Manuscript prepared for J. Name
with version 2015/09/17 7.94 Copernicus papers of the LaTeX class copernicus.cls.
Date: 13 February 2017

Open-source sea ice drift algorithm for Sentinel-1 SAR imagery using a combination of feature-tracking and pattern-matching

Stefan Muckenhuber and Stein Sandven

Nansen Environmental and Remote Sensing Center (NERSC), Thormøhlensgate 47,
5006 Bergen, Norway

Correspondence to: S. Muckenhuber (stefan.muckenhuber@nersc.no)

Abstract. An open-source sea ice drift algorithm for Sentinel-1 SAR imagery is introduced based on the combination of feature-tracking and pattern-matching. Feature-tracking produces an initial drift estimate and limits the search area for the consecutive pattern-matching, that provides small to medium scale drift adjustments and normalised cross coefficient values. The algorithm is designed to combine the two approaches in the most meaningful way in order to benefit from the respective advantages. The main advantages of the considered feature-tracking approach are the computational efficiency and the independence of the vectors in terms of position, lengths, direction and rotation. Pattern-matching on the other side allows better control over vector positioning and resolution. The pre-processing of the Sentinel-1 data has been optimised to retrieve a feature distribution that depends less on SAR backscatter peak values. Applying the algorithm with the recommended parameter setting, sea ice drift retrieval with a vector spacing of 4 km on Sentinel-1 images covering 400 km x 400 km, takes about 4 minutes on a standard 2.7 GHz processor with 8 GB memory. The corresponding recommended patch size for the pattern-matching step, that defines the final resolution of each drift vector is 34×34 pixels (2.7×2.7 km). For validation, calculated drift results from 241 Sentinel-1 image pairs have been compared to buoy GPS data, collected in 2015 between 15th January and 22nd April and covering an area from $80.5°$ N to $83.5°$ N and $12°$ E to $27°$ E. We found a logarithmic normal distribution of the error with a median at 341.9 m. All software requirements necessary for applying the presented sea ice drift algorithm are open-source to ensure free implementation and easy distribution.

1 Introduction

Sea ice drift has a strong impact on sea ice distribution on different temporal and spatial scales. Motion of sea ice due to wind and ocean currents causes convergence and divergence zones, resulting in formation of ridges and opening/closing of leads. On large scales, ice export from the Arctic and Antarctic into lower latitudes, where the ice eventually melts away, contributes to a strong season-

25 ality of total sea ice coverage (IPCC, 2013). Due to a lack of ground stations in sea ice covered areas, satellite remote sensing represents the most important tool for observing sea ice conditions on medium to large scales. Despite the strong impact of sea ice drift and the opportunities given by latest satellite remote sensing techniques, there is a lack of extensive ice drift data sets providing sufficient resolution for estimating sea ice deformation on a spatial scaling of less than 5 km.

30 Our main regions of interest are the ice covered seas around Svalbard and East of Greenland. Characteristic for this area are a large variation of different ice types (Marginal Ice Zone, First Year Ice, Multi Year Ice etc.), a strong seasonality of ice cover and a wide range of drift velocities. Focus was put on the winter/spring period, since the area of interest experiences the highest ice cover during this time of the year.

35 Early work from Nansen (1902) established the rule-of-thumb that sea ice velocity resembles 2 % of the surface wind speed with a drift direction of about 45° to the right (Northern Hemisphere) of the wind. This wind driven explanation can give a rough estimate for instantaneous ice velocities. However, the respective influence of wind and ocean current strongly depends on the temporal and spatial scale. Only about 50 % of the long-term (several months) averaged ice drift in the Arctic can
40 be explained by geostrophic winds, whereas the rest is related to mean ocean circulation. This proportion increases to more than 70 % explained by wind, when considering shorter time scales (days to weeks). The wind fails to explain large-scale ice divergence patterns and its influence decreases towards the coast (Thorndike and Colony, 1982).

Using GPS drift data from the International Arctic Buoy Program (IABP), Rampal et al. (2009)
45 analysed the general circulation of the Arctic sea ice velocity field and found that the fluctuations follow the same diffusive regime as turbulent flows in other geophysical fluids. The monthly mean drift using 12 h displacements was found to be in the order of 0.05 to 0.1 m/s and showed a strong seasonal cycle with minimum in April and maximum in October. The IABP dataset also revealed a positive trend in the mean Arctic sea ice speed of +17 % per decade for winter and +8.5 % for summer
50 considering the time period 1979–2007. This is unlikely to be the consequence of increased external forcing. Instead, the thinning of the ice cover is suggested to decrease the mechanical strength which eventually causes higher speed given a constant external forcing (Rampal et al. , 2009b).

Fram Strait represents the main gate for Arctic ice export and high drift velocities are generally found in this area with direction southward. Based on moored Doppler Current Meters mounted
55 near 79° N 5° W, Widell et al. (2003) found an average southward velocity of 0.16 m/s for the period 1996–2000. Daily averaged values were usually in the range 0–0.5 m/s with very few occasions above 0.5 m/s.

Space-borne Synthetic Aperture Radar (SAR) are delivering systematic acquisitions of sea ice covered oceans since the early 1990's and Kwok et al. (1990) showed that sea ice displacement
60 can be calculated from consecutive SAR scenes. SAR is an active imaging sensor operating in the microwave spectrum and produces data regardless of solar illumination and cloud cover. The geo-

physical processor system from Kwok et al. (1990) has been used to calculate sea ice drift fields in particular over the Western Arctic (depending on SAR coverage) once per week with a spatial resolution of 10-25 km for the time period 1997–2012. This extensive dataset makes use of SAR data from RADARSAT-1 operated by the Canadian Space Agency, and from ENVISAT (Environmental Satellite) ASAR (Advanced Synthetic Aperture Radar) operated by ESA (European Space Agency). A high-resolution sea ice drift algorithm for SAR images from ERS-1 (European Remote-sensing Satellite from ESA) based on pattern-matching was introduced by Thomas et al. (2008), allowing drift calculation up to 400 m resolution. The work on this algorithm has been continued by Hollands and Dierking (2011), who derived sea ice drift from ENVISAT ASAR data. Berg and Eriksson (2014) introduced a hybrid algorithm for sea ice drift retrieval from ENVISAT ASAR data using phase correlation and a feature based matching procedure that is activated if the phase correlation value is below a certain threshold. Komarov and Barber (2014) and Muckenhuber et al. (2016) have evaluated the sea ice drift retrieval performance of dual-polarisation SAR imagery using a combination of phase/cross-correlation and feature-tracking based on corner detetction respectively. Muckenhuber et al. (2016) has shown that feature-tracking provides on average around four times as many vectors using HV polarisation compared to HH polarisation. Making use of Sentinel-1 SAR data, an operational sea ice drift product with 10 km resolution is provided by the Danish Technical University (Pedersen et al. (2015), http://www.seaice.dk/) as part of the Copernicus Marine Environment Monitoring Service (CMEMS, http://marine.copernicus.eu).

After the successful start of the Sentinel-1 mission in early 2014, high-resolution SAR images are delivered for the first time in history within a few hours after acquisition as open-source data to all users. This introduced a new era in SAR Earth observation with great benefits for both scientists and other stack holders. The sea ice covered oceans in the European Arctic Sector represent an important area of interest and with Sentinel-1 having a revisit time of less than one day in the Arctic (ESA, 2012), our area of interest is monitored on a daily basis. Muckenhuber et al. (2016) published an open-source feature-tracking algorithm to derive computationally efficient sea ice drift from Sentinel-1 data. This paper follows up the work from Muckenhuber et al. (2016) and aims to improve the feature-tracking approach by combining it with pattern-matching. Unlike Berg and Eriksson (2014), the feature-tracking step is performed initially and serves as a first guess to limit the computational effort of the pattern-matching step.

Contemporary algorithms for deriving displacement vectors between two consecutive images are based either on feature-tracking or pattern-matching.

Feature-tracking detects distinct patterns (features) in both images and tries to connect similar features in a second step without the need for knowing the locations. This can be done computationally efficient and the resulting vectors are often independent of their neighbours in terms of position, lengths, direction and rotation, which is an important advantage for resolving shear zones, rotation and divergence/convergence zones. The considered feature-tracking approach identifies fea-

tures without taking the position of other features into account and matches features from one image
to the other without taking the drift and rotation information from surrounding vectors into account
(Muckenhuber et al., 2016). However, due to the independent positioning of the features, very close
features may share some pixels and since all vectors from the resolution pyramid are combined, the
feature size varies among the matches, which implies a varying resolution. In addition, the resulting
vector field is not evenly distributed in space and large gaps may occur between densely covered
areas, which can eventually lead to missing a shear or divergence/convergence zone.

