\mathcal{R}^1\boldsymbol{f} = 0$ implies that

$$x_{hs} + y_{h+s,s} = x_{hs} + y_{h+s+1,s} = 0. \tag{29}$$

It follows that

$$y_{-2n+1,s} = y_{-2n+2,s} = \ldots = y_{0s} = \ldots = y_{2n-1,s}, \tag{30}$$

i.e. y_{hs} must be independent of h. Since $y_{0s} = -x_{-s-1,s} = 0$, all y_{hs} are zero. Consequently all x_{hs} are also zero by the assumption $\mathcal{R}^1\boldsymbol{f} = 0$. By (4), the complete graphs $D_{2^{k+1}}(h, s)$ are constructed by recursively combining the graphs D_{2^j} of pairs of adjacent subimages; the subimage size is $2^{k+1} \times 2^j$, for $j = 0, \ldots, k$. The above argument may now be applied to any pair of graphs $D_{2^j}^{(L)}$ and $D_{2^j}^{(R)}$. Hence if x_{hs} and y_{hs} are defined as in (28), with $n = 2^j$, and $x_{hs} + y_{hs} = 0$, then $x_{hs} = 0$ and $y_{hs} = 0$. By induction which terminates when $j = 1$, it follows that $\boldsymbol{f} = 0$.

The same conclusion is reached assuming that $\mathcal{R}^i\boldsymbol{f} = 0$ for $i = 2, 3$, or 4. Each quadrant map $\mathcal{R}^i$ is injective, and each $\mathcal{R}^i\boldsymbol{f}$ contains all information needed to reconstruct $\boldsymbol{f}$.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

APPENDIX B
DECOMPOSITION OF THE DRT DOMAIN

The recursive backprojection algorithm of the GDB transform is such that, in the backprojected image, each pixel p is the sum of all DRT points $(\mathcal{R}^i)(h,s)$ whose associated graphs contain p [32]. From the linear algebra viewpoint, the GDB backprojection algorithm is the transpose $\boldsymbol{R}^T$ of the forward transform. To see this, consider that $\boldsymbol{R}^T$ maps between the vector spaces

$$\boldsymbol{R}^T : \mathbb{R}^P \to \mathbb{R}^Q,$$

where $P = 6N^2 - 2N$ and $Q = N^2$ as before. The two spaces have orthonormal basis vectors $\boldsymbol{g}_k$, $k = 1, \ldots, P$ and $\boldsymbol{e}_k$, $k = 1, \ldots, Q$ with unity in the kth component and zeros elsewhere. Let $\Pi_j \subset \{1, 2, \ldots, Q\}$ denote the set of pixels included in the jth graph ($j = 1, \ldots, P$). The components of $\boldsymbol{R}^T$ are

$$\begin{aligned}\left(\boldsymbol{R}^T\right)_{ij} &= \boldsymbol{e}_i^T \boldsymbol{R}^T \boldsymbol{g}_j = \boldsymbol{g}_j^T \boldsymbol{R} \boldsymbol{e}_i \\ &= \begin{cases} 1 & \text{if } i \in \Pi_j \\ 0 & \text{otherwise.} \end{cases}\end{aligned}$$

Hence, for any $\boldsymbol{G} = \sum_{j=1}^{P} G_j \boldsymbol{g}_j \in \mathbb{R}^P$,

$$\left(\boldsymbol{R}^T \boldsymbol{G}\right)_i = \sum_{j=1}^{P} \left(\boldsymbol{R}^T\right)_{ij} G_j = \sum_{j: i \in \Pi_j} G_j,$$

which is the ith pixel of the backprojection of $\boldsymbol{G}$.

