The Cryosphere Discuss., doi:10.5194/tc-2016-261, 2016
Manuscript under review for journal The Cryosphere
Published: 5 December 2016

The Cryosphere
Discussions
Open Access

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. A computational efficient feature-tracking algorithm produces an initial drift estimate and limits the search area for the pattern-matching, that provides small to medium scale drift adjustments and normalised cross correlation values as quality measure. The algorithm is designed to utilise the respective advantages of the two approaches and allows drift calculation at user defined locations. The pre-processing of the Sentinel-1 data has been optimised to retrieve a feature distribution that depends less on SAR backscatter peak values. A recommended parameter set for the algorithm has been found using a representative image pair over Fram Strait and 350 manually derived drift vectors as validation. Applying the algorithm with this parameter setting, sea ice drift retrieval with a vector spacing of 8 km on Sentinel-1 images covering 400 km x 400 km, takes less than 3.5 minutes on a standard 2.7 GHz processor with 8 GB memory. For validation, buoy GPS data, collected in 2015 between 15^{th} January and 22^{nd} April and covering an area from 81° N to 83.5° N and 12° E to 27° E, have been compared to calculated drift results from 261 corresponding Sentinel-1 image pairs. We found a logarithmic distribution of the error with a peak at 300 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 seasonality 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.

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.

Space-borne Synthetic Aperture Radar (SAR) are delivering systematic acquisitions of sea ice covered oceans since the early 90s and Kwok et al. (1990) showed that sea ice displacement 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 geophysical processor system from Kwok et al. (1990) has been used to calculate sea ice drift fields for the entire Arctic every week with a spatial resolution of 5 km for the time period 1997–2012. This extensive dataset makes use of SAR data from Radarsat and ENVISAT (Environmental Satellite). A high-resolution sea ice drift algorithm for SAR images from ERS-1 (European Remote-sensing Satellite) 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. Komarov and Barber (2014) and Muckenhuber et al. (2016) have evaluated the sea ice drift retrieval performance of dual-polarisation SAR imagery using pattern-matching and feature-tracking 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.

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. Making use of Sentinel-1 data, an operational sea ice drift product with 10 km resolution is provided by the Danish Technical University (DTU) as part of the Copernicus Marine Environment Monitoring Service (CMEMS, http://marine.copernicus.eu). 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.

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 independent of their neighbours, which is an important advantage for resolving shear zones, rotation and divergence/convergence zones. However, the resulting vector field is not evenly distributed in space and large gaps may occur between densely covered areas (Muckenhuber et al., 2016). 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 and delivers a comparable quality estimate for each vector. 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 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 implementation 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). 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 Mode Ground Range Detected with Medium Resolution” and the presented algorithm is built for processing this data type. The considered images have a resolution of 93 m range × 87 m azimuth with residual planimetric distortions within 10 m (Schubert et al., 2014). The covered area per image is 400 km × 400 km and the data are provided with a pixel spacing of 40 m × 40 m in both HH and HV polarisation.

For parameter tuning, we used the image pair 'Fram Strait', including 350 manually derived drift vectors as validation, from Muckenhuber et al. (2016). The acquisition times of the two consecutive images are 2015-03-28 07:44:33 (UTC) and 2015-03-29 16:34:52 (UTC), and the covered area including validation vectors are shown in Figure 6. 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 validate the algorithm results, we used GPS data from drift 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 considered drift data have been collected in 2015 between 15^{th} January and 22^{nd} April, and cover an area ranging from 81° N to 83.5° N and 12° E to 27° E. The buoys recorded their positions either hourly 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 (Korosov 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 high latitudes.

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

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

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 × 87 m azimuth, and decreases the computational effort.

To apply the feature-tracking algorithm from Muckenhuber et al. (2016), the SAR backscatter values σ^0 have to be converted into intensity values i with $0 \leq i \leq 255$ for $i \in \mathbb{R}$. Before the conversion, we change the linear scaling of the raw backscatter values σ^0_{raw} to a logarithmic scaling and get the backscatter values $\sigma^0 = 10 * \lg \sigma^0_{\mathrm{raw}}$. The conversion is done by using Eq. (2) and setting all values outside the domain to 0 and 255.