Pattern-matching, on the other hand, takes a small template from the first image at the starting
location of the vector and tries to find a match on a larger template from the second image. Despite
a considerable computational effort, this approach is widely used, since it allows to define the vector
positions. For practical reasons, a pyramid approach is generally used to derive high-resolution ice
drift. This speeds up the processing, but limits the independence of neighbouring vectors, since they
depend on a lower resolution estimate (Thomas et al., 2008).

The objective of this paper is to combine the two approaches in the most meaningful way in
order to benefit from the respective advantages. The main advantages of the considered feature-
tracking approach are the computational efficiency and the independence of the vectors in terms
of position, lengths, direction and rotation. Pattern-matching on the other side allows better control
over vector positioning and resolution, which is a necessity for computing divergence, shear and
total deformation.

The presented algorithm, all necessary software requirements (python incl. Nansat, openCV and
SciPy) and the satellite data from Sentinel-1 are open-source. A free and user friendly implementa-
tion shall support an easy distribution of the algorithm among scientists and other stakeholders.

The paper is organised as follows: The used satellite products and buoy data are introduced in
Section 2. The algorithm description including data pre-processing is given in Section 3, together
with tuning and validation methods. Section 4 presents the pre-processing, parameter tuning and
validation results and provides a recommended parameter setting. The discussion including outlook
can be found in Section 5.

2   Data

The Sentinel-1 mission is a joint initiative of the European Commission and the European Space
Agency (ESA) and represents the Radar Observatory for the Copernicus Programme, a European
system for monitoring the Earth with respect to environmental and security issues. The mission
includes two identical satellites, Sentinel-1A (launched in April 2014) and Sentinel-1B (launched
in April 2016), each carrying a single C-band SAR with a centre frequency of 5.405 GHz and
dual-polarisation support (HH+HV, VV+VH) also for wide swath mode. Both satellites fly in the
same near-polar, sun-synchronous orbit and the revisit time is less than 1 day in the Arctic (ESA,

2012). The main acquisition mode of Sentinel-1 over sea ice covered areas is Extra Wide Swath

135 Mode Ground Range Detected with Medium Resolution (EW GRDM) and the presented algorithm is built for processing this data type. The covered area per image is $400\,\mathrm{km} \times 400\,\mathrm{km}$ and the data are provided with a pixel spacing of $40\,\mathrm{m} \times 40\,\mathrm{m}$ in both HH and HV polarisation. The introduced algorithm can utilise both HH and HV channel. However, the focus of this paper is put on using HV polarisation, since this channel provides on average four times more feature tracking vectors

140 than HH (Muckenhuber et al., 2016), representing a better initial drift estimate for the combined algorithm.

To illustrate the algorithm performance and explain the individual steps, we use an image pair acquired over Fram Strait. The acquisition times of the two consecutive images are 2015-03-28

145 07:44:33 (UTC) and 2015-03-29 16:34:52 (UTC), and the covered area is shown in Figure 3. This image pair covers a wide range of different ice conditions (multiyear ice, first-year ice, marginal ice zone etc.) and the ice situation is representative for our area and time period of interest.

To evaluate suitable search limitations and validate the algorithm, we use GPS data from drift

150 buoys that have been set out in the ice covered waters north of Svalbard as part of the Norwegian Young Sea Ice Cruise (N-ICE2015) project of the Norwegian Polar Institute (Spreen and Itkin, 2015). The ice conditions during the N-ICE2015 expedition are describe on the project website (http://www.npolar.no/en/projects/n-ice2015.html) as challenging. The observed ice pack, mainly consisting of 1.3-1.5 m thick multiyear and first-year ice, drifted faster than expected and was very

155 dynamic. Closer to the ice edge, break up of ice floes has been observed due to rapid ice drift and the research camp had to be evacuated and re-established four times. This represents a good study field, since these challenging conditions are expected in our area and time period of interest. The considered GPS data have been collected in 2015 between 15th January and 22nd April, and cover an area ranging from 80.5° N to 83.5° N and 12° E to 27° E. The buoys recorded their positions either hourly

160 or every three hours. In the later case, the positions have been interpolated for each hour.

3 Method

3.1 Data pre-processing

To process Sentinel-1 images within Python (extraction of backscatter values and corresponding geolocations, reprojection, resolution reduction etc.), we use the open-source software Nansat (Ko-

165 rosov et al., 2016). Nansat is a scientist-friendly Python toolbox for 2-D satellite Earth observation data, and builds on the Geospatial Data Abstraction Library (http://www.gdal.org). As done in Muckenhuber et al. (2016), we change the projection of the the provided ground control points (latitude/longitude values given for certain pixel/line coordinates) to stereographic and use spline

interpolation to calculate geographic coordinates. This provides a good geolocation accuracy also at
170   high latitudes. The pixel spacing of the image is changed by averaging from 40 m to 80 m, which
is closer to the sensor resolution of 93 m range $\times$ 87 m azimuth, and decreases the computational
effort.

For each pixel p, the Sentinel-1 data file provides a digital number DN_p and a normalisation coef-
ficient A_p, from which the normalised radar cross section σ^0_{raw} is derived by the following equation:

175

$$\sigma^0_{\text{raw}} = DN_p^2/A_p^2 \tag{1}$$

The normalised radar cross section σ^0_{raw} reveals a logarithmic distribution and the structures in the
sea ice are mainly represented in the low and medium backscatter values rather than in the highlights.
Therefore, we change the linear scaling of the raw backscatter values σ^0_{raw} to a logarithmic scaling
180   and get the backscatter values $\sigma^0 = 10 * \lg(\sigma^0_{\text{raw}})$ [dB]. A representative backscatter distribution over
sea ice is shown in Figure 1. Using a logarithmic scaling provides a keypoint distribution for the
feature tracking algorithm that depends less on high peak values, while the total number of vectors
increases.
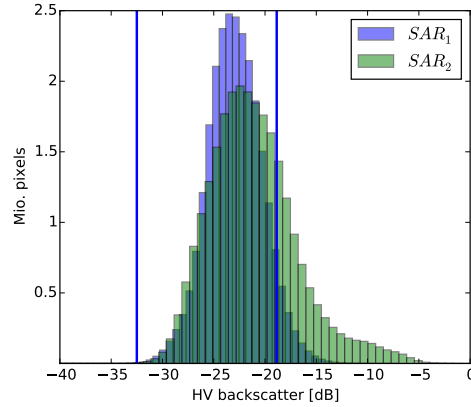
Figure 1. Histogram of HV backscatter values σ^0 from image pair Fram Strait. The lower and upper brightness
boundaries $\sigma^0_{\text{min}} = -32.5$ dB and $\sigma^0_{\text{max}} = -18.86$ dB are shown with blue lines and illustrate the domain for
the intensity values i.

To apply the feature-tracking algorithm from Muckenhuber et al. (2016), the SAR backscatter
185   values σ^0 have to be converted into intensity values i with $0 \leq i \leq 255$ for $i \in \mathbb{R}$. This conversion is
done by using Eq. (2) and setting all values outside the domain to 0 and 255.

$$i = 255 \cdot \frac{\sigma^0 - \sigma^0_{\text{min}}}{\sigma^0_{\text{max}} - \sigma^0_{\text{min}}} \tag{2}$$

The upper brightness boundary $\sigma^0_{\max}$ is set according to the recommended value from Muckenhuber et al. (2016), i.e. -18.86 dB and -10.97 dB for HV and HH respectively. The lower boundary $\sigma^0_{\min}$ was chosen to be -32.5 dB (HV) and -25 dB (HH), since this was found to be a reasonable range of expected backscatter values. Figure 2 shows the image pair Fram Strait after the conversion into intensity values.
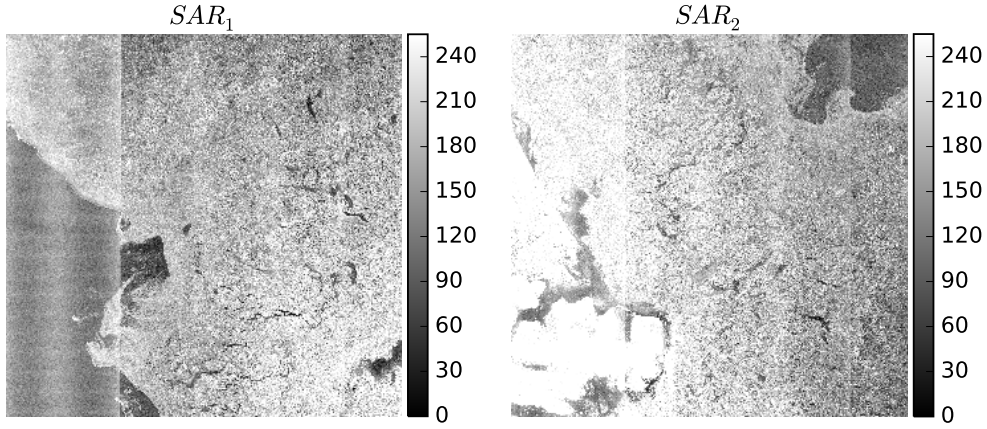
Figure 2. Image pair Fram Strait in HV polarisation after conversion (Equation 2) from backscatter values σ^0 into intensity values with range $0 \leq i \leq 255$ using lower and upper brightness boundaries $\sigma^0_{\min} = -32.5$ dB and $\sigma^0_{\max} = -18.86$ dB.