The vector space $\mathbb{R}^P$ can be decomposed as the direct sum of the range of $\mathcal{R}$ and its orthogonal complement, $\mathbb{R}^P = \mathcal{I}(\boldsymbol{R}) \oplus \mathcal{I}(\boldsymbol{R})^\perp$. Moreover, $\mathcal{I}(\boldsymbol{R})^\perp$ can be identified with the null space of the Hermitian adjoint $\boldsymbol{R}^+$, and $\boldsymbol{R}^+ = \boldsymbol{R}^T$ since $\boldsymbol{R}$ is real [39]. Thus, $\mathbb{R}^P$ is the orthogonal direct sum

$$\mathbb{R}^P = \mathcal{I}(\mathcal{R}) \oplus \mathcal{N}(\mathcal{R}^+), \tag{31}$$

where $\mathcal{R}^+$ as in the continuous case denotes the backprojection operator/algorithm.

APPENDIX C
ANALYSIS OF THE PRESS INVERSE

The approximate inverse algorithm of [33], $\mathcal{B}$, can be represented by a $Q \times P$ matrix $\boldsymbol{B}$, and the product $\boldsymbol{BR}$ is square ($Q \times Q$). Provided $\boldsymbol{BR}$ is nonsingular, and by defining the residual error matrix $\boldsymbol{E} = \boldsymbol{1}_Q - \boldsymbol{BR}$, the identity $(\boldsymbol{BR})^{-1}\boldsymbol{BR} = \boldsymbol{1}_Q$ may be written

$$\left(\boldsymbol{1}_Q - \boldsymbol{E}\right)^{-1} \boldsymbol{BR} = \boldsymbol{1}_Q, \tag{32}$$

and $\left(\boldsymbol{1}_Q - \boldsymbol{E}\right)^{-1} \boldsymbol{B}$ is seen to be a left-inverse of $\boldsymbol{R}$. If the spectral radius of $\boldsymbol{E}$ (largest eigenvalue magnitude) is $\rho(\boldsymbol{E}) < 1$, then [40, Ch. 7.10]

$$\left(\boldsymbol{1}_Q - \boldsymbol{E}\right)^{-1} = \sum_{k=0}^{\infty} \boldsymbol{E}^k,$$

(a Neumann series) and the left-inverse becomes

$$\boldsymbol{R}_L^{-1} \equiv \left(\boldsymbol{1}_Q + \boldsymbol{E} + \boldsymbol{E}^2 + \ldots\right) \boldsymbol{B}. \tag{33}$$

TABLE II
SPECTRAL RADIUS OF $\boldsymbol{E}$

N	64	128	256	512	1024	2048
ρ	0.838	0.895	0.931	1.027	1.293	1.562
$\log_{10}\sigma$	-14.6	-14.5	-13.9	-14.4	-14.4	-14.4

From (33) follows the recurrence relation for the iteratively inverse DRT of [33], namely

$$\boldsymbol{f}_0 = \boldsymbol{B}\boldsymbol{d} \tag{34a}$$

$$\boldsymbol{f}_{k+1} \equiv \left(\sum_{j=0}^{k+1} \boldsymbol{E}^j\right) \boldsymbol{B}\boldsymbol{d} = \boldsymbol{f}_k + \boldsymbol{B}\left(\boldsymbol{d} - \boldsymbol{R}\boldsymbol{f}_k\right). \tag{34b}$$

This scheme involves only matrix-vector multiplications which are implemented by the fast, recursive algorithms $\mathcal{R}$ and $\mathcal{B}$. There is no need to compute explicitly the matrices $\boldsymbol{B}$ and $\boldsymbol{E}^j$, which is practically impossible for large images.