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

The upper brightness boundary σ^0_{max} is set to the logarithm of the recommended value from Muckenhuber et al. (2016), i.e. $\log(0.013)$ and $\log(0.08)$ for HV and HH respectively. The lower boundary σ^0_{min} was chosen to be -3.25 (HV) and -2.5 (HH), since this was found to be a reasonable range of expected backscatter values.

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. Computational 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 and quality estimate. The algorithm consists of three main steps:

I Feature-tracking

The feature-tracking algorithm used in this work is adopted 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 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 two shortest Hamming distances is below 0.75.

II Pattern-matching

The used pattern-matching approach is based on the maximisation of the normalised cross correlation. The normalised cross correlation of two equally sized windows g and h is defined as:

$$NCC(g,h) = \frac{\sum_{i,j}(g_{ij}-\bar{g})(h_{ij}-\bar{h})}{\sqrt{\sum_{i,j}(g_{ij}-\bar{g})^2\sum_{i,j}(h_{ij}-\bar{h})^2}} \tag{3}$$

with g_{ij} (h_{ij}) representing the value of g (h) at the location i,j and $\bar{g}$ ($\bar{h}$) the mean value of g (h) (Hollands , 2012). Considering a window g from a SAR image and a window h that is moved with step size 1 pixel over a quadratic area of a consecutive SAR image results in a matrix with NCC values. The highest value in this matrix, i.e. the maximum cross correlation MCC, represents the location of the best match and serves as a quality estimate of the matching performance.

III Combination

After data pre-processing as described above, the feature-tracking algorithm from Muckenhuber et al. (2016) is applied with a maximum drift filter of 0.5 $\mathrm{m/s}$. This provides a number of un-evenly distributed vectors with start positions x_1, y_1 on the first image (SAR$_1$) and end positions x_2, y_2 on the subsequent image (SAR$_2$). To filter outliers, the starting point of each vector is simulated using two functions $f_{x_1}(x_2,y_2)$ and $f_{y_1}(x_2,y_2)$, that represent the least-squares solutions between x_1, y_1 and the third degree polynomial of x_2 and y_2. Vectors that have a start position x_1 or y_1 further than 100 pixels (8 km) away from the simulated point are removed.

The remaining feature-tracking vectors are used to estimate the drift on the entire first image, i.e. x_2 and y_2 values are provided for each pixel on SAR$_1$ (left and middle panel in Figure 1). The interpolation is constructed by triangulating between the start positions on SAR$_1$ and performing linear barycentric interpolation on each triangle to find x_2 and y_2 based on the three neighbouring feature-tracking vectors . To provide a drift estimate for the surrounding area, we extrapolate x_2 and y_2 using two functions $f_{x_2}(x_1,y_1)$ and $f_{y_2}(x_1,y_1)$, that are derived from the least-squares solutions between x_2, y_2 and a linear combination of x_1 and y_1.

This initial drift estimate is used to perform efficient pattern-matching based on normalised cross correlation on a pre-defined grid or chosen points of interest. Figure 2 shows a pattern-matching example from image pair 'Fram Strait' used in Muckenhuber et al. (2016).

A small template t_1 with a given size is taken around the point of interest from SAR$_1$. A larger template t_2 with centre at the location x_2, y_2 defined by the corresponding drift estimate from feature-tracking is taken from SAR$_2$. The size of t_2 is defined by the distance d to the nearest feature-tracking vector with a lower and upper threshold (d_{min}, d_{max}):

$$\mathrm{side}(t_2) = \mathrm{side}(t_1) + 2*d \tag{4}$$

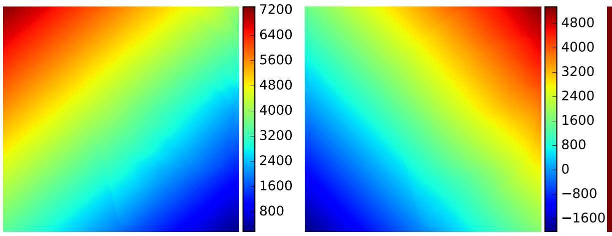

Figure 1. Estimated drift based on feature-tracking, and distance d to the nearest feature-tracking vector using the image pair 'Fram Strait' from Muckenhuber et al. (2016). The left and middle panel show the two components x_2 and y_2 of the estimated end positions on the second image (SAR$_2$) for each pixel on the first image (SAR$_1$). The right panel shows the distribution of d on SAR$_1$ with a lower and upper threshold d_{min}, d_{max}.