3.2 Sea ice drift algorithm

The presented sea ice drift algorithm is based on a combination of feature-tracking and pattern-matching, and is designed to utilise the respective advantages of the two approaches. Computationally efficient feature-tracking is used to derive a first estimate of the drift field. The provided vectors serve as initial search position for pattern-matching, that provides accurate drift vectors at each given location including rotation estimate and maximum cross coefficient value. As illustrated in the flowchart in Figure 3, the algorithm consists of five main steps: I Feature tracking, II Filter, III First guess, IV Pattern matching and V Final drift product.

I Feature-tracking

The feature-tracking algorithm used in this work is an adjusted version from Muckenhuber et al. (2016), who introduced a computationally efficient sea ice drift algorithm for Sentinel-1 based on the ORB (Oriented FAST and Rotated BRIEF) algorithm from Rublee et al. (2011). ORB uses the concept of the FAST keypoint detector (Rosten and Drummond, 2006) to find corners on several resolution levels. The patch around each corner is then described using an modified version of the
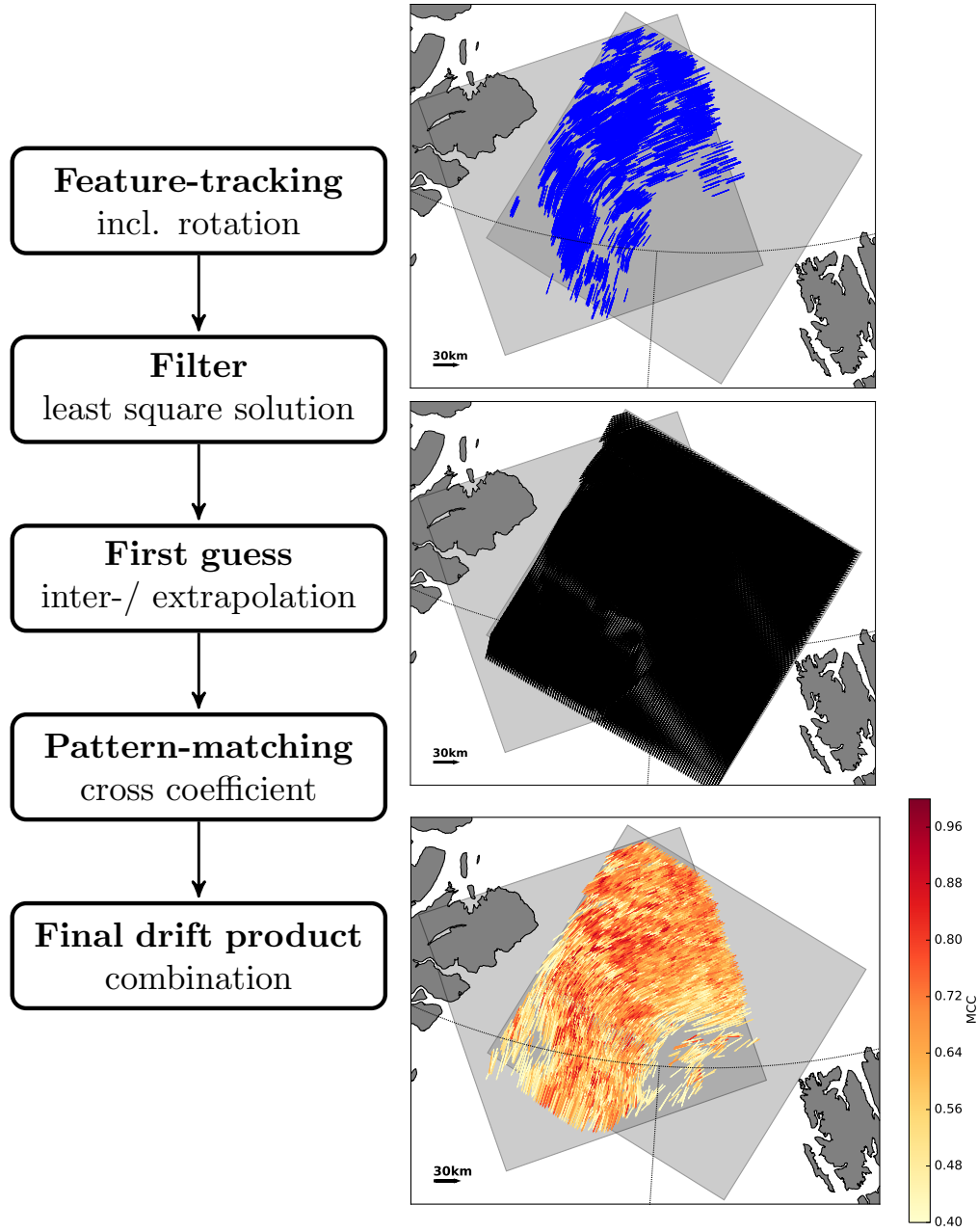
Figure 3. The flowchart on the left depicts the five main steps of the algorithm. The right column illustrates the evolution of the drift results using image pair Fram Strait in HV polarisation and a grid with 4 km spacing. Blue vectors are derived applying an adjusted version of the feature tracking algorithm from Muckenhuber et al. (2016). Black vectors indicate the initial drift estimate (first guess) based on filtered feature-tracking vectors. The final drift product (yellow to red vectors) are derived from combining the first guess with pattern-matching adjustment and applying a minimum cross coefficient value. In this example, a total of 4725 vectors have been found with a MCC value above 0.4 in 4 min.

binary BRIEF descriptor from Calonder et al. (2010). To ensure rotation invariance, the orientation of the patch is calculated using the intensity-weighted centroid. Muckenhuber et al. (2016) applies a Brute Force matcher that compares each feature from the first image to all features in the second image. The comparison of two features is done using the Hamming distance, that represents the number of positions in which the two compared binary feature vectors differ from each other. The best match is accepted if the ratio of the shortest and second shortest Hamming distances is below a certain threshold. Given a suitable threshold, the ratio test will discard a high number of false matches, while eliminating only a few correct matches.

Muckenhuber et al. (2016) found the most suitable parameter setting for our area and time period of interest, including a Hamming distance threshold of 0.75, a maximum drift filter of $0.5\,\mathrm{m/s}$, a patch size of 34×34 pixels and a resolution pyramid with 7 steps combined with a scaling factor of 1.2. Due to the resolution pyramid, the considered feature area varies from $2.7 \times 2.7\,\mathrm{km}$ to $9.8 \times 9.8\,\mathrm{km}$ and the resulting drift field represents a resolution mixture between these boundaries.

We adjust the algorithm from Muckenhuber et al. (2016) by applying a logarithmic scaling for the SAR backscatter values σ_0 instead of the previous used linear scaling (Section 3.1). In addition, we extract for each vector the rotation information α, i.e. how much the feature rotates from the first to the second image.

Applying the adjusted feature-tracking algorithm provides a number of un-evenly distributed vectors (e.g. blue vectors in Figure 3) with start positions x_{1f}, y_{1f} on the first image (SAR_1), end positions x_{2f}, y_{2f} on the subsequent image (SAR_2) and corresponding rotation values α_f. The index f represents a feature-tracking vector and ranges from 1 to F, with F being the total number of derived feature-tracking vectors.