Although the iterative inverse (34) works very well for test images (Fig. 3) and random images [33], numerical analysis suggests that the condition $\rho(\boldsymbol{E}) < 1$ does not hold for image sizes $N \geq 512$. The spectral radius of $\boldsymbol{E}$, estimated using Arnoldi iteration [65]–[67], is shown in TABLE II. The Press approximate inverse algorithm involves a high-pass filter which we have here implemented using symmetric boundary conditions. Using other boundary conditions (zero-padding, periodicity, or replication of nearest value) results in slightly different values but does not alter the conclusion. The eigenpairs $(\lambda, \boldsymbol{g})$ estimated in this way are good approximations of true eigenpairs at the outer edge of the spectrum; for $(\lambda, \boldsymbol{g})$ corresponding to ρ in TABLE II, the residual is $\sigma \equiv \|\boldsymbol{E}\boldsymbol{g} - \lambda\boldsymbol{g}\|_2 \sim 10^{-14}$ (in operator 2-norm), which is close to machine precision. This means that the Neumann series $\sum_{j=0}^{\infty} \boldsymbol{E}^j$ does not converge, but the inverse $(\boldsymbol{1}_Q - \boldsymbol{E})^{-1} = (\boldsymbol{B}\boldsymbol{R})^{-1}$ still exists provided only that $\lambda = 1$ is not an eigenvalue of $\boldsymbol{E}$. If $\boldsymbol{E}$ has s distinct eigenvalues $\lambda_1, \ldots, \lambda_s$, Jordan decomposition gives $\boldsymbol{E} = \boldsymbol{C}\,\mathrm{diag}\,(\boldsymbol{J}_{\lambda_1,1}, \ldots, \boldsymbol{J}_{\lambda_s,t_s})\boldsymbol{C}^{-1}$, where $\boldsymbol{C}$ is a non-singular $Q \times Q$ matrix and there are $t_j = \dim_{\mathbb{C}} \mathcal{N}(\boldsymbol{E} - \lambda_j \boldsymbol{I}_Q)$ Jordan blocks for eigenvalue λ_j. The approximation error for the kth iterate in (34) may then be written

$$\begin{aligned} \boldsymbol{f} - \boldsymbol{f}_k &= \boldsymbol{f} - (\boldsymbol{1}_Q - \boldsymbol{E})^{-1}\left(\boldsymbol{1}_Q - \boldsymbol{E}^{k+1}\right)\boldsymbol{B}\boldsymbol{d} \\ &= \boldsymbol{P}^{-1}\,\mathrm{diag}\left(\boldsymbol{J}_{\lambda_1,1}^{k+1}, \ldots, \boldsymbol{J}_{\lambda_s,t_s}^{k+1}\right)\boldsymbol{P}\boldsymbol{f}, \end{aligned} \tag{35}$$

where $\boldsymbol{P} \equiv \boldsymbol{C}^{-1}\boldsymbol{B}\boldsymbol{R} \approx \boldsymbol{C}^{-1}$. For $|\lambda| < 1$, $\lim_{k\to\infty} \boldsymbol{J}_{\lambda,i}^k = \boldsymbol{0}$ [40, Ch. 7.10], and in fact the spectrum of $\boldsymbol{E}$ is concentrated in the range $0 < |\lambda| < 1$. For $N = 512$, only four eigenvalues found by Arnoldi iteration have magnitude $|\lambda| > 1$. With respect to a basis obtained by the coordinate transformation matrix $\boldsymbol{P}^{-1}$, only the relatively few components of $\boldsymbol{f}$ corresponding to Jordan blocks with $|\lambda| \geq 1$ will not be correctly reconstructed in the limit $k \to \infty$, and the RMS error will be small. However, the amplification of the blocks $\boldsymbol{J}_{\lambda,*}^k$ for $|\lambda| > 1$ destroys convergence for sufficiently many iterations k. The reconstruction algorithm (34) will therefore not work in all instances. This is demonstrated in Fig. 4 using an eigenimage of $\boldsymbol{E}$ for $N = 512$ with $|\lambda| = \rho = 1.027$.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

APPENDIX D
TREND COMPUTATION

This appendix summarizes the approximation algorithm $\mathcal{T}(p, D)$ (cf. Section III-B). The image/DTM $\boldsymbol{f} \in \mathbb{R}^{N\times N}$ is downsampled to size $N/D \times N/D$, and interpolated back to size $N \times N$ using a Chebyshev polynomial tensor product basis of total degree p. Since N is a power of 2, D is also conveniently taken to be a power of 2 ($1 \leq D < N$). We let $\{\, T_k(x) \,\}_{k=0}^{\infty}$ denote the set of Chebyshev polynomials of the first kind on the interval $[-1, 1]$ [68, Ch. 22]. Treating $\boldsymbol{f}$ as a set of discrete samples of a function $f(x, y)$ on the square $[-1, 1] \times [-1, 1]$, interpolation is accomplished via the expansion