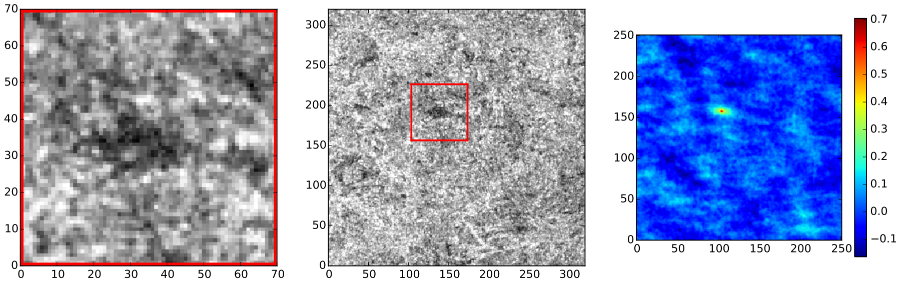

Figure 2. Pattern-matching using initial drift estimate from feature-tracking: Small template t_1 (left) around point of interest on SAR$_1$ is rotated from $-\beta$ to $+\beta$ and matched with large template t_2 (middle) from SAR$_2$, that has its centre at estimated end position x_2, y_2. The right contour plot shows the normalised cross correlation matrix of the rotation β^* that provided the highest maximum cross correlation $MCC(\beta^*)$. The estimated end position x_2, y_2 of this example has to be adjusted by -21 pixels, +32 pixels to fit with the location of $MCC(\beta^* = 2°) = 0.71$. NB: X and Y-axis represent pixel coordinates.

with $d_{min} \leq d \leq d_{max}$ for $d \in \mathbb{N}$ (example of d distribution in right panel of Figure 1). The two templates t_1 and t_2 are matched using maximisation of normalised cross correlation. Template t_1 is rotated starting with the initial rotation between the two Sentinel-1 images and going from $-\beta$ to $+\beta$ with step $\Delta\beta$. The result with the highest cross correlation value is returned.

In the last step, the small to medium scale displacement adjustments derived from pattern-matching are added to the estimated drift from feature-tracking. The maximum cross correlation values serve as individual quality measure for each drift vector and vectors that have a MCC value below the threshold MCC_{min} are removed.

3.3 Parameter tuning

The size of template t_1 and t_2 are crucial for a reliable drift result and for limiting the computational effort. As shown in Equation 4, the maximum size of t_2 is limited by the upper threshold d_{max}. To find the most useful values for d_{max} and the size of t_1, we varied the two parameters within the domains $20 \leq \text{side}(t_1) \leq 140$ with $\Delta\text{side}(t_1) = 2$, and $50 \leq d_{max} \leq 200$ with $\Delta d_{max} = 10$ (values given in pixels). For each combination, we calculated the drift on image pair 'Fram Strait' at the starting locations of the 350 manually derived validation vectors and compared the results using root mean square distance $RMSD$:

$$RMSD = \sqrt{\frac{\sum_i (u_i - U_i)^2 + (v_i - V_i)^2}{N}} \qquad (5)$$

The index i represents a vector pair consisting of a calculated vector and a validation vector at the same location. The eastward and northward drift components of the calculated vector are u_i and v_i. The validation vector has the corresponding drift components U_i and V_i. N is the total number of vector pairs.

During parameter tuning, the minimum value of the normalised cross correlation MCC_{min} was set to zero. We applied rotation on t_1 ranging from $-10°$ $(-\beta)$ to $+10°$ $(+\beta)$ with step size $\Delta\beta = 1°$. The lower threshold d_{min} was set to 20 pixels to allow for small scale drift adjustments close to the locations of feature-tracking vectors.

3.4 Comparison with buoy data

Sentinel-1 image pairs have been selected automatically according to position and timing of the buoy data. Drift vectors have been calculated 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:

$$D = \sqrt{(u - U)^2 + (v - V)^2} \qquad (6)$$

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 Logarithmic scaling of σ^0

The start and end positions of the feature-tracking vectors are tied to keypoints that are found during an initial processing step in the drift algorithm from Muckenhuber et al. (2016). Looking at the dis-

Table 1. Linear and logarithmic scaling of the backscatter values σ^0 in HV and HH polarisation: range, i.e. lower and upper brightness boundaries as used in Equation 2, and number of matched keypoints using image pair 'Fram Strait' (matched keypoints with maximum pixel value 255 in brackets).