II Filter

To reduce the impact of potentially erroneous feature-tracking vectors on the following steps, outliers are filtered according to drift and rotation estimates derived from least squares solutions using a third degree polynomial function. Considering a matrix $\mathbf{A}$, that contains all end positions x_{2f}, y_{2f} in the following form

$$\mathbf{A} = \begin{pmatrix} 1 & x_{21} & y_{21} & x_{21}^2 & y_{21}^2 & x_{21} * y_{21} & x_{21}^3 & y_{21}^3 \\ 1 & x_{22} & y_{22} & x_{22}^2 & y_{22}^2 & x_{22} * y_{22} & x_{22}^3 & y_{22}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{2F} & y_{2F} & x_{2F}^2 & y_{2F}^2 & x_{2F} * y_{2F} & x_{2F}^3 & y_{2F}^3 \end{pmatrix} \tag{3}$$

, we derive three vectors $\boldsymbol{b}_{x_1}$, $\boldsymbol{b}_{y_1}$ and $\boldsymbol{b}_\alpha$, that represent the least squares solutions for $\mathbf{A}$ and $\boldsymbol{x}_1 = (x_{11}, ..., x_{1F})$, $\boldsymbol{y}_1 = (y_{11}, ..., y_{1F})$ and $\alpha = (\alpha_1, ..., \alpha_F)$ respectively. The starting position x_{1f}, y_{1f} and the rotation α_f of each vector can then be simulated using a third degree polynomial function

$f(x_{2f}, y_{2f}, \boldsymbol{b})$ depending on the end position x_{2f}, y_{2f} and the corresponding least squares solution $\boldsymbol{b} = (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7)$.

$$f(x_{2f}, y_{2f}, \boldsymbol{b}) = b_0 + b_1 x_{2f} + b_2 y_{2f} + b_3 x_{2f}^2 + b_4 y_{2f}^2 + b_5 x_{2f} y_{2f} + b_6 x_{2f}^3 + b_7 y_{2f}^3 \qquad (4)$$

245   If the simulated start position, derived from $f(x_{2f}, y_{2f}, \boldsymbol{b})$, deviates from the feature-tracking start position x_{1f}, y_{1f} by more than 100 pixels, the vector is deleted. The same accounts for rotation outliers. If the simulated rotation deviates from the feature-tracking rotation α_f by more than $60°$, the vector is deleted. We found a third degree polynomial function to be a good compromise between allowing for small to medium scale displacement and rotation discontinuities, and excluding very

250   unlikely vectors, that eventually would disturb the following steps. The parameters for the filter process, i.e. 100 pixels (displacement) and $60°$ (rotation), have been chosen according to visual interpretation using several representative image pairs. Figure 4 illustrates the filter process by depicting the results from image pair Fram Strait.
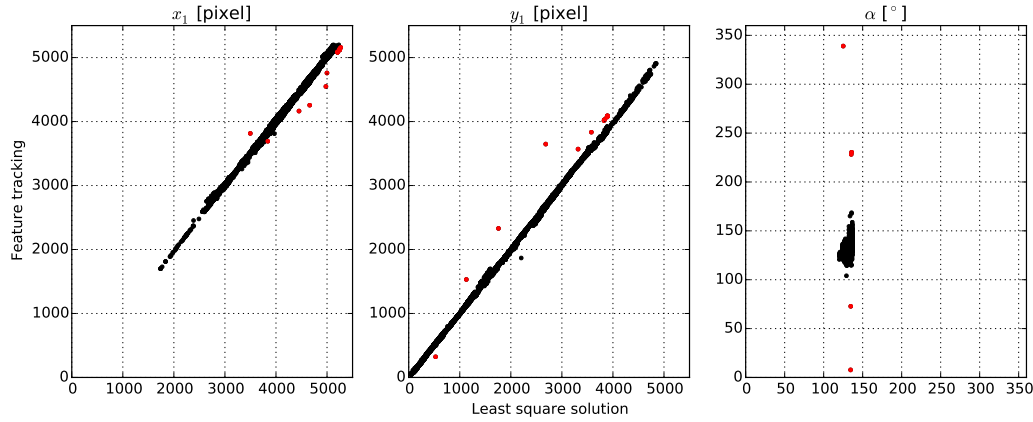
Figure 4. Filter process applied on image pair Fram Strait. The x-axis represent the simulated start position and rotation, derived from $f(x_{2f}, y_{2f}, \boldsymbol{b})$ and the y-axis represent the feature-tracking start position x_{1f}, y_{1f} and rotation α_f. Red points were identified as outliers and deleted.

255   **III First guess**

The remaining feature-tracking vectors are used to estimate the drift incl. rotation on the entire first image, i.e. estimated x_2, y_2 and α values are provided for each pixel on SAR_1 (Figure 5).

Between the feature-tracking vectors, estimated values are constructed by triangulating the input

260   data and performing linear barycentric interpolation on each triangle. That means, the estimated values represent the weighted mean of the three neighbouring feature-tracking values. The interpolated value v_p at any pixel p inside the triangle is given by Equation 5, where v_1, v_2, v_3 represent the

feature-tracking values at the corners of the triangle and A_1, A_2, A_3 are the areas of the triangle constructed by p and the two opposite corners, e.g. A_1 is the area between p, and the corners with value v_2 and v_3.

$$v_p = \frac{A_1 v_1 + A_2 v_2 + A_3 v_3}{A_1 + A_2 + A_3} \tag{5}$$

To provide a first guess for the surrounding area, values are estimated based on the least squares solutions using a linear combination of x_1 and y_1. Considering a matrix $\mathbf{C}$, that contains all start positions x_{1f}, y_{1f} in the following form

$$\mathbf{C} = \begin{pmatrix} 1 & x_{11} & y_{11} \\ 1 & x_{12} & y_{12} \\ \vdots & \vdots & \\ 1 & x_{1F} & y_{1F} \end{pmatrix} \tag{6}$$

, we derive three vectors $\boldsymbol{d}_{x2}$, $\boldsymbol{d}_{y2}$ and $\boldsymbol{d}_\alpha$, that represent the least squares solutions for $\mathbf{C}$ and $\boldsymbol{x}_2 = (x_{21}, ..., x_{2F})$, $\boldsymbol{y}_2 = (y_{21}, ..., y_{2F})$ and $\alpha = (\alpha_1, ..., \alpha_F)$ respectively. The estimated end position x_2, y_2 and rotation α at any location can then be simulated using the linear function $f(x_1, y_1, \boldsymbol{c})$ depending on the start position x_1, y_1 and the corresponding least squares solution $\boldsymbol{d} = (d_0, d_1, d_2)$.

$$f(x_1, y_1, \boldsymbol{d}) = d_0 + d_1 x_1 + d_2 y_1 \tag{7}$$

An example for the resulting first guess, i.e. estimated values for x_2, y_2 and α on SAR_1, is shown in Figure 5 and corresponding vectors are shown in black in Figure 3. Note that rotation α includes both the relative image rotation from SAR_1 to SAR_2 and the actual rotation of the feature itself. The introduced algorithm provides also the relative image rotation by projecting the left corners of SAR_2 onto SAR_1 and calculating the angle between the left edges of SAR_1 and SAR_2. The actual rotation of the features can easily be obtained by subtracting the relative image rotation from α.

IV Pattern-matching

The estimated drift field derived from feature-tracking provides values for x_2, y_2 and α at any location on SAR_1. The uncertainty of this estimate however, increases with distance d to the closest feature-tracking vector. Therefore, small to medium scale adjustments of the estimates are necessary, depending on the distance d. We apply pattern-matching at chosen points of interest to provide more accurate drift vectors and adjust the rotation estimate at these specific locations.

The used pattern-matching approach is based on the maximisation of the normalised cross coefficient. Considering a small template t_1 around the point of interest from SAR_1 with size $t_{1s} \times t_{1s}$
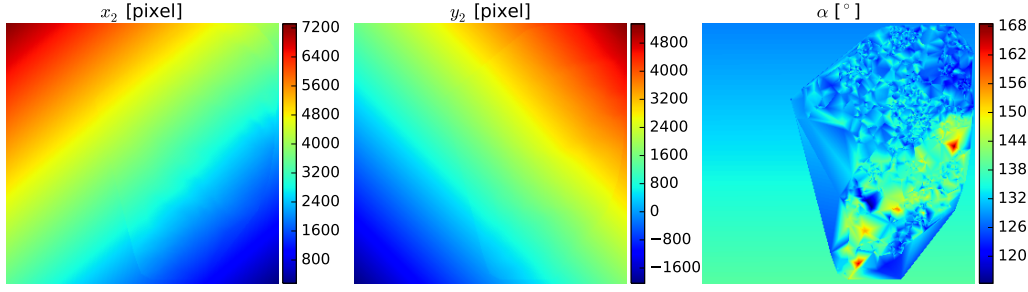
Figure 5. Example of estimated drift and rotation (first guess) based on filtered feature-tracking vectors using image pair Fram Strait. The three panels show the components x_2, y_2 of the estimated end positions and the estimated rotation α for each pixel on the first image (SAR$_1$).