$$f(x, y) \approx \sum_{0\leq j+k\leq p} w_{jk} T_j(x) T_k(y). \tag{36}$$

There are $\eta_p = \sum_{d=0}^{p} \binom{d+1}{d} = (p + 2)(p + 1)/2$ terms in this series. The coefficients w_{jk} are determined by a least squares fit of $N/D \times N/D$ samples. This can be implemented by multiplication with a generalized matrix inverse P of size $\eta_p \times (N/D)^2$, which may be precomputed (i.e. need only be computed once). The Chebyshev polynomials $T_0(x), \ldots, T_p(x)$ are also precomputed at N evenly spaced points on the interval $[-1, 1]$, and stored in a matrix $\boldsymbol{C} \in \mathbb{R}^{N\times(p+1)}$. The latter is accomplished by the following Matlab function given in Listing 1.

Listing 1. Compute Chebyshev polynomials (Matlab)

```
function C = chebpol(N,p)
x = -1:2/(N-1):1;
C = zeros(N,p+1);
C(:,1) = ones(N,1);
if p > 1
  C(:,2) = x';
end
if p > 2
  for k = 3:p+1
    C(:,k) = 2*x'.*C(:,k-1)-C(:,k-2);
  end
end
```

The following code fragment given in Listing 2 generates the Chebyshev polynomials $\boldsymbol{C}$ and the generalized inverse $\boldsymbol{P}$ for fixed N, p, and D.

Listing 2. Compute Chebyshev polynomials and the generalized inverse (Matlab)

```
M = N/D;
Ep = (p+1)*(p+2)/2;
C = chebpol(M,p);
L = single(zeros(M, M, Ep));
[j,k] = meshgrid(0:p,0:p);
for n = [find(j+k<=p)';1:Ep]
  L(:,:,n(2)) = ...
  C(:,j(n(1))+1).*C(:,k(n(1))+1)';
end
PI = pinv(reshape(L, M^2, Ep));
```

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

```
if D ~= 1
  C = chebpol(N,p);
end
```

If $\boldsymbol{g} \in \mathbb{R}^{N/D \times N/D}$ is the downsampled image, then the coefficients w_{jk} and trend $\boldsymbol{t} \in \mathbb{R}^{N \times N}$ may be obtained in Listing 3.

Listing 3. Compute approximate image by Chebyshev interpolation (Matlab)

```
w = PI*g(:);
t = zeros(N);
for n = [find(j+k<=p)';1:Ep]
  t = t + w(n(2)) * C(:,j(n(1))+1)...
        * C(:,k(n(1))+1)';
end
t = double(t);
```

Depending on N/D and p, this approach may require too much memory to compute and store the $(p+2)(p+1)/2 \times (N/D)^2$ matrix P. This simple solution therefore cannot be considered general, but has nevertheless worked well in our applications (both MBES and SAR) for image sizes up to $N = 2048$ (also with $D = 1$ and $p \sim 10$).

APPENDIX E
NOTE ON COMPUTING TIME

The stopping criterion influences the execution time of the DRT inverse, and the tolerance r_k (relative residual norm, Section II-C4) should not be set smaller than necessary. The code for this paper was written in C as Matlab executable (MEX) files and ran on a single 2.93 GHz CPU (core) on a Linux computer with 8 GB RAM. Less than five iterations with GMRES usually sufficed to obtain a good result, and for image size $N = 1024$ ($\sim 10^6$ unknowns and $\sim 6 \cdot 10^6$ equations) good results were obtained in about 20 s execution time, although the rate of convergence may depend on the data.

