Ocean & Sea Ice SAF

Algorithm Theoretical Basis Document for the OSI SAF Low Resolution Sea Ice Drift Product

GBL LR SID — OSI-405-c

Version 1.3 — May 2016

Thomas Lavergne

The EUMETSAT Network of Satellite Application Facilities

OSI SAF
Ocean and Sea Ice

This page is left blank intentionally.

Documentation Change Record:

Document version	Date	Author	Description
v0.9	26.03.2009	TL	Submitted to review
v1.0	29.04.2011	TL	Small updates and typos
v1.1	15.04.2015	TL	Describe multi-sensor merging algorithm, update for new sensors.
v1.2	14.12.2015	TL	Describe algorithms for uncertainties (OSI-405-c PCR).
v1.3	26.05.2016	TL	Remove preliminary validation results from ATBD as part of ORR.

Table of contents

1. Introduction

1.1 Scope of this document

This Algorithm Theoretical Basis Document shall describe the computing steps implemented in the Low Resolution Sea Ice Drift processing software running as part of the EUMETSAT Ocean and Sea Ice Satellite Application Facility (OSI SAF). It aims at introducing and, to some extent, giving justification for the scientific and algorithmic choices which led to near-real-time sea ice motion processing in the EUMETSAT OSI SAF.

Technical aspect of the product (file data format, timeliness, etc...) are rather to be found in the Product User's Manual (PUM). Results from validation against ground truth sea ice drift measurements are gathered in (VALR).

General information on the EUMETSAT OSI SAF are available from www.osi-saf.org.

After introducing common notation in the remaining of the current chapter, the necessary pre-processing steps are described in chapter 2, the motion tracking algorithm in chapter 3 and finally the multi-sensor analysis algorithm in chapter 4. Conclusions are outlined in chapter 5.

1.2 Definitions and common notations

1.2.1 Date and Time

As any image-based motion tracking algorithms, the low resolution sea ice drift processing quantifies the amount of motion between two time stamps by analysing the change in intensity patterns between two images. Those are referred to as $\mathcal{I}_0$ (start image) and $\mathcal{I}_1$ (stop image). Unless specified otherwise, subscript 0 (1) always refers to the *start* (*stop*) time stamp of the displacement.

The OSI SAF low resolution sea ice drift product is computed between daily maps of satellite signals, as opposed to between individual satellite swaths. $\mathbf{D}_0$ ($\mathbf{D}_1$) is the date for the start (stop) of the displacement. $\mathbf{D}$@hh can be used to express a time stamp in a specific day. All hours are *UTC*.

1.2.2 Grids and spatial projection

Three grids are of interest to us: `nh100`, `nh125` and `nh625`. They vary in extent and grid resolution, but share the same Earth mapping projection, that is a polar stereographic mapping, true scale at $70N$ and central meridian at $45W$. The Earth datum is an ellipsoid with radius 6378273 m and 6356889.44891 m (Aaboe et al. (2016); WWW-NSIDC-PGRID).

Tag	n_x	n_y	$\mathbf{A}_x$ [km]	$\mathbf{A}_y$ [km]	$\mathbf{B}_x$ [km]	$\mathbf{B}_y$ [km]
nh100	760	1120	10.0	10.0	-3850	5850
nh125	608	896	12.5	12.5	-3850	5850
nh625	119	177	62.5	62.5	-3750	5750

Table 1: Grid characteristics for nh100, nh125 and nh625. $\mathbf{A}_x$ ($\mathbf{A}_y$) is the grid spacing and $(\mathbf{B}_x, \mathbf{B}_y)$ the coordinates of the upper-left corner of the grids when the central meridian line points towards the bottom of the figure.

Table 1 holds the characteristics of the three grids that are used in the low resolution sea ice drift processing. nh100 is one of the official grids for the OSI SAF sea ice products (concentration, edge, type) (Aaboe et al. (2016)). nh625 is the grid for the low resolution sea ice drift product (*output* grid). nh125 is an intermediate grid used for the processing of the daily images.

Appendix A describes the mapping formulas from a geographical point (λ, ϕ) into a (x, y) point in the projection plane and, eventually, into $[i, j]$ indexes in the two-dimensional array of the grid. Those formulas are used in the rest of the document each time a remapping is needed.

Cell coordinates always point to the center of a grid cell. Thus, point $[i, j]$ is at the centre of a square grid cell (pixel) whose corner points are $(i - \frac{1}{2}, j - \frac{1}{2})$, $(i - \frac{1}{2}, j + \frac{1}{2})$, *etc*...

1.2.3 Summer season

The OSI SAF low resolution sea ice drift product has global coverage, and thus cover all sea ice in the Northern Hemisphere (NH) and Southern Hemisphere (SH). The quality of the drift estimates vary with season, and it is convenient to refer to winter and summer period. The definition of these seasons is different in both hemispheres:

- In NH, winter is from October 1st to April 30th. Summer is from May 1st to September 30th. May and September are transition months with a ”core” summer season (June, July, August).
- In SH, winter is from April 1st to October 31st. Summer is from November 1st to March 31st. November and March are transition months with a ”core” summer season (December, January, February).

1.2.4 Unit for vectors and uncertainties

Unless otherwise specified, all vector components $\mathbf{d}X$and $\mathbf{d}Y$stand for the sea ice displacement *after a 48 hours displacement* and have thus unit km (and not km/day).

The associated uncertainties and validation statistics are presented as 1 standard deviation σ of the components $\mathbf{d}X$and $\mathbf{d}Y$(still *after a 48 hours displacement*). They thus also have unit of km (and not km/day).

2. Preprocessing of satellite images

2.1 Land and open water masks

Land and open-water masks for the start and stop time stamps are taken from the OSI SAF operational sea ice edge product (OSI-402). Those are documented in Aaboe et al. (2016). They contain enough information to decide if a geographical location is:

1. over land;
2. close to coast;
3. over open ice;
4. over closed ice;
5. in open water.

The ice edge product files for $\mathbf{D}_0$ and $\mathbf{D}_1$ are remapped from `nh100` to `nh125` and regridded using a nearest-neighbour selection. During this step, the polar observation hole (north of $87.5N$) is filled with closed ice.

2.2 Satellite data pre-processing

2.2.1 Processing in swath projection

Depending on the sensor being processed, different steps are taken on the swath files.

Angular correction of ASCAT σ^0

C-band σ^0 as those acquired by Metop ASCAT exhibit a strong dependency with respect to the zenithal incidence angle (Ezraty and Cavanié (1999)). Computing a daily averaged map of σ^0 for subsequent analysis thus necessarily requires a zenithal angle correction step. Although advanced methods exist for such a correction, a simple approach is chosen which proves good enough for the purpose of extracting sea ice motion vectors over closed ice.

σ^0_θ triplets (*fore*, *mid* and *aft* antennas) from ASCAT Level 1b spatially averaged high resolution product (12.5km sampling rate) are corrected to $45°$ incidence angle by the linear relationship in equation 2.1:

$$\sigma^0_c = \sigma^0_\theta - 0.19 \times (45 - \theta) \tag{2.1}$$

From the triplet of corrected values, the one whose incidence angle is closest to $45°$ is kept.

More information on the ASCAT instrument and products are given in ASCAT PG.

AMSR2 and SSMIS brightness temperatures

The brightness temperatures for the AMSR-E and AMSR2 at 37 GHz channels, the SSM/I at 85 GHz channels, and the SSMIS at 91 GHz channels are used as-is.

2.2.2 Daily averaging of swath data

Once the swath-based corrections have been applied (section 2.2), a daily average map of satellite signal is computed from all swath available in the range $\mathbf{D}@00$ to $\mathbf{D}@24$. Each corrected satellite value is remapped into the `nh125` grid and a series of weights are computed and associated to it.

Irrespective of the sensor being processed, we note $\gamma_t[i,j]$ an individual, corrected, satellite observation with sensing time t (t in hours, $t \in [0:24[$), remapped in `nh125` at cell $[i,j]$.

Temporal weight during the integration window

For reducing the span of sensing time contributing to each location, yet ensuring spatial continuity, a temporal weighting function $\mathcal{W}_T(t)$ is applied:

$$W_T(t) = -\frac{1}{12} \times |12 - t| + 1 \tag{2.2}$$

As can be seen from equation 2.2, $W_T(0) = W_T(24) = 0$ and $W_T(12) = 1$, with linear variations around the central $t = 12$ value.

Spatial weight into neighbouring pixels

Because the spatial sampling of the sensors we are interested in is close to the remapping grid resolution (12.5km), it is necessary to implement a limited area spatial weighting for each $\gamma_t[i,j]$. In this scheme, each $\gamma_t[i,j]$ contributes to cell $[i,j]$ as well as to the 8 neighbouring grid cells $[n,m]$ with weight $\mathcal{W}_S^{i,j}(n,m)$:

$$\begin{aligned} \ell(n,m,i,j) &= \sqrt{(n-i)^2 + (m-j)^2} \\ \mathcal{W}_S^{i,j}(n,m) &= \exp(-0.5 \times \frac{\ell^2}{0.75^2}) \\ n \in [i-1, i+1] \quad &\text{and} \quad m \in [j-1, j+1] \end{aligned} \tag{2.3}$$

Equation 2.3 takes values between 0 and 1 that can be pre-computed in a table:

0,03	0,41	0,03
0,41	1,00	0,41
0,03	0,41	0,03

The weighting function defined by equation 2.3 does not take into account the shape and orientation of the actual footprint nor does it acknowledge that grid cells are not spaced evenly on the Earth sphere. Nevertheless, it proves enough for the purpose of filling the gaps induced by a simplistic nearest-neighbour gridding strategy.

Weighting equation

Finally, the satellite signal value assigned to $\mathcal{I}[i,j]$ is computed as a weighted average over several $\gamma_t[n,m]$, $t \in [0,24[$ and $(n,m) \in [i-1,i+1] \times [j-1,j+1]$:

$$\begin{aligned} \mathcal{I}[i,j] &= \frac{1}{\mathcal{K}} \sum_{n=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{k=1}^{k(n,m)} W_S^{n,m}[i,j] \times W_T(t(k)) \times \gamma_{t(k)}[n,m] \\ \mathcal{K} &= \sum_{n=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{k=1}^{k(n,m)} W_S^{n,m}[i,j] \times W_T(t(k)) \end{aligned} \tag{2.4}$$

$k(n,m)$ is the number of γ_t that contribute to cell $[n,m]$. $t(k)$ is the sensing time associated to each of them.

At this stage, there is no special treatment for pixels classified as land, sea-ice or open water. Equation 2.4 is adapted along the borders of the grid.

Mean sensing time

Equation 2.4 is not only applied to compute $\mathcal{I}[i,j]$ from γ_t but for the *average sensing time* $\mathcal{T}[i,j]$ as well. The latter is an important quantity as it translates into position-dependent start and stop time for the drift vectors. It is computed by equation 2.4 only with $t(k)[n,m]$ in place for $\gamma_{t(k)}[n,m]$.

Effect of the temporal weighting function $W_T(t)$

The temporal weighting introduced in section 2.2.2 act as a sharpening filter towards the central time of the averaging period. It aims at reducing the *motion blur* which arises from the steady displacement of sea ice during the averaging period of the daily image.

2.2.3 Feature enhancement with Laplacian filtering

As in Ezraty et al. (2007b), a Laplacian filter is applied to the average image computed in section 2.2.2. It aims at:

- removing intensity gradients across the image plane as well as between the start and stop image;
- enhancing intensity patterns;

The Laplacian (or Laplace operator) ∇^2 is closely linked to the second derivatives of multi-dimensional functions (WWW-WIKIP-LAPLACE) and is computed as:

$$\begin{aligned} \mathcal{L}[i,j] = &\frac{1}{N^+} \sum_{n=i-1}^{i+1} \sum_{m=i-1}^{i+1} \delta_{\text{na}}(n,m)\delta_{\text{ice}}(n,m)\delta_1^{i,j}(n,m)\mathcal{I}[n,m] \\ &- \frac{1}{N^-} \sum_{n=i-2}^{i+2} \sum_{m=i-2}^{i+2} \delta_{\text{na}}(n,m)\delta_{\text{ice}}(n,m)\delta_2^{i,j}(n,m)\mathcal{I}[n,m] \end{aligned} \tag{2.5}$$

with

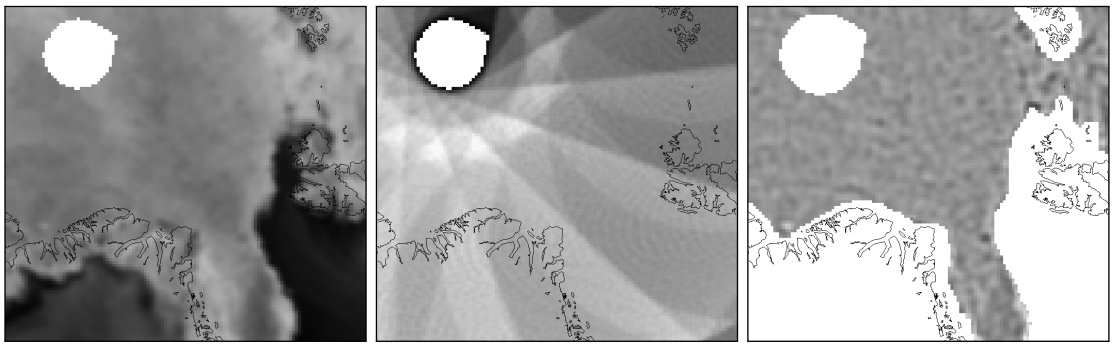
$$\begin{aligned}
\delta_k^{i,j}(n,m) &= \begin{cases} 1 & \text{if } |i-n| = k \text{ or } |j-m| = k; \\ 0 & \text{otherwise.} \end{cases} \\
N^+ &= \sum_{n=i-1}^{i+1} \sum_{m=i-1}^{i+1} \delta_{\text{na}}(n,m)\delta_{\text{ice}}(n,m)\delta_1^{i,j}(n,m) \leq 8 \\
N^- &= \sum_{n=i-1}^{i+1} \sum_{m=i-1}^{i+1} \delta_{\text{na}}(n,m)\delta_{\text{ice}}(n,m)\delta_2^{i,j}(n,m) \leq 16
\end{aligned}$$

In equation 2.5, $\delta_{\text{na}}(n,m)$ has value 0 if $\mathcal{I}[n,m]$ is non available (a missing value from the swath) and $\delta_{\text{ice}}(n,m)$ is 1 only over *open or closed ice*, as specified by the ice mask (section 2.1). It means that only *valid*, *sea ice* pixels enter the Laplacian field in order to limit spurious features along the ice edge, coastline or close to the polar observation hole.

$\mathcal{L}[i,j]$ is only computed if the centre cell $[i,j]$ is itself over sea ice, *i.e.* $\delta_{\text{ice}}(i,j) = 1$.

In the event when $N^+ < 5$ or $N^- < 9$, not enough pixels are available for computing $\mathcal{L}$ and a missing value is stored at grid cell $[i,j]$.

Note that, conversely to Ezraty et al. (2007b), no median or further filtering is applied on the Laplacian image.



(a) (b) (c)

Figure 1: Example intensity (a), sensing time (b) and Laplacian (c) for a daily image of the Arctic Ocean (near Fram Strait). The satellite signal is from the AMSR-E instrument, 37GHz H-pol channel on December, 21st 2008. The observation hole close to North Pole is clearly visible in the upper left corners. Straight line patterns in sub-image (b) correspond to overlapping swaths. The Laplacian filter (c) successfully enhanced the intensity features of (a).

Example images of the information available at the end of this pre-processing step are given in figure 1. The ice motion tracking algorithm is applied on the pair of $\mathcal{L}_0$ and $\mathcal{L}_1$ images. It is described in chapter 3.

3. Motion tracking

3.1 Introduction

As for many other motion tracking methods applied in geophysics (Maslanik et al. (1998); Notarstefano et al. (2007)), sea ice drift is tracked from a pair of images, with a block-based strategy. Each block (*aka feature*, *sub-image*, *etc*...) is composed of a limited ensemble of pixels from the first image (the *reference* block) and centred at a tracking location, for which the most similar block in the second image (the *candidate* block) is looked for. The degree of similarity is assessed by a metric, which often is the correlation coefficient between the reference and candidate blocks. The maximum correlation indicates the best match and the two-dimensional offset between the centre points of the two blocks is the drift vector.

The description given above applies to the well known MCC (Maximum Cross Correlation) technique which has been successfully applied by many investigators (Emery et al. (1991); Ezraty et al. (2008); Haarpaintner (2006); Notarstefano et al. (2007); Schmetz et al. (1993), among others). In the MCC, the search for the best candidate block is *discrete* and *exhaustive*. Discrete since the offsets between the centre points are in whole number of pixels and exhaustive because all candidate blocks are evaluated before the best can be chosen.

The same description applies to the CMCC (Continuous Maximum Cross Correlation) method. CMCC is the strategy developed and implemented for the low resolution sea ice drift product of the OSI SAF. Conversely to the MCC, however, the search for the best candidate is performed in a continuous manner over the two-dimensional plane and, as a consequence, the search algorithm is not exhaustive.

Although more complex to implement and, by nature, less robust than the MCC technique, the CMCC has the advantage of removing the *quantization noise* which has hindered the retrieval of smooth motion vector fields when the time span between the images is shortened (Ezraty et al. (2007a); Haarpaintner (2006); Kwok et al. (1998)).

In the following sections, the motion tracking algorithm is further described. It consists in three steps:

- selection of tracking locations and preliminary screening (section 3.2);
- block based maximisation of the correlation metric via the CMCC (section 3.3);
- filtering and correction step (section 3.5).

Finally, section 3.6 describes the approach to providing per grid-cell uncertainties.

3.2 Selection of tracking locations and preliminary screening

The start locations for the drift arrows, *aka* the tracking locations, are distributed over grid `nh625`. Its resolution is 5 times lower than the one of the image grid, `nh125`. Indeed grid points in `nh625` are collocated with every 5th of those in `nh125`.

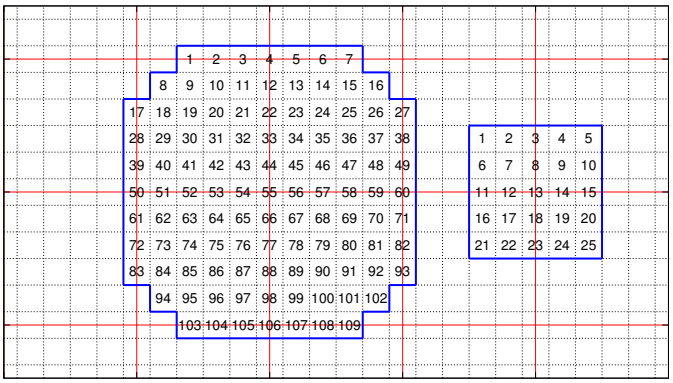


Figure 2: Shapes of the nominal (left) and reduced (right) blocks used in the motion tracking algorithm. The red grid intersects at `nh625` locations while the black dashed lines are contour of the image pixels, whose centre points are on grid `nh125`. The reduced block is used when there is not enough valid sea ice image pixels around the centre location.

Figure 2 draws the shape of the blocks used in the motion tracking step. Both a nominal (left) and a reduced (right) block shape can be activated. They mimic circular areas of radius 75km (37.5km) centred on North Pole. In those blocks, pixel number 55 (13) are the centre pixels. In figure 2 the nominal block is included in the one used by Ezraty et al. (2007b), a 11x11 pixels square area. As in Haarpaintner (2006), the reduced block is used close to the ice edge, the coastline or an area with missing satellite data. The aim is to try and estimate ice drift vectors as close as possible to those cases. The reduced block is always used in place of the nominal one, not at the same time.

Before any motion vector is retrieved, a screening algorithm browse through all the start locations on `nh625` and filter out all grid points which do not qualify for motion extraction. In the process, the shape of the block that is to be used at each drift location is decided upon.

Each step of the screening process is applied on the list of drift locations that passed all the previous steps. In the event where no grid points are left after one of the screening steps, the motion processing is skipped. The first step is applied to the list of all the drift locations from `nh625`.

Masking of land pixel (step 1) Blocks whose centre pixel is over land are discarded;

Masking of pixels with not enough ice (step 2) The start and stop blocks of the two ice masks are loaded. Discard the grid locations whose blocks are not totally over ice.

Masking of pixels with missing data (step 3) The start and stop blocks of the Laplacian images are loaded. Discard the grid locations whose blocks hold missing data.

Detailed flow charts for each step are given in appendix B.

Start locations that are not discarded are passed on to the motion tracking processing, along with the information on which block shape to use for each of them.

3.3 Block based maximisation of the correlation metric

3.3.1 Notations

We note $\mathcal{L}_0(x,y)[i]$ the i^{th} pixel of the *start* sub-image centred at point (x,y), extracted from the $\mathcal{L}_0$ image. (x,y) are the coordinates expressed in the underpinning projection, see appendix A.

As can be seen from figure 2, the total number of pixels in a sub-image depends on the block shape, $N_n = 109$ and $N_r = 25$. Letter N is used when no distinction between the block shapes is made.

The mean and standard deviation values for a given sub-image are:

$$\begin{aligned} \langle \mathcal{L}_0(x,y) \rangle &= \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_0(x,y)[i] \\ \sigma(\mathcal{L}_0(x,y)) &= \sqrt{\langle \mathcal{L}_0^2(x,y) \rangle - \langle \mathcal{L}_0(x,y) \rangle^2} \end{aligned}$$

Similar values can be computed for a *stop* sub-image centred at (u,v): $\langle \mathcal{L}_1(u,v) \rangle$ and $\sigma(\mathcal{L}_1(u,v))$.

As introduced in section 3.1, the match between a start and a stop sub-image is evaluated via the correlation metric:

$$\rho(x,y,\delta_x,\delta_y) = \frac{\sum_{i=1}^{N} (\mathcal{L}_0(x,y)[i] - \langle \mathcal{L}_0(x,y) \rangle)(\mathcal{L}_1(x+\delta_x, y+\delta_y)[i] - \langle \mathcal{L}_1(x+\delta_x, y+\delta_y) \rangle)}{\sigma(\mathcal{L}_0(x,y)) \sigma(\mathcal{L}_1(x+\delta_x, y+\delta_y))} \tag{3.1}$$

By construction, $\rho(x,y,\delta_x,\delta_y)$ takes values between -1 and $+1$. High values indicate a good match between the sub-images. This is further interpreted as having found the offsets $\delta_x = u - x$ and $\delta_y = v - y$ which best explain the local change in intensity between the two sub-images. (δ_x, δ_y) is the drift vector.

3.3.2 On the fly computation of *virtual* sub-images

Pixels of the candidate block $\mathcal{L}_1(x+\delta_x, y+\delta_y)$ are computed from bi-linear interpolations of the pixels of $\mathcal{L}_1$. For example, $\mathcal{L}_1(u,v)[i]$ is given by:

$$\begin{aligned} \mathcal{L}_1(u,v)[i] = &(1-\epsilon_u) \times (1-\epsilon_v) \times \mathcal{L}_1(\bar{u},\bar{v})[i] \\ &+ (1-\epsilon_u) \times \epsilon_v \times \mathcal{L}_1(\bar{u},\bar{v}+s_v)[i] \\ &+ \epsilon_u \times (1-\epsilon_v) \times \mathcal{L}_1(\bar{u}+s_u,\bar{v})[i] \\ &+ \epsilon_u \times \epsilon_v \times \mathcal{L}_1(\bar{u}+s_u,\bar{v}+s_v)[i] \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} \bar{t} &= \mathrm{Trunc}(t) \\ \epsilon_t &= |t - \bar{t}| \\ s_t &= \frac{t}{|t|} \end{aligned}$$

For example, for $t = -2.8$, $\bar{t} = -2$, $\epsilon_t = 0.8$ and $s_t = -1$. Equation 3.2 permits computing *virtual* sub-images at continuously varying centre points (u, v) and thus building a continuous optimisation framework to the estimation of motion vectors from a pair of images.

3.4 Continuous optimisation of motion vectors

Finding the motion vector (δ_x, δ_y) at position (x, y) can be expressed as the following maximisation problem:

$$\max_{(x,y)\in\mathcal{D}} \rho(x, y, \delta_x, \delta_y) \tag{3.3}$$

which is solved at all (x, y) points the motion vector is searched for. Each optimisation is conducted independently from the others. $\mathcal{D}$ is a validity domain for (δ_x, δ_y). Equation 3.3 thus defines a two dimensional optimisation problem with domain constraint.

3.4.1 Maximisation of $\rho(x, y, \delta_x, \delta_y)$

Equation 3.3 is solved by the Nelder Mead algorithm (Nelder and Mead (1968); Lagarias et al. (1998)). This algorithm was chosen since it is simple to implement and does not require computing the derivatives. It furthermore has good convergence and computational properties in problems with low dimensionality.

Initialisation

Starting points for the optimisation are sampled on a length-angle regular grid around point $(0, 0)$ as on figure 3.

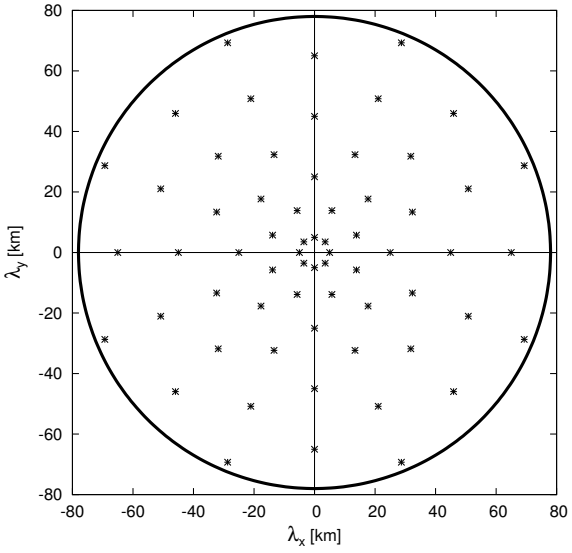


Figure 3: Location of the preliminary function evaluations for choosing starting points to the Nelder Mead algorithm. The circle locates the limit of validity domain $\mathbf{D}$.

The length increment is set to 10km and the angular increment to $45°$. The circle has radius $\mathbf{L}$, the maximum drift distance defining $\mathcal{D}$.

$\rho(x, y, \delta_x, \delta_y)$ is computed at each of those points and the best 3 vertexes are kept for initialising the Nelder Mead optimisation.

Numerical convergence test

Termination and convergence is tested upon via a relative difference of function values at the current *best* and *worst* vertexes, f_b and f_w. Specifically, the algorithm is said to converge if and only if $|f_b - f_w| < (f_b + f_w) \times \tau + \epsilon$, with τ and ϵ small and positive floating point values. As a safeguard, the maximum number of iterations is set to 1000. No convergence test is performed on the size and shape of the final simplex.

3.4.2 Implementation of the validity domain $\mathcal{D}$

$\mathcal{D}$ is a disc shaped domain expressing the *a-priori* knowledge we bring to the optimisation problem. Its purpose is to limit the search area for the solution vector during the optimisation process. It is defined by a centre point (x_c, y_c) and radius $\mathbf{L}$.

$$(\delta_x, \delta_y) \in \mathcal{D}_{x_c,y_c} \Leftrightarrow d(x_c, y_c; \delta_x, \delta_y) < \mathbf{L} \tag{3.4}$$

In equation 3.4, $d(x_c, y_c; \delta_x, \delta_y)$ is the distance (along the Earth surface) between the centre point of $\mathcal{D}$ and the tip of the drift vector (δ_x, δ_y) (see equation A.3). (x_c, y_c) represents our best *a-priori* knowledge at the time of performing the optimisation. It is initially set to $(0, 0)$.

Equation 3.4 cannot be used *as is* in the optimisation routine since it leads to abrupt and non-linear behaviour. $\mathcal{D}$ is instead implemented as a *soft* constraint based on a mono-dimensional sigmoid function $W(d)$:

$$W(d) = \frac{1}{1 + e^{k(d-\mathbf{L})}} \tag{3.5}$$

In equation 3.5, k is a parameter controlling the steepness of the sigmoid around the cut-off value $\mathbf{L}$. By construction, $W(\mathbf{L}) = 0.5$. By using a large enough value for k, the W can be made arbitrarily close to the Heaviside step function, yet remaining smooth and continuous.

Equation 3.6 and figure 4 illustrate how the penalty is applied to the correlation function $\rho(x, y, \delta_x, \delta_y)$.

$$\rho_D(x, y, \delta_x, \delta_y) = (\rho(x, y, \delta_x, \delta_y) + 1) \times W(d(x_c, y_c; \delta_x, \delta_y)) - 1 \tag{3.6}$$

Figure 4 plots a mono-dimensional example of applying a sigmoid penalty function to a synthetic correlation function. Evaluations for x lower than L are dominated by the correlation value $\rho(x)$ while those occurring outside the domain (x larger than L) return very bad scores, that is close to -1.

In equation 3.6, ρ_D is the penalised correlation function. Finding the maximum of ρ_D is taken as a proxy for solving the original, constrained, optimisation problem of equation 3.3. ρ_D is the functional entering the Nelder Mead algorithm.

It is customary to compute $\mathbf{L}$ as a maximum expected velocity v_{max}, multiplied by the time separation between the two images $\mathbf{D}_1$ - $\mathbf{D}_0$. $\mathbf{L}$ is thus the maximum expected straight-line distance that can be covered in the given time.

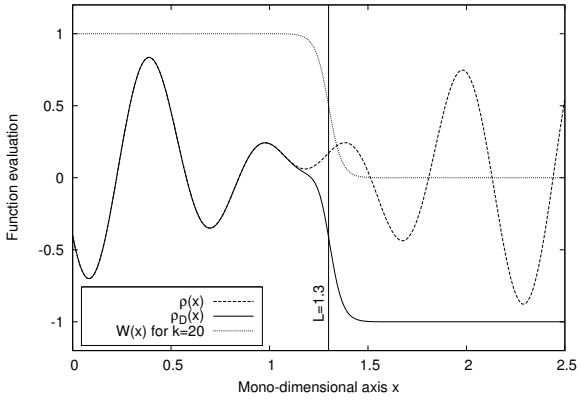


Figure 4: Example soft constraint implemented with a sigmoid penalisation function W and its application on a synthetic, mono-dimensional correlation signal ρ. The $\mathbf{L}$ parameter is 1.3 and k is 20.

For the operational implementation of the OSI-405 product, $\mathbf{D}_1$ - $\mathbf{D}_0$ is 2 days, and v_{max} is $0.45m.s^{-1}$. Individual ice floes can be recorded with higher hourly velocities in dynamic areas, however v_{max} corresponds to a max speed averaged over 48 hours, and a spatial extent of approximately 100 km. This value of v_{max} is a compromise between allowing very long drift distances, controlling computation time, and the number of gross erroneous vectors (see section below).

3.5 Detection and correction of erroneous vectors

Once the motion tracking processing described above has been applied to each of the start position selected by the preliminary checks in section 3.2, a filtering step is taken to correct or remove obviously erroneous vectors.

Causes for those erroneous vectors include:

1. convergence of the Nelder Mead algorithm in a local (non-global) maximum;
2. noise in the sub-images;
3. edge effects in the sub-images.

Whatever the reason be, the filtering step we implement is based on the distance from individual displacement vectors to the average of its neighbouring vectors. If this distance is less than a fixed threshold, the displacement vector being tested is validated and another vector is tested upon. Otherwise, a new motion tracking optimisation is triggered. The Nelder Mead algorithm is initialised and run like in the previous section, except that the validity domain $\mathbf{D}$ is adapted (center and radius) to translate the new constrain.

Let Δ_{avg} be the distance between the tip of the current drift vector (δ_x, δ_y) and the tip of the zonal average drift vector $(\delta_x^{\text{avg}}, \delta_y^{\text{avg}})$. The average drift vector is computed from the 8 neighbouring drift vectors, that is the 8 closest vectors *not including the current one*. The local $\mathbf{D}$ domain is then the disc with centre $(\delta_x^{\text{avg}}, \delta_y^{\text{avg}})$ and radius $\Delta_{\text{max}}^{\text{avg}}$. In the current implementation, $\Delta_{\text{max}}^{\text{avg}}$ is set to 10km. Neighbouring vectors with a maximum correlation

value of less than 0.5 are not used, to avoid degrading the average drift field with possibly wrong estimates.

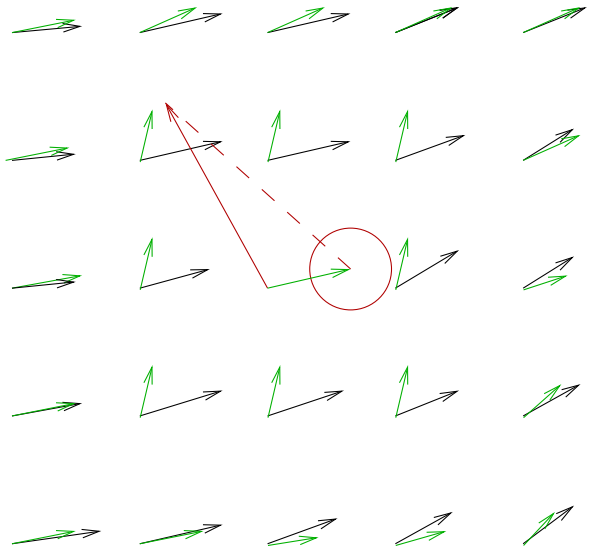


Figure 5: Example case where the current drift vector (in red) is obviously erroneous, considered the smoother vector field from the first estimate from CMCC (in black). The locally averaged vector field is plotted in green. Δ_{avg} is the length of the dashed red line. The red disc has radius $\Delta_{\text{max}}^{\text{avg}}$ and is the validity domain $\mathbf{D}$ that is used to re-optimise the drift vector.

Figure 5 illustrates a typical case where a single erroneous vector is surrounded by a smooth vector field. Since the central estimate is not used in the average, isolated wrong vectors stand out very easily in terms of Δ_{avg}.

During this second optimisation, the search for the maximum is limited to the area eclosed by the red circle. If a satisfying maximum correlation is found inside $\mathbf{D}$ it is kept and the surrounding average vectors are immediately updated, as well as each Δ_{avg} lengths. If the constrained optimisation does not converge or if the new vector does not have a good enough maximum correlation value, both the old and new vectors are discarded and the average vectors, as well as Δ_{avg} at the neighbouring locations are updated.

Although the method described above works in many cases, it sometimes fail when several erroneous vectors are close one to each other. This happens especially when noise dominates the signal in a large region of one of the image. If the case, the order in which the vectors are corrected has an influence on the final efficiency for the filtering.

To minimize this influence, motion vectors are first sorted from the largest to the shortest Δ_{avg} and the filtering is applied to the vector exhibiting the worst of those distances. Since, changing a vector has an influence on its direct neighbours, the sortering is repeated after each correction. A mechanism is put in place to avoid falling into an infinite loop. This strategy also ensures that the good vectors around an erroneous estimate are not modified before the latter is actually processed through the filter (figure 5).

3.6 Uncertainties for individual motion vectors

To provide physically-based uncertainties on a per grid cell basis is an effort for many if not all remote sensing products, and a feature that has been wished for long for the Low Resolution Sea Ice Drift product of the OSI SAF. In the field of ice motion remote sensing, recent progress have been made for example by Sumata et al. (2014), Sumata et al. (2015b), Sumata et al. (2015a) or Hollands et al. (2015), but so far no ice motion product comes with per grid cell uncertainty for daily fields, and no algorithms is published.

Our first approach to uncertainties is pragmatic and might be refined later. It is based on statistics obtained from the validation against buoy trajectories, supplemented by a dependency on the departure of the *start* time of each drift vector (section 2.2.2) to the daily central time *12 UTC*. These uncertainties are used as input to the multi-sensor merging algorithm (chapter 4).

3.6.1 Uncertainties based on validation statistics

For practical purposes, we base our single-sensor uncertainties on the statistics obtained from validating the drift vectors against buoy trajectories. These validation statistics are different for each sensor and can be binned with the value of `status_flag` ((VALR)). In addition, it is clear from graphs in ((VALR)) that they vary with season and hemisphere.

Table 2 (resp. table 3) summarizes the values used as single-sensor uncertainties for the winter season (October to April inclusive) (resp. core of summer season: June to August inclusive). In table 3, the ”-” symbol indicates that the vectors are so un-reliable, that they are discarded from the product grid, and are thus not associated an uncertainty, nor are used in the multi-sensor merged product.

In Table 2 and table 3, the columns are for different `status_flag` that are defined in (PUM) and duplicated here for convenience:

- 30 (Nominal) : The vector was retrieved by first pass of CMCC, independently of others.
- 20 (Small Pattern) : The CMCC was applied with a smaller radius for the sub-images, due to the proximity to coast, edge or missing value.
- 21 (Corrected by neighbours): The vector was not retrieved in the first CMCC step but at the second pass, constrained using the neighbouring vectors.

During the two transition months (May and September), the values from table 2 and table 3 are linearly weighted with the day number so that there is a smooth transition from May 1st (winter values) to June 1st (summer values) and from September 1st (summer values) to October 1st (winter values).

The same methodology is used in SH, except for the definition of the winter, summer, and transition periods (see section 1.2.3).

3.6.2 Adapting uncertainties for time mis-registration

The uncertainty values provided in the previous section are valid when using the single-sensor ice drift vectors with accurate start t_0 and stop t_1 time, that are provided in the product files. These can vary between *8 UTC* and *16 UTC* across the product grid depending on the orbit and instrument characteristics (see figure 2 in (PUM) for an illustration).

Hemisphere	Product	Nominal (30)	Small patt. (20)	Corr. neighb. (21)
NH	AMSR2 36.4 GHz	2.1	5.0	9.0
	AMSR2 18.7 GHz	2.5	5.0	9.0
	SSMIS	3.5	5.0	9.0
	ASCAT	3.9	5.0	9.0
SH	AMSR2 36.4 GHz	2.7	5.0	9.0
	AMSR2 18.7 GHz	3.0	5.0	9.0
	SSMIS	5.3	5.0	9.0
	ASCAT	3.1	5.0	9.0

Table 2: Values of σ_k (unit km) used for single-sensor product uncertainty in winter, depending on satellite instrument, imaging channel, value of `status_flag` and hemisphere. The uncertainties pertain to both the $\mathbf{d}X$and $\mathbf{d}Y$components, after a 48 hours displacement (see section 1.2.4)

Hemisphere	Product	Nominal (30)	Small patt. (20)	Corr. neighb. (21)
NH	AMSR2 (GW1) 37 GHz	7.5	-	-
	AMSR2 (GW1) 18 GHz	5.0	-	-
	SSMIS (all DMSPs)	-	-	-
	ASCAT (all METOPs)	-	-	-
SH	AMSR2 (GW1) 37 GHz	9.8	-	-
	AMSR2 (GW1) 18 GHz	6.0	-	-
	SSMIS (all DMSPs)	-	-	-
	ASCAT (all METOPs)	-	-	-

Table 3: Same as table 2 but for summer season.

When neglecting $\mathtt{t}_0$ and $\mathtt{t}_1$ and rather use the vectors as if from $\mathbf{D}$@12 to $\mathbf{D}+\mathbf{2}$@12, the uncertainties must be raised. We adopt a 2nd order polynomial formula for this raised uncertainty $\sigma_k^{12}(\delta_t)$ (equation 3.7).

$$\begin{aligned} \delta_t &= |t - t_0| \\ \sigma_k^{12}(\delta_t) &= a_k \times \delta_t^2 + b_k \times \delta_t + \sigma_k \end{aligned} \tag{3.7}$$

where δ_t has units hours, and σ_k is that from the previous section. a_k and b_k have unit depend on the satellite instrument and are specified in table 4. It is easily seen from equation 3.7 that the uncertainties for vectors with $\mathtt{t}_0$ close to *12 UTC* will not be raised significantly from σ_k.

The values entering table 4 were obtained by running a validation experiment where we deliberately collocated buoys and satellite product with wrong time information. A series of such collocations are run where we mis-registered $\mathtt{t}_0$from -10 hours to +10 hours. Figure 6 illustrates this polynomial fit in the case of the AMSR-2 GW1 single-sensor estimate.

Knowledge of $\sigma_k^{12}(\delta_t)$ is also necessary for using the single-sensor uncertainties in computation of the the multi-sensor product, as the latter is designed to be from *12 UTC* to *12 UTC* two days later. The methods to generate the multi-sensor product are described in the next chapter.