and a larger template t_2 around the location x_2, y_2 (defined by the corresponding first guess) from SAR_2 with size $t_{2s} \times t_{2s}$, the normalised cross coefficient matrix **NCC** is defined as (Hollands , 2012):

$$\mathbf{NCC}(x,y) = \frac{\sum_{x',y'}(t_1'(x',y')t_2'(x+x',y+y'))}{\sqrt{\sum_{x',y'}t_1'(x',y')^2 \sum_{x',y'}t_2'(x+x',y+y'))^2}} \tag{8}$$

$$t_1'(x',y') = t_1(x',y') - \frac{1}{t_{1s}^2}\sum_{x'',y''}t_1(x'',y'') \tag{9}$$

$$t_2'(x+x',y+y') = t_2(x+x',y+y') - \frac{1}{t_{1s}^2}\sum_{x'',y''}t_2(x+x'',y+y'') \tag{10}$$

with $t_1(x',y')$ and $t_2(x',y')$ representing the value of t_1 and t_2 at location x',y'. The summations are done over the size of the smaller template, i.e. x', y', x'' and y'' go from 1 to t_{1s}. Template t_1 is moved with step size 1 pixel over template t_2 both in horizontal (x) and vertical (y) direction and the cross coefficient values for each step are stored in the matrix **NCC** with size $(1 + t_{s2} - t_{s1}) \times (1 + t_{s2} - t_{s1})$. The highest value in the matrix **NCC**, i.e. the the maximum normalised cross coefficient value MCC, represents the location of the best match and the corresponding location adjustment is given by dx and dy.

$$(\frac{1+t_{s2}-t_{s1}}{2}+dx, \frac{1+t_{s2}-t_{s1}}{2}+dy) = argmax(\mathbf{NCC}(x,y) \tag{11}$$

To restrict the search area t_{2s} to a circle, we set all values of **NCC** that are further than $t_{2s}/2$ away from the centre position to zero.

To account for rotation adjustment, the matrix **NCC** is calculated several times: template t_1 is rotated around the initially estimated rotation α from $\alpha - \beta$ to $\alpha + \beta$ with step size $\Delta\beta$. The angle β
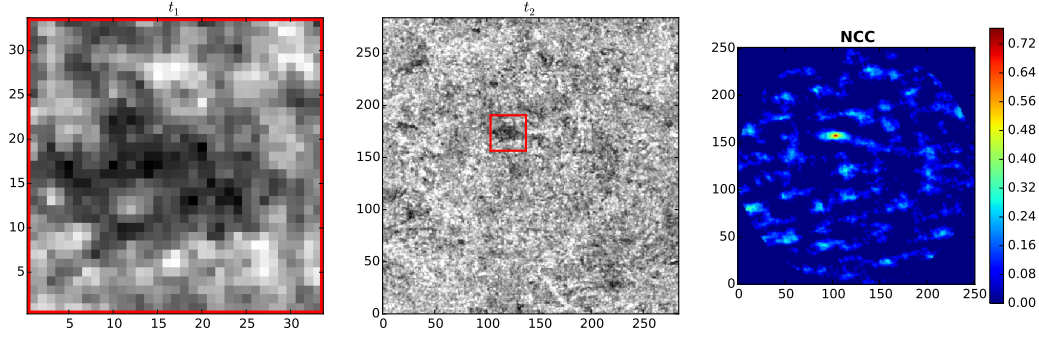
Figure 6. Pattern-matching using initial drift estimate from feature-tracking: The small template t_1 (left) around the point of interest on SAR_1 is rotated from $\alpha - \beta$ to $\alpha + \beta$ and matched with the large template t_2 (middle) from SAR_2, that has its centre at the estimated end position x_2, y_2. The right contour plot shows the normalised cross coefficient matrix **NCC** of the rotation β^* that provided the highest maximum cross coefficient MCC. The estimated end position x_2, y_2 of this example has to be adjusted by $dx = -21$ pixels, $dy = 32$ pixels to fit with the location of $MCC = 0.71$. Rotation adjustment β^* was found got be $3°$. NB: X and Y-axis represent pixel coordinates.

is the maximum additional rotation and represents therefore the rotation restriction. The **NCC** matrix with the highest cross coefficient value MCC is returned.

To illustrate the pattern-matching process, an example, taken from image pair Fram Strait, is shown in Figure 6.

315

The described process demands the specification of four parameters: t_{1s}, t_{2s}, β and $\Delta\beta$.

The size of the small template $t_{1s} \times t_{1s}$ defines the considered area that is tracked from one image to the next and hence, affects the resolution of the resulting drift product. In order to be consistent with the resolution of the feature-tracking step and achieve our goal of a sea ice drift product with a spatial scaling of less than 5 km, we use the size of the feature-tracking patch of the pyramid level with the highest resolution to define the size of t_1. That means, we use $t_{s1} = 34$ pixels (2.7 km).

The size of the larger template $t_{2s} \times t_{2s}$ restricts the search area on SAR_2, i.e. how much the first guess can be adjusted geographically, and the angle β restricts the rotation adjustment of the first guess α. The three parameter t_{2s}, β and $\Delta\beta$ have a strong influence on the computational efficiency of the drift algorithm. Meaning that an increase of t_{2s}, β and a decrease of $\Delta\beta$ increase the computational effort of the pattern-matching step. Based on visual interpretation of several representative image pairs, we found $\Delta\beta = 3°$ to be a good compromise between matching performance and computational efficiency.

Since the uncertainty of the first guess increases with distance d (Figure 7) to the closest feature-tracking vector, the search restrictions t_{2s} and β should increase with d. To find useful restrictions for t_{2s} and β, we calculated drift vectors using very high values for t_{2s} and β, i.e. being computationally
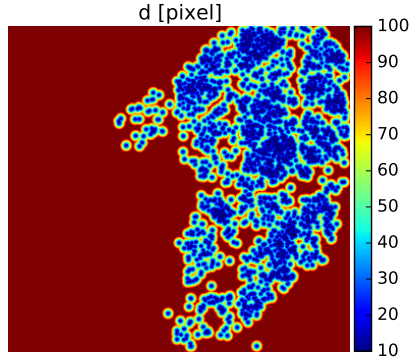
Figure 7. Example to illustrate the distribution of distance d to the closest feature-tracking vector using image pair Fram Strait. Values outside the range $d_{min} \leq d \leq d_{max}$ are set to $d_{min} = 10$ and $d_{max} = 100$. The points with value d_{min} represent the start positions x_1, y_1 of the feature-tracking vectors on SAR_1.

more demanding than we anticipate, and compared the results with the GPS drift buoy dataset from the N-ICE2015 expedition. Based on the results (Section 4) we found the following functions to represent useful restrictions for our area and time period of interest.

$$t_{2s}(d) = t_{1s} + 2d \qquad d_{min} \leq d \leq d_{max} \qquad d \in \mathbb{N} \tag{12}$$

$$\beta(d) = \begin{cases} 9 & \text{if } d < d_{max} \\ 12 & \text{if } d \geq d_{max} \end{cases} \tag{13}$$

The values for d_{min}, d_{max}, β and $\Delta\beta$ can easily be varied in the algorithm to adjust for e.g. different areas, drift conditions or a different compromise between matching performance and computational efficiency.

V Final drift product

In the last step, the small to medium scale displacement adjustments from pattern-matching are added to the estimated first guess derived from feature-tracking. Using buoy comparison, we found that the probability for large displacement errors decreases with increasing MCC value (Section 4). Therefore, vectors that have a MCC value below the threshold MCC_{min} are removed. We found $MCC_{min} = 0.4$ to be a good filter value, but this value can easily be adjusted in the algorithm depending on the sought compromise between amount of vectors and error probability. The algorithm returns the final drift vectors in longitude, latitude, the corresponding first guess rotation α and the rotation adjustment β in degrees and the maximum cross coefficient value MCC. An example for

the final product is depicted with yellow to red coloured vectors in Figure 3. The colour scale refers to the MCC value, indicating the probability for an erroneous vector.

3.3 Comparison with buoy data

Sentinel-1 image pairs have been selected automatically according to position and timing of the GPS buoy data from the N-ICE2015 expedition. Each pair yielded more than 300 drift vectors using the feature-tracking algorithm from Muckenhuber et al. (2016) and had a time difference between the two acquisitions of less than three days. Drift vectors have been calculated with the presented algorithm starting at the buoy GPS position with the least time difference to the acquisition of the first satellite image. The distance D between the calculated end position on the second image and the buoy GPS position with the least time difference to the second satellite acquisition has been calculated using the following equation:

$$D = \sqrt{(u-U)^2 + (v-V)^2} \tag{14}$$

where u and v represent eastward and northward drift components of the displacement vector derived by the algorithm, and U and V the corresponding drift components of the buoy.

4 Results

4.1 Search restrictions evaluation

To find suitable values for restricting the size of the search window t_{2s} and the rotation range defined by β, we calculated drift vectors, that can be compared to the considered GPS buoy dataset, using restrictions that are computationally more demanding than we anticipate for the recommended setting, i.e. $t_{2s} = 434$ pixels and $\beta = 18°$. These values corresponds to a possible pattern-matching adjustment of up to 200 pixels (16 km) and 18° in any direction independent of the distance d to the closest feature-tracking vector.