The results shown in Fig. 15c and Fig. 16d suggest that images can be processed in disjoint blocks, and program execution may then be considerably faster. The main reason is that an explicit DRT pseudoinverse matrix may be precomputed; inversion of each block is then carried out non-iteratively, by matrix multiplication, yet with arbitrarily low tolerance r_k. So for small block sizes ($N \leq 64$) we have computed the matrix representations of the three iterative algorithms of Section II-C, which is not feasible for large images due to the memory requirement. The solution to the normal equations (7) can be obtained by multiplication with the Moore-Penrose pseudoinverse. The cost of inversion by matrix multiplication is independent of the accuracy with which the matrix representations of the Press and GMRES inverses are computed (and the Press inverse is an exact left-inverse for $N < 512$). The cost of obtaining the (pseudo-)inverse matrices does depend on the accuracy, but this is a one-time-only operation.

The time for processing the $(1024/32)^2$ blocks of the DTM shown in Fig. 16d was about 9.1 s (only one quadrant of the image is shown in the figure). This includes not just the time to invert the block DRTs, but also the time to compute the forward DRTs and apply the DRT domain noise suppression filter. In this example the forward DRT was also computed using a matrix representation, i.e., of the GDB algorithm. The number of additions in the GDB

TABLE III
ACCURACY ASSESSMENT: ECDF (H AND Δf)

Class	Assignments			Total	Omiss-ions	Comm-issions	MA
	#1	#2	#2				
#1	50.1	9.5	3.4	63	20.5%	21.7%	65.4%
#2	9.3	55.1	5.6	70	21.4%	16.4%	67.7%
#3	4.4	2.0	52.6	59	10.8%	15.3%	77.5%

(a) Original image

Class	Assignments			Total	Omiss-ions	Comm-issions	MA
	#1	#2	#2				
#1	57.5	4.1	1.4	63	8.7%	4.7%	87.2%
#2	2.9	64.2	2.9	70	8.3%	7.0%	85.7%
#3	0.1	0.8	58.1	59	1.4%	7.3%	91.9%

(b) DRT-filtered image

algorithm is $4N^2 \log_2 N$, which for large N is much lower than for conventional $\mathcal{O}(N^3)$ DRTs (e.g., discrete approximations to the classical Radon transform). For $N = 128$, we have $4N^2 \log_2 N = 458752$, while there are 8388608 non-zero elements (ones) in the matrix representation of the GDB algorithm, which gives a ratio of $1 : 18$. For $N \leq 64$, however, matrix multiplication using an efficient low-overhead library may be faster than the recursive algorithm.

APPENDIX F
CLASSIFICATION ACCURACY ASSESSMENT

This appendix contains supplementary data about the classification experiments, cf. Section V-C and Table I. Table III shows the accuracy assessment for classification with ECDFs applied to invariant geometrical properties (minimum curvature and Laplacian combined) in 32×32 pixels blocks. The six best features were employed, obtained using forward feature selection, cf. Fig. 18. The experiment, including feature selection, was run 20 times, and Table III shows the average values. Hence, the values for assignments are also fractional. The mapping accuracy (MA) for a class ω is defined as

$$\text{MA} = \frac{N_{\text{cor.}}(\omega)}{N_{\text{cor.}}(\omega) + N_{\text{omi.}}(\omega) + N_{\text{com.}}(\omega)}, \tag{37}$$

where $N_{\text{cor.}}(\omega)$ is the number of correctly classified observations, and $N_{\text{omi.}}(\omega)$ and $N_{\text{com.}}(\omega)$ are the number of omissions and commissions, respectively, in class ω.

Table IV shows the corresponding results for classification with rotation-invariant LBPs in 32×32 pixels blocks using the five best features.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380

TABLE IV
ACCURACY ASSESSMENT: ROTATION-INVARIANT LBPS (Δf)

Class	Assignments			Total	Omiss-ions	Comm-issions	MA
	#1	#2	#2				
#1	28.4	16.4	18.2	63	54.9%	43.8%	31.4%
#2	15.7	49.5	4.8	70	29.4%	29.1%	54.8%
#3	11.9	4.0	43.1	59	26.9%	39.1%	52.6%

(a) Original image

Class	Assignments			Total	Omiss-ions	Comm-issions	MA
	#1	#2	#2				
#1	63.0	0	0	63	0%	0.1%	99.9%
#2	0.1	61.5	8.4	70	12.1%	12.7%	78.1%
#3	0.0	8.9	50.1	59	15.1%	14.3%	74.4%

(b) DRT-filtered image

ACKNOWLEDGMENT

The authors would like to thank the Norwegian Defence Research Establishment (FFI) for permission to use the data, especially the hydrography group and its supervisor A. Karlsen, who collected and made the data available.