	Linear scaling ($\sigma^0 = \sigma^0_{\text{raw}}$)	Logarithmic scaling ($\sigma^0 = 10 * \lg \sigma^0_{\text{raw}}$)
HV range [$\sigma^0_{\min}$, $\sigma^0_{\max}$]	[0, 0.013]	[-3.25; log(0.013)]
HH range [$\sigma^0_{\min}$, $\sigma^0_{\max}$]	[0, 0.08]	[-2.5; log(0.08)]
HV matched keypoints	11244 (4653)	15614 (2196)
HH matched keypoints	2840 (725)	4454 (317)

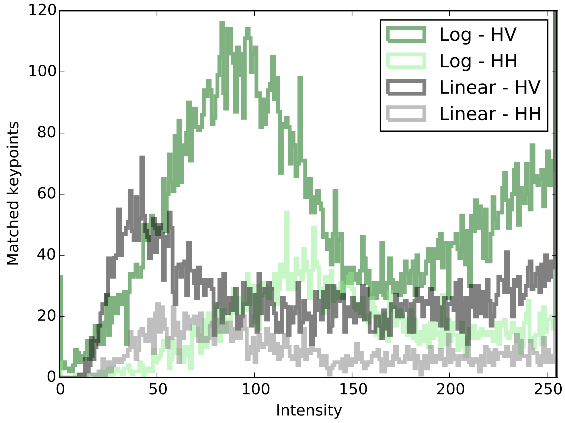

Figure 3. Number of matched keypoints for each intensity value in image pair 'Fram Strait' using linear (HV: black, HH: gray) and logarithmic scaling (HV: darkgreen, HH: lightgreen) for the backscatter values σ^0. Total numbers, and matches at intensity maximum 255 are shown in Table 1.

tribution of the intensity values among the matched keypoints, we found a strong peak at maximum intensity 255, meaning that most matched keypoints are recognised at very high backscatter values (NB: the number of vectors is equal to the number of matched keypoints divided by two). Before the conversion of the backscatter values σ^0 into the intensity values i (Equation 2), the scaling of σ^0 can be changed from linear ($\sigma^0 = \sigma^0_{\text{raw}}$) to logarithmic ($\sigma^0 = 10 * \lg \sigma^0_{\text{raw}}$). Table 1 and Figure 3 show the intensity value distribution of matched keypoints from image pair 'Fram Strait'. Using a logarithmic instead of a linear scaling provided a keypoint distribution that depends less on high peak values (the number of keypoints with intensity value 255 decreased from 5378 to 2513), while the total number of vectors increased from 7042 to 10034.

Table 2. Recommended parameter setting for sea ice drift retrieval from Sentinel-1 using the presented algorithm.

Parameter	Meaning	Recommended setting
$[\sigma^0_{\text{min}}, \sigma^0_{\text{max}}]$ (HH)	Brightness boundaries for HH channel	[-2.5, log(0.08)]
$[\sigma^0_{\text{min}}, \sigma^0_{\text{max}}]$ (HV)	Brightness boundaries for HV channel	[-3.25, log(0.013)]
side(t_1)	Size of template t_1	70 pixels
$[d_{min}, d_{max}]$	Influence domain for size of t_2	[20 pixels, 125 pixels]
MCC_{min}	Threshold for cross correlation	0.35
$[-\beta, +\beta, \Delta\beta]$	Rotation angle and increment	[-10°, +10°, 2°]

4.2 Parameter tuning

Figure 4 shows the $RMSD$ (Equation 5) calculated for image pair 'Fram Strait' using HH (left panel) and HV polarisation (right panel). Based on this evaluation, our experience with the algorithm behaviour, and considering a good compromise between computational efficiency and high quality of the resulting vector field, we recommend the parameter setting shown in Table 2. The following testing and validation process is conducted using this parameter setting.