Based on an automatic search, we found 240 matching Sentinel-1 image pairs (consisting of 110 images), that allowed for comparison with 689 buoy vectors (Figure 8). The distance D (Equation 14) between the buoy location at the time of the second image SAR_2 and the corresponding algorithm result, represents the error estimate for one vector pair. To identify algorithm results that are more likely erroneous, vector pairs with a value D above 1000 m are marked with red dots in Figure 9 and Figure 10. Vector pairs with $D < 1000$ m are plotted with black dots.

Figure 9 and Figure 10 show the resulting pattern-matching adjustment of location (dx, dy) and rotation ($d\beta$) using the computationally demanding restrictions. The values are plotted against distance d to the next feature tracking vector in order to identify the dependence of the parameters on d. The blue lines in Figure 9 and Figure 10 indicate the recommended restrictions. This represents a
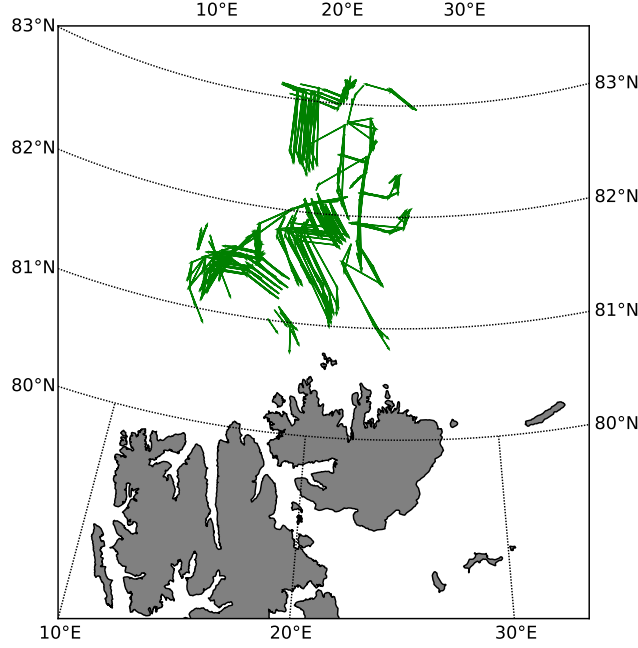
Figure 8. Considered buoy trajectories from the N-ICE2015 expedition that were used for comparison with algorithm results.

compromise between computational efficiency and allowing the algorithm to adjust the first guess as much as needed for our time period and area of interest. The corresponding functions for $t_{2s}(d)$ and

385   $\beta(d)$ are given in Equation 12 and Equation 13 and the recommended boundary values for distance d are $d_{min} = 10$ and $d_{max} = 100$.

4.2 Validation

Using the recommended search restrictions from above, the algorithm has been validated against the N-ICE2015 GPS buoy data set (Figure 8). The automatic search provided 241 image pairs (consisting

390   of 110 images) and 714 vectors for comparison for the considered time period (15th January to 22nd April) and area (80.5° N to 83.5° N and 12° E to 27° E). NB: this is a higher number of vectors than found for the evaluation of the search restrictions, since the used search windows t_2 are smaller and vectors closer to the SAR edge may be included.

    The results of the conducted validation are shown in Figure 11. We found that the probability for

395   a large D value (representative for the error) decreases with increasing maximum cross coefficient value MCC. Therefore we suggest to exclude matches with a MCC value below a certain threshold MCC_{min}. This option is embedded into the algorithm, but can easily be adjusted or turned off by setting $MCC_{min} = 0$. Based on the findings shown in Figure 11, we recommend a cross coefficient
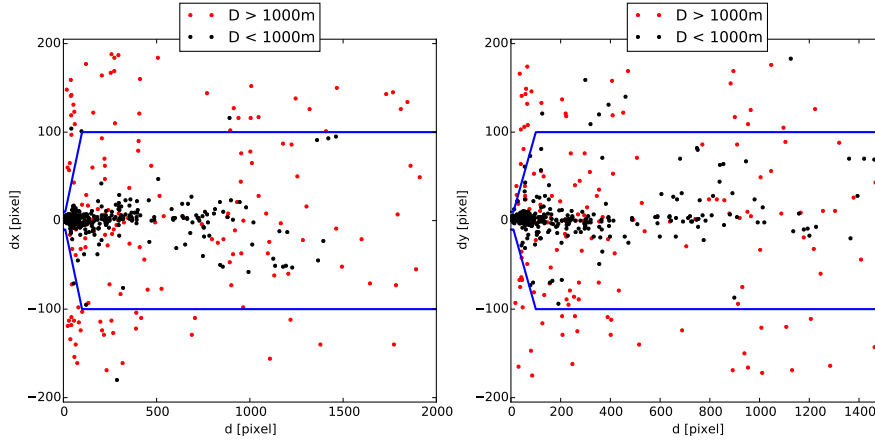
Figure 9. Pattern-matching location adjustment dx and dy in x and y direction versus distance d to closest feature tracking vector. D represents the difference between buoy GPS position and algorithm result. The blue lines indicate the recommended setting for t_{2s} (Equation 12) with $d_{min} = 10$ and $d_{max} = 100$.
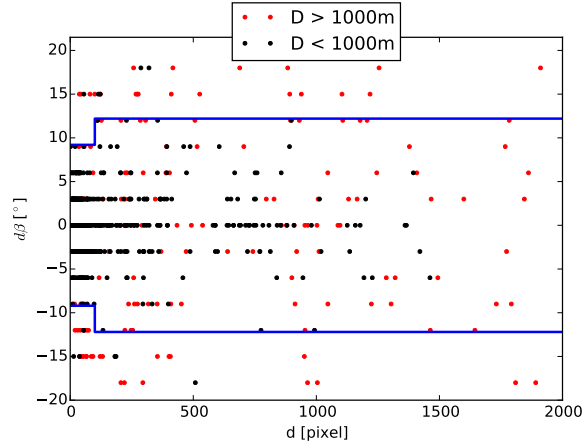
Figure 10. Pattern-matching rotation adjustment $d\beta$ versus distance d to closest feature tracking vector. D represents the difference between buoy GPS position and algorithm result. The blue lines indicate the recommended setting for β (Equation 13) with $d_{min} = 10$ and $d_{max} = 100$.

threshold $MCC_{min} = 0.4$ for our time period and area of interest. Using the suggested threshold
400   reduces the number of vector pairs from 714 to 565.

The conducted validation also reveals a logarithmic normal distribution of the distance D (Equation 14) that can be expressed by the following probability density function (solid red line in Figure 11):

$$lnN(D; \mu, \sigma) = \frac{1}{\sigma D \sqrt{2\pi}} e^{-\frac{(\ln D - \mu)^2}{2\sigma^2}} \tag{15}$$
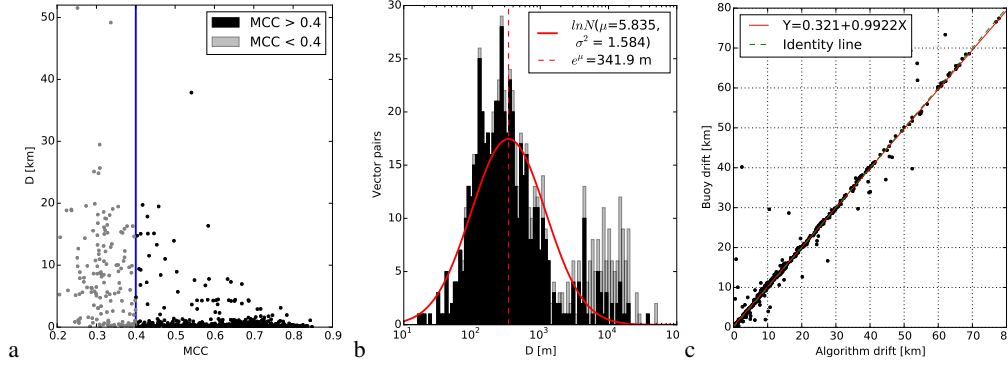
Figure 11. Calculated ice drift using recommended search restrictions compared to buoy GPS data. Light grey represents vectors with maximum cross coefficient values $MCC < 0.4$ and results after using the suggested threshold $MCC_{min} = 0.4$ are shown in black. (a) MCC values against distance D (Equation 14) between algorithm and buoy end position. The blue line indicates the recommended setting for $MCC_{min} = 0.4$. (b) Logarithmic histogram of distance D with 100 bins between 10 m and 10^5 m including a logarithmic normal distribution (solid red line) that was fitted to the results with $MCC > 0.4$. (c) Comparison of drift distance derived from algorithm against buoy displacement.