REFERENCES

[1] M. di Bisceglie, R. Episcopo, C. Galdi, and S. Liberata, "Destriping MODIS data using overlapping field-of-view method," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 2, pp. 637–651, Feb. 2009.

[2] M. Bouali and S. Ladjal, "Toward optimal destriping of MODIS data using a unidirectional variational model," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 2924–2935, Aug. 2011.

[3] H. Carfantan and J. Idier, "Statistical linear destriping of satellite-based pushbroom-type images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 4, pp. 1860–1871, Apr. 2010.

[4] X. Lurton, *An Introduction to Underwater Acoustics: Principles and Applications*, 2nd ed. Berlin, Heidelberg: Springer, 2010.

[5] J. E. Hughes Clarke, L. A. Mayer, and D. E. Wells, "Shallow-water imaging multibeam sonars: A new tool for investigating seafloor processes in the coastal zone and on the continental shelf," *Marine Geophysical Researches*, vol. 18, no. 6, pp. 607–629, Dec. 1996.

[6] M. J. Smith and C. D. Clark, "Methods for the visualization of digital elevation models for landform mapping," *Earth Surface Processes and Landforms*, vol. 30, no. 7, pp. 885–900, Jul. 2005.

[7] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 269–285, Apr. 1976.

[8] U. C. Herzfeld and C. A. Higginson, "Automated geostatistical seafloor classification—principles, parameters, feature vectors, and discrimination criteria," *Computers & Geosciences*, vol. 22, no. 1, pp. 35–52, Feb. 1996.

[9] M. F. J. Wilson, B. O'Connel, C. Brown, J. C. Guinan, and A. J. Grehan, "Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope," *Marine Geodesy*, vol. 30, no. 1-2, pp. 3–35, May 2007.

[10] J. E. Hughes Clarke, "Dynamic motion residuals in swath sonar data: Ironing out the creases," *Int. Hydrographic Review*, vol. 4, no. 1, pp. 6–23, 2003.

[11] L. Hellequin, J.-M. Boucher, and X. Lurton, "Processing of high-frequency multibeam echo sounder data for seafloor characterization," *IEEE J. Ocean. Eng.*, vol. 28, no. 1, pp. 78–89, Jan. 2003.