4.3 Computational efficiency

The processing time depends on the parameter setting and the chosen vector distribution. Using the recommended parameter setting from Table 2, allows high-resolution sea ice drift retrieval from a Sentinel-1 image pair within a few minutes. Figure 5 depicts calculated ice drift vectors for the image pair 'Fram Strait' on a grid with 8 km (100 pixels) spacing. The corresponding processing times are shown in Table 3. 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 1145 vectors with a normalised cross correlation value above 0.35, is less than 3.5 minutes. NB: The vectors near Svalbard are located in the marginal ice zone. This is a very challenging area for drift algorithms based on consecutive images and the results have to be treated with caution.

4.4 Validation

The manually derived vectors from the image pair 'Fram Strait' have been compared with calculated drift vectors at the same locations (Figure 6) using $RMSD$ from Equation 5 and the recommended parameters from Table 2. 335 vectors with a normalised cross correlation value above 0.35 could be used for the comparison and the resulting $RMSD$ value is 540 m. NB: nine vectors close to the image border could not be used for comparison. The templates t_1 and t_2 are larger then the patch size used for simple feature-tracking (34 pixels) and hence, the restricted area at the image border is slightly increased.

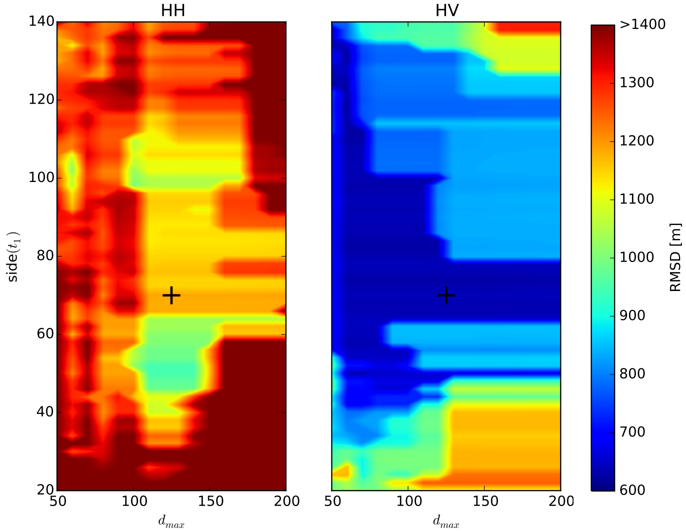

Figure 4. $RMSD$ (Equation 5) calculated for image pair 'Fram Strait' using different values for d_{max} (X-axis) and for the side length of template t_1 (Y-axis). Left panel shows HH polarisation and right panel HV polarisation. $RMSD$ values are given in meters, X and Y-axis represent pixel values. Black crosses mark recommended parameter setting: $d_{max} = 125$ pixels and side$(t_1) = 70$ pixels.

Table 3. Processing time for sea ice drift retrieval from image pair 'Fram Strait' on a grid with 8 km (100 pixels) spacing using HV polarisation (Figure 5).

Process	Time [s]
Create Nansat objects from Sentinel-1 image pair	21
Read matrixes from Nansat objects	49
I Feature-tracking	66
II Pattern-matching and III Combination	65
$\sum$ Sea ice drift retrieval	201

To compare the drift results from the algorithm with GPS positions from the N-ICE2015 buoy data set, 261 Sentinel-1 image pairs have been selected automatically for the considered time period 265 (15th January to 22nd April) and area (81° N to 83.5° N and 12° E to 27° E). 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. The satellite and buoy data sets provide 633 possible displacement pairs for comparison. Using the suggested threshold