Table 1. Recommended parameter setting for sea ice drift retrieval from Sentinel-1 using the presented algorithm.

Parameter	Meaning	Recommended setting
$[\sigma^0_{\min}, \sigma^0_{\max}]$ (HH)	Brightness boundaries for HH channel	[-25 dB, -10.97 dB]
$[\sigma^0_{\min}, \sigma^0_{\max}]$ (HV)	Brightness boundaries for HV channel	[-32.5 dB, -18.86 dB]
t_{1s}	Size of template t_1	34 pixels (2.7 km)
$[d_{min}, d_{max}]$	Boundaries for distance d	[10 pixels, 100 pixels]
MCC_{min}	Threshold for cross coefficient	0.4
$\Delta\beta$	Rotation angle increment	3°

405   with μ and σ being the mean and standard deviation of the variable's natural logarithm. We found the mean and variance of the distribution lnN to be $\mu = 5.835$ and $\sigma^2 = 1.584$. The median of the logarithmic normal distribution is $e^\mu = 341.9\,m$ (dashed red line in Figure 11).

4.3   Recommended parameter setting

Based on the restriction evaluation, our experience with the algorithm behaviour, and considering a
410   good compromise between computational efficiency and high quality of the resulting vector field, we recommend the parameter setting shown in Table 1. The corresponding recommended values for $t_{2s}(d)$ and $\beta(d)$ are given in Equation 12 and Equation 13.

Table 2. Processing time for sea ice drift retrieval from image pair Fram Strait on a grid with 4 km (50 pixels) spacing using HV polarisation (Figure 3). Representative for an image with large overlap and good coverage with feature-tracking vectors.

Process	Time [s]
Create Nansat objects from Sentinel-1 image pair and read matrixes	70
I Feature-tracking	66
II Pattern-matching and III Combination	107
$\sum$ Sea ice drift retrieval	243

4.4 Computational efficiency

The processing time depends on the parameter setting and the chosen vector distribution. Using
the recommended parameter setting from Table 1, allows high-resolution sea ice drift retrieval from
a Sentinel-1 image pair within a few minutes. Figure 3 depicts calculated ice drift vectors for the
image pair Fram Strait on a grid with 4 km (50 pixels) spacing. The corresponding processing times
are shown in Table 2. The calculations have been done using a MacBook Pro from early 2013 with
a 2.7 GHz Intel Core i7 processor and 8 GB 1600 MHz DDR3 memory. The total processing time
for 4725 vectors with a normalised cross coefficient value above 0.4, is about 4 minutes. This can be
considered a representative value for an image pair with large overlap, good coverage with feature-
tracking vectors and 4 km grid spacing.

The initial process in Table 2 'Create Nansat objects from Sentinel-1 image pair and read matrixes'
takes the same amount of computational effort for all image pairs consisting of Sentinel-1 images
with 400x400 km coverage.

The process 'I Feature-tracking' depends on the setting of the feature-tracking algorithm and
varies strongly with the chosen number of features. Using the recommended setting from Mucken-
huber et al. (2016), that includes the number of features to be 100000, the presented computational
effort can be considered representative for all image pairs, independent of chosen points of interest
and overlap of the SAR scenes.

The last process 'II Pattern-matching and III Combination' however, depends on the considered
image pair and the chosen drift resolution. The computational effort is proportional to the number
of chosen points of interest. Given a evenly distributed grid of points of interest, the computational
effort increases with overlapping area of the SAR scenes, since pattern-matching adjustments are
only calculated in the overlapping area. The effort potentially decreases with a higher number of
well distributed feature-tracking vectors, since the size of the search windows t_2 (and slightly the
range of the angle β) increases with distance d to the closest feature-tracking vector.

5   Discussion and outlook

To estimate the accuracy of the introduced algorithm, we compared drift results from 240 Sentinel-1
image pairs with corresponding GPS positions from the N-ICE2015 buoy data set. We found a log-
arithmic error distribution with a median at 341.9 m (Figure 11). The derived error values represent
a combination of the following error sources:

- Timing: Buoy GPS data were collected every 1-3 hours and the timing does not necessarily
  match with the satellite acquisition time.

- Resolution: The algorithm returns the drift of a pattern (recommended size = 34 pixels, see
  Table 1), whereas the buoy measures the drift at a single location.

- Conditions: The ice conditions around the buoy is not known well enough to exclude the
  possibility that the buoy is floating in a lead. In this case, the buoy trajectory could represent a
  drift along the lead rather then the drift of the surrounding sea ice.

- actual error of the algorithm.

A main advantage of the combined algorithm compared to simple feature-tracking, is the user de-
fined positioning of the drift vectors. The current algorithm setup allows the user to choose whether
the drift vectors should be positioned at certain points of interest or on a regular grid with adjustable
spacing. Constricting the pattern-matching process to the area of interest minimises the computa-
tional effort according to the individual needs.

The recommended parameters shown in Table 1 are not meant as a fixed setting, but should rather
give a suggestion and guideline to estimate the expected results and the corresponding computational
effort. The parameters can easily be varied in the algorithm setup and should be chosen according to
availability of time, computational power, number of image pairs, needed accuracy, area of interest
and expected ice conditions (e.g. strong rotation).

The presented combination of feature-tracking and pattern-matching can be applied to any other
application that aims to derive displacement vectors computationally efficient from two consecutive
images. The only restriction is that images need to depict edges, that can be recognised as keypoints
for the feature-tracking algorithm, and the conversion into intensity values i (Equation 2) needs to
be adjusted according to the image type.

The remote sensing group at NERSC is currently developing a new pre-processing step to remove
thermal noise on HV images over ocean and sea ice. First tests have shown a significant improvement
of the sea ice drift results using this pre-processing step before applying the presented algorithm. This
is ongoing work and will be included into a future version of the algorithm.

The European Space Agency is also in the process of improving their thermal noise removal for
Sentinel-1 imagery. Noise removal in range direction is driven by a function that takes measured

noise power into account. Until now, noise measurements are done at the start of each data acquisition, i.e. every 10-20 minutes, and a linear interpolation is performed to provide noise values every 3 seconds. The distribution of noise measurements showed a bimodal shape and it was recently dis-

475 covered that lower values are related to noise over ocean while higher values are related to noise over land. This means, that Sentinel-1 is able to sense the difference of the earth surface brightness temperature similar to a passive radiometer. When the data acquisition includes a transition from ocean to land or vice versa, the linear interpolation fails to track the noise variation. The successors of Sentinel-1A/B are planned to include more frequent noise measurements. Until then, ESA wants

480 to use the 8-10 echoes after the burst that are recorded while the transmitted pulse is still travelling and the instrument is measuring the noise. This will provide noise measurements every 0.9 seconds and allows to track the noise variations in more detail. In addition, ESA is planning to introduce a change in the data format during 2017 that shall remove the noise shaping in azimuth. These efforts are expected to improve the performance of the presented algorithm significantly.

485 Having a computationally efficient algorithm with adjustable vector positioning allows not only to provide near-real time operational drift data, but also the investigation of sea ice drift over large areas and long time periods. Our next step is to embed the algorithm into a super-computing facility to further test the performance in different regions, time periods and ice conditions. The goal is to deliver large ice drift datasets and open-source operational sea ice drift products with a spatial

490 resolution of less than 5 km.

This work is linked to the question how to combine the different timings of the individual image pairs in a most useful way. Having more frequent satellite acquisitions, as we get with the Sentinel-1 satellite constellation, enables to derive displacements for shorter time gaps and the calculated vectors will reveal more details e.g. rotational motion due to tides. As part of a scientific cruise with

495 KV-Svalbard in July 2016, we deployed three GPS trackers on loose ice floes and pack-ice in Fram Strait. The trackers send their position every 5-30 min to deliver drift information with high temporal resolution. This efforts shall help to gain a better understanding of short-term drift variability and by comparison with calculated sea ice drift, we will investigate how displacement vectors from subsequent satellite images relate to sea ice displacements with higher temporal resolution.

500 The focus of this paper in terms of polarisation was put on the HV channel, since this polarisation provides on average four times more feature tracking vectors than HH and therefore delivers a finer initial drift for the first guess. We found our area of interest well covered with HV images, but other areas in the Arctic and Antarctic reveal a better coverage in HH polarisation. Considering the four representative feature-tracking image pairs from Muckenhuber et al. (2016), the the relatively best

505 HH polarisation performance (i.e. most vectors from HH, while at the same time fewest vectors from HV) was the image pair that showed the least time difference, i.e. 8 h, compared to 31 h, 33 h and 48 h. Therefore, we assume that the HV polarisation provides more features that are better preserved over time. And more consistent features would also favour the performance of the pattern-matching

step. However, at this point, this is just an assumption and will be addressed in more detail in our future work.