[12] R. E. Crippen, "A simple spatial filtering routine for the cosmetic removal of scan-line noise from Landsat TM P-tape imagery," *Photogramm. Eng. Remote Sens.*, vol. 55, no. 3, pp. 327–331, Mar. 1989.
[13] D. Helder, B. Quirk, and J. Hood, "A technique for the reduction of banding in Landsat thematic mapper images," *Photogramm. Eng. Remote Sens.*, vol. 58, no. 10, pp. 1425–1431, Oct. 1992.
[14] J.-J. Pan and C.-I. Chang, "Destriping of Landsat MSS images by filtering techniques," *Photogramm. Eng. Remote Sens.*, vol. 58, no. 10, pp. 1417–1423, Oct. 1992.
[15] V. R. Algazi and G. E. Ford, "Radiometric equalization of nonperiodic striping in satellite data," *Comput. Graph. Image Process.*, vol. 16, no. 3, pp. 287–295, 1981.
[16] G. Corsini, M. Diani, and T. Walzel, "Striping removal in MOS-B data," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1439–1446, May 2000.
[17] B. K. P. Horn and R. J. Woodham, "Destriping Landsat MSS images by histogram modification," *Comput. Graph. Image Process.*, vol. 10, no. 1, pp. 69–83, 1979.
[18] M. Wegener, "Destriping multiple detector imagery by improved histogram matching," *Int. J. Remote Sens.*, vol. 11, no. 5, pp. 859–875, May 1990.
[19] F. L. Gadallah and F. Csillag, "Destriping multidetector imagery with moment matching," *Int. J. Remote Sens.*, vol. 21, no. 12, pp. 2505–2511, Aug. 2000.
[20] H. Shen, W. Jiang, H. Zhang, and L. Zhang, "A piece-wise approach to removing the nonlinear and irregular stripes in MODIS data," *Int. Journal of Remote Sensing*, vol. 35, no. 1, pp. 44–53, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1080/2150704X.2013.860564
[21] P. Rakwatin, W. Takeuchi, and Y. Yasuoka, "Stripe noise reduction in MODIS data by combining histogram matching with facet filter," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1844–1856, Jun. 2007.
[22] H. Shen and L. Zhang, "A MAP-based algorithm for destriping and inpainting of remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 5, pp. 1492–1502, May 2009.
[23] F. Tsai and W. W. Chen, "Striping noise detection and correction of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4122–4131, Dec. 2008.
[24] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
[25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.
[26] D. Wilken, P. Feldens, T. Wunderlich, and C. Heinrich, "Application of 2D Fourier filtering for elimination of stripe noise in side-scan sonar mosaics," *Geo-Marine Lett*, vol. 32, no. 4, pp. 337–347, Aug. 2012.
[27] J. Chen, Y. S. an H. Guo, W. Wang, and B. Zhu, "Destriping CMODIS data by power filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 2119–2124, Sep. 2003.
[28] R. N. Bracewell and A. C. Riddle, "Inversion of fan-beam scans in radio astronomy," *Astrophys. J.*, vol. 150, pp. 427–434, Nov. 1967.
[29] F. Natterer and F. Wübbeling, *Mathematical Methods in Image Reconstruction*. Philadephia, PA: Soc. Ind. Appl. Math., 2001.
[30] G. Beylkin, "Discrete Radon transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 2, pp. 162–172, Feb. 1987.
[31] W. A. Götz and H. J. Druckmüller, "A fast digital Radon transform—an efficient means for evaluating the Hough transform," *Pattern Recognition*, vol. 29, no. 4, pp. 711–718, Apr. 1996.
[32] M. L. Brady, "A fast discrete approximation algorithm for the Radon transform," *SIAM J. Comput.*, vol. 27, no. 1, pp. 107–119, 1998.
[33] W. H. Press, "Discrete Radon transform has an exact, fast inverse and generalizes to operations other than sums along lines," *Proc. Natl. Acad. Sci. USA (PNAS)*, vol. 103, no. 51, pp. 19 249–19 254, Dec. 2006.
[34] K. T. Smith and F. Keinert, "Mathematical foundations of computed tomography," *Appl. Opt.*, vol. 24, no. 23, pp. 3950–3957, Dec. 1985.
[35] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Trans. Nucl. Sci.*, vol. 21, no. 3, pp. 21–43, Jun. 1974.
[36] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.
[37] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM. Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Jun. 1982.
[38] D. C.-L. Fong and M. Saunders, "LSMR: An iterative algorithm for sparse least-squares problems," *SIAM. J. Sci. Comput.*, vol. 33, no. 5, pp. 2950–2971, 2011.
[39] D. G. Luenberger, *Optimization by Vector Space Methods*, ser. Series in decision and control. New York, NY: John Wiley & Sons, 1969.

[40] C. D. Meyer, *Matrix analysis and applied linear algebra*. Philadephia, PA: Soc. Ind. Appl. Math. (SIAM), 2000, ch. 7.10.

[41] I. C. F. Ipsen and C. D. Meyer, "The idea behind Krylov methods," *Amer. Math. Monthly*, vol. 105, no. 10, pp. 889–899, Dec. 1998.

[42] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM. J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 856–869, Jul. 1986.

[43] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, ser. Frontiers in applied mathematics. Philadephia, PA: Soc. Ind. Appl. Math., 1995, vol. 16.