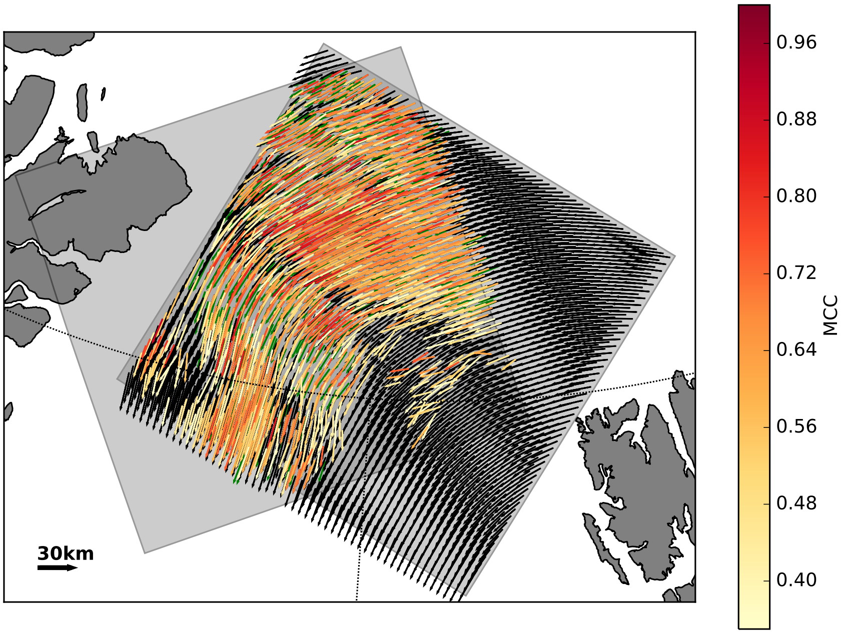

Figure 5. Sea ice drift from image pair 'Fram Strait' on a grid with 8 km spacing using HV polarisation. Black vectors indicate the initial drift estimate from feature-tracking. Coloured vectors are derived from combining feature-tracking and pattern-matching with a minimum cross correlation value $MCC_{min} = 0.35$. The colour indicates the maximum cross correlation MCC. A total of 1145 vectors have been found with a MCC value above 0.35.

for cross correlation $MCC_{min} = 0.35$ reduces the number of vector pairs to 540. The results of the comparison are shown in Figure 7. We found a logarithmic distribution of the distance D (Equation 6) with a peak at 300 m (3.75 pixels).

5 Discussion and outlook

Muckenhuber et al. (2016) compared their drift results (based on simple feature-tracking) and drift vectors from the Copernicus Marine Environment Monitoring Service (CMEMS) with the same manually drawn vectors as we use in Section 4. The CMEMS product is provided by the Technical University of Denmark (DTU), has a resolution of 10 km and is based on pattern-matching techniques (Pederson et al. (2015), http://www.seaice.dk/). Since the start locations of the drift vectors from these two algorithms do not coincide with the validation vectors, Muckenhuber et al. (2016) used the nearest neighbours for comparison and a maximum distance of 5 km. Table 4 shows the validation results from simple feature-tracking as done in Muckenhuber et al. (2016), pattern-matching

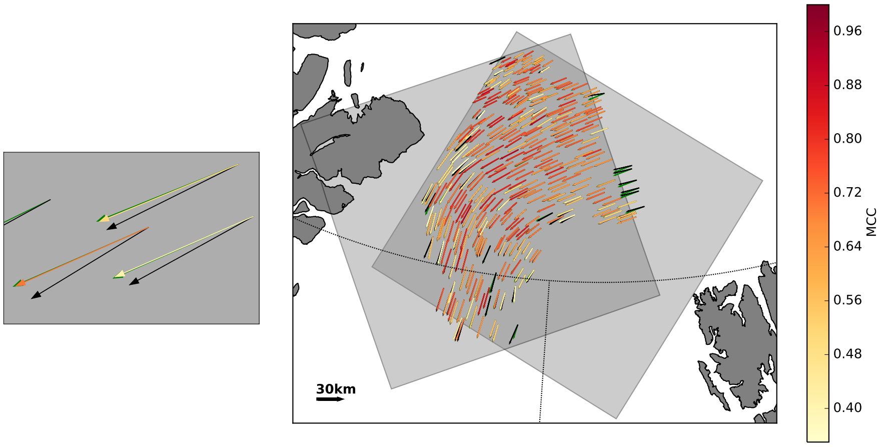

Figure 6. Sea ice drift derived from image pair 'Fram Strait' using HV polarisation: manually drawn validation vectors (green), initial drift estimate from feature-tracking (black) and vectors from combined feature-tracking and pattern-matching (colour according to maximum cross correlation MCC). The right panel shows the entire scene and the left panel depicts the algorithm procedure on an enlarged area.