Utilising the advantage of dual polarisation (HH+HV) is certainly possible with the presented algorithm, but increases the computational effort. A simple approach is to combine the feature tracking vectors derived from HH and HV and produce a combined first-guess. Pattern-matching can be performed based on this combined first-guess for both HH and HV individually and the results can be compared and eventually merged into a single drift product. Having two drift estimates for the same position, from HH and HV pattern-matching respectively, would also allow to disregard vectors that disagree significantly. However, this option would increase the computational effort by two, meaning that the presented Fram Strait example would need about 8 min processing time.

After implementing the presented algorithm into a super-computing facility, we aim to test and compare the respective performance of HV, HH and HH+HV on large datasets to identify the respective advantages.

The current setting of the feature-tracking algorithm applies a maximum drift filter of 0.5 m/s. We found this to be a reasonable value for our time period and area of interest. However, when considering extreme drift situations in Fram Strait and a short time interval between image acquisitions, this threshold should be adjusted.

As mentioned above, we deployed three GPS tracker in Fram Strait and they recorded their positions with a temporal resolution of 5-30 min between 8th July until 9th September 2016 in an area covering 75° N to 80° N and 4° W to 14° W. Considering the displacements with 30 min interval, we found velocities above 0.5 m/s on a few occasions, when the tidal motion adds to an exceptionally fast ice drift.

The GPS data from the hovercraft expedition FRAM2014-2015 (https://sabvabaa.nersc.no), that was collected with a temporal resolution of 10 s between 31st August 2014 until 6th July 2015, did not reveal a single 30 min interval during which the hovercraft was moved by ice drift more than 0.45 m/s. The hovercraft expedition started at 280 km south from the North Pole towards the Siberian coast, crossed the Arctic Ocean towards Greenland and was picked up in the north-western part of Fram Strait.

In case the estimated drift from feature-tracking reaches velocities close to 0.5 m/s, the pattern-matching step might add an additional degree of freedom of up to 8 km, which could eventually lead to a higher drift result than 0.5 m/s, depending on the time interval between the acquisitions. The smaller the time difference, the larger is the potentially added velocity. In order to be consistent when combining the drift information from several image pairs with different timings, one should apply a maximum drift filter on the final drift product of the presented algorithm that has the same maximum velocity as the feature-tracking filter. The corresponding function is implemented in the distributed open-source algorithm.

Appendix A: Open-source distribution

The presented sea ice drift retrieval method is based on open-source satellite data and software to ensure free application and easy distribution. Sentinel-1 SAR images are distributed by ESA for free within a few hours of acquisition under https://scihub.esa.int/dhus/. The algorithm is programmed in Python (source code: https://www.python.org) and makes use of the open-source libraries Nansat, openCV and SciPy. Nansat is a scientist friendly Python toolbox for processing 2-D satellite Earth observation data (source code: https://github.com/nansencenter/nansat). OpenCV (Open Source Computer Vision) is a computer vision and machine learning software library and can be downloaded under http://opencv.org. SciPy (source code: https://www.scipy.org) is a Python-based ecosystem of software for mathematics, science, and engineering. The presented sea ice drift algorithm is distributed as open-source software under https://github.com/nansencenter/sea_ice_drift.

Acknowledgements. This research was supported by the Norwegian Research Council project IceMotion (High resolution sea-ice motion from Synthetic Aperture Radar using pattern tracking and Doppler shift, project number 239998/F50). We thank Polona Itkin and Gunnar Spreen for providing us the buoy GPS data that were collected as part of the N-ICE2015 project with support by the Norwegian Polar Institute's Centre for Ice, Climate and Ecosystems (ICE) and its partner institutes. The used satellite data were provided by the European Space Agency. We thank Nuno Miranda for information on ESA's de-noising efforts for Sentinel-1. We also thank the developer group of the Earth observation toolbox Nansat and Anton Korosov for his inputs.

References

Berg, A. and Eriksson L.E.B.: Investigation of a Hybrid Algorithm for Sea Ice Drift Measurements Using
565   Synthetic Aperture Radar Images IEEE Transactions on Geoscience and Remote Sensing, Vol. 52, No. 8,
        5023–5033, 2014.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P.: BRIEF: Binary Robust Independent Elementary Features,
        CVLab, EPFL, Lausanne, Switzerland, 2010.

ESA: Sentinel-1 ESA's Radar Observatory Mission for GMES Operational Services, ESA Communications,
570   SP-1322/1, ISBN: 978-92-9221-418-0, ISSN: 0379-6566, 2012.

Hollands, T.: Motion tracking of sea ice with SAR satellite data, dissertaiton, Section 2: Estimation of motion
        from images, University Bremen, 2012.

Hollands, T. and Dierking, W.: Performance of a multiscale correlation algorithm for the estimation of sea-ice
        drift from SAR images: initial results, Ann. Glaciol., 52, 311–317, 2011.

575 IPCC – Intergovernmental Panel on Climate Change: Climate Change 2013: The Physical Science Basis, Fifth
        Assessment Report, AR5, 317–382, 323–335, 2013.

Komarov, A.S., and Barber, D.G.: Sea Ice Motion Tracking From Sequential Dual-Polarization RADARSAT-
        2 Images, IEEE Transactions on Geoscience and Remote Sensing, Vol. 52(1), No. 1, 121–136, doi:
        10.1109/TGRS.2012.2236845, 2014.

580 Korosov A.A., Hansen W.M., Dagestad F.K., Yamakawa A., Vines A., Riechert A.: Nansat: a Scientist-
        Orientated Python Package for Geospatial Data Processing, Journal of Open Research Software, 4: e39,
        DOI: http:// dx.doi.org/10.5334/jors.120, 2016

Kwok, R., Curlander J.C., McConnell R., and Pang S.: An Ice Motion Tracking System at the Alaska SAR
        Facility, IEEE Journal of Oceanic Engineering, Vol. 15, No. 1, 44–54, 1990.

585 Muckenhuber S., Korosov A.A., and Sandven S. (2016): Open-source feature-tracking algorithm for sea ice drift
        retrieval from Sentinel-1 SAR imagery, The Cryosphere, 10, 913-925, doi:10.5194/tc-10-913-2016, 2016

Nansen, F.: The Oceanography of the North Polar Basin. Scientific Results, Vol. 3, 9, Longman Green and Co.,
        Kristinania, Norway, 1902.

Pedersen, L.T., Saldo, R. and Fenger-Nielsen, R.: Sentinel-1 results: Sea ice operational moni-
590   toring, Geoscience and Remote Sensing Symposium (IGARSS), IEEE International, 2828–2831,
        doi=10.1109/IGARSS.2015.7326403, 2015

Rampal, P., Weiss, J., Marsan, D. and Bourgoin M.: Arctic sea ice velocity field: General circula-
        tion and turbulent-like fluctuations, Journal of Geophysical Research: Oceans, Vol. 114, Nr. C10014,
        doi=10.1029/2008JC005227, 2009.

595 Rampal, P., Weiss, J. and Marsan, D.: Positive trend in the mean speed and deformation rate of Arctic sea ice
        1979–2007, Journal of Geophysical Research: Oceans, Vol. 114, Nr. C5013, doi=10.1029/2008JC005066,
        2009b.

Rosten, E. and Drummond, T.: Machine learning for high-speed corner detection, in European Conference on
        Computer Vision, ISBN 978-3-540-33833-8, 430–443, doi: 10.100711744023_34, 2006.

600 Rublee, E., Rabaud, V., Konolige, K., and Bradski, G.: ORB: an efficient alternative to SIFT or SURF, IEEE I.
        Conf. Comp. Vis. (ICCV), ISBN: 978-1-4577-1101-5, 2564–2571, doi: 10.1109ICCV.2011.6126544, 6–13
        Nov, 2011.

Spreen, G. and Itkin, P.: N-ICE2015 buoy data, Norwegian Polar Institute, https://data.npolar.no/dataset/6ed9a8ca-95b0-43be-bedf-8176bf56da80, 2015.

605 Thomas, M., Geiger, C. A., and Kambhamettu, C.: High resolution (400 m) motion characterization of sea ice using ERS-1 SAR imagery, Cold Reg. Sci. Technol., 52, 207–223, 2008.

Thorndike, A. S. and Colony, R.: Sea ice motion in response to geostrophic winds, Journal of Geophysical Research: Oceans, Vol. 87, Nr. C8, 5845–5852, doi=10.1029/JC087iC08p05845, 1982.

Widell, K., Østerhus, S. and Gammelsrød, T.: Sea ice velocity in the Fram Strait monitored by moored instru-
610 ments, Geophysical Research Letters, Vol. 30, Nr. 19, doi=10.1029/2003GL018119, 2003.