[44] B. Jähne, *Digital Image Processing*, 6th revised and extended ed. Berlin Heidelberg New York: Springer, 2005.

[45] M. Einsiedler and T. Ward. (2015, Mar.) Functional analysis, spectral theory, and applications. Draft version. [Online]. Available: https://dl.dropboxusercontent.com/u/2098511/FAnotes.pdf

[46] M. P. do Carmo, *Differential geometry of curves and surfaces*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

[47] T. F. Chan and J. Shen, *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. Philadelphia, PA: SIAM, 2005.

[48] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[49] L. Wang and D. C. He, "Texture classification using texture spectrum," *Pattern Recognition*, vol. 23, no. 8, pp. 905–910, 1990.

[50] T. Ojala and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions," *Pattern Recognition*, vol. 32, no. 3, pp. 477–486, 1999.

[51] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[52] M. Pietikäinen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," *Pattern Recognition*, vol. 33, no. 1, pp. 43–52, 2000.

[53] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[54] M. Hovland, "Elongated depressions associated with pockmarks in the western slope of the Norwegian Trench," *Marine Geology*, vol. 51, no. 1-2, pp. 35–46, Feb. 1983.

[55] R. Bøe, L. Rise, and D. Ottesen, "Elongate depressions on the southern slope of the Norwegian Trench (Skagerrak): morphology and evolution," *Marine Geology*, vol. 146, no. 1-4, pp. 191–203, Apr. 1998.

[56] G. Dahlquist and Å. Björck, *Numerical Methods*, ser. Dover Books on Mathematics. Dover Publications, 2003.

[57] J. E. Nichol and V. Vohora, "Noise over water surfaces in Landsat TM images," *Int. Journal of Remote Sensing*, vol. 25, no. 11, pp. 2087–2093, Jun. 2004.

[58] H. Schulerud and F. Albregtsen, "Many are called, but few are chosen. feature selection and error estimation in high dimensional spaces," *Comput. Methods Programs Biomed.*, vol. 73, no. 2, pp. 91–99, Feb. 2004.

[59] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proc. Natl. Acad. Sci. USA (PNAS)*, vol. 99, no. 10, pp. 6562–6566, May 2002.

[60] M. van Ginkel, C. L. Luengo Hendriks, and L. J. van Vliet, "A short introduction to the Radon and Hough transforms and how they relate to each other," TU Delft, Quantitative Imaging Group Technical Report Series QI-2004-01, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.9419

[61] HOUGHPEAKS. MathWorks Matlab R2013b Documentation. Image processing toolbox. [Online]. Available: http://www.mathworks.se/help/images/ref/houghpeaks.html

[62] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography," *J. Theoret. Biol.*, vol. 29, no. 3, pp. 471–481, Dec. 1970.

[63] D. B. Goldgof, T. S. Huang, and H. Lee, "A curvature-based approach to terrain recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 11, pp. 1213–1217, Nov. 1989.

[64] G. Francheschetti and R. Lanari, *Synthetic Aperture Radar Processing*. Boca Raton, FL: CRC Press, 1999.

[65] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM. J. Matrix Anal. & Appl.*, vol. 17, no. 4, pp. 789–821, Oct. 1996.

[66] R. B. Lehoucq, D. C. Sorensen, and C. Yang. (1998) ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. Web version available. [Online]. Available: http://www.caam.rice.edu/software/ARPACK/UG/ug.html

[67] EIGS. MathWorks Matlab R2012b Documentation. [Online]. Available: http://www.mathworks.com/help/matlab/ref/eigs.html

[68] U. W. Hochstrasser, "Orthogonal polynomials," in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, M. Abramowitz and I. A. Stegun, Eds. National Bureau of Standards, 1964.

[69] P. Wessel and W. H. F. Smith, "New, improved version of Generic Mapping Tools released," *EOS Trans. AGU*, vol. 79, no. 47, p. 579, Nov. 1998.

DOI til publisert versjon/DOI to published version: 10.1109/TGRS.2015.2436380