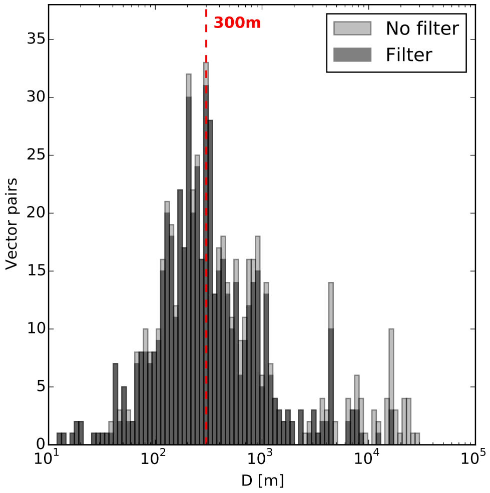

Figure 7. Buoy GPS data compared to calculated ice drift. Logarithmic histogram of distance D (Equation 6) with 100 bins between 10 m and 10^5 m. Light grey shows the unfiltered results ($MCC_{min} = 0$) and dark grey shows the results after using the suggested threshold for cross correlation $MCC_{min} = 0.35$. The peak of the distribution is marked with a red line at 300 m.

Table 4. Comparison of presented algorithm (combined feature-tracking + pattern-matching), simple feature-tracking as done in Muckenhuber et al. (2016), and CMEMS data using image pair 'Fram Strait' and 350 manually derived drift vectors as validation. $RMSD$ is the root mean square distance from Equation 5. The # vector pairs refers to the number of used vector pairs for comparison, i.e. vector pairs with maximum 5 km distance. The average distance is measured between the start positions of the validation vectors and the corresponding nearest neighbour vectors from the algorithm.

Algorithm	$RMSD$ [m]	# vector pairs	Average distance [m]
Feature-tracking + pattern-matching	540	335	0
Feature-tracking	563	314	1702±1325
CMEMS (pattern-matching)	1690	201	3440±1105

as done by DTU for CMEMS, and the presented combined feature-tracking + pattern-matching algorithm. The combined algorithm provides the highest accuracy (represented by $RMSD = 540$ m) and the highest number of vectors pairs that can be used for comparison (335). Unlike simple feautre-tracking, the combined algorithm allows to choose the positions of the drift vectors, which makes it possible to place them at the same locations as the validation vectors. This is represented by an average distance of 0 m. As discussed in Muckenhuber et al. (2016), the manually drawn vectors cannot be considered as perfect validation, since there might be an error introduced during the manual ice drift identification. Hence, we expect the error originating from the combined algorithm to be less than 540 m.

To further estimate the accuracy of the algorithm, we compared the drift results from 261 Sentinel-1 image pairs with corresponding GPS positions from the N-ICE buoy data set. We found a logarithmic error distribution with a peak at 300 m (Figure 7). 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 = 70 pixels, see Table 2), 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.

Hence, the actual error of the presented algorithm is expected to be even lower than 300 m. This means, that the algorithm accuracy is in the scale of the satellite image resolution.

A main advantage of the combined algorithm compared to simple feature-tracking, is the user defined 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 computational effort according to the individual needs.

The recommended parameters shown in Table 2 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 of sea ice group at NERSC is currently developing a new pre-processing step to remove thermal noise on HV images. 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.

Having a computational 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 nest task is to combine the different timings of the individual image pairs in a most useful way. This task is linked to the question how sea ice displacement relates to real sea ice velocity. Having more frequent satellite acquisitions, as we will get with the Sentinel-1 satellite constellation, enables to derive displacements for shorter time gaps and the calculated vectors are getting closer to the real sea ice velocity. As part of a scientific cruise with KV-Svalbard in July 2016, we deployed GPS trackers on loose ice floes and pack-ice in Fram Strait. The trackers send their position every 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 displacements from subsequent satellite images relate to real sea ice velocity.

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 the developer group of the Earth observation toolbox Nansat and Anton Korosov for his inputs. We also 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.

References

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, SP-1322/1, ISBN: 978-92-9221-418-0, ISSN: 0379-6566, 2012.

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

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.

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.

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.

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

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

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.

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.

Schubert, A., Small, D., Meier, E., Miranda, N., and Geudtner, D.: Spaceborne Sar Product Geolocation Accuracy: A Sentinel-1 Update, Geoscience and Remote Sensing Symposium (IGARSS), IEEE International, 2675–2678, doi=10.1109/IGARSS.2014.6947025, 2014

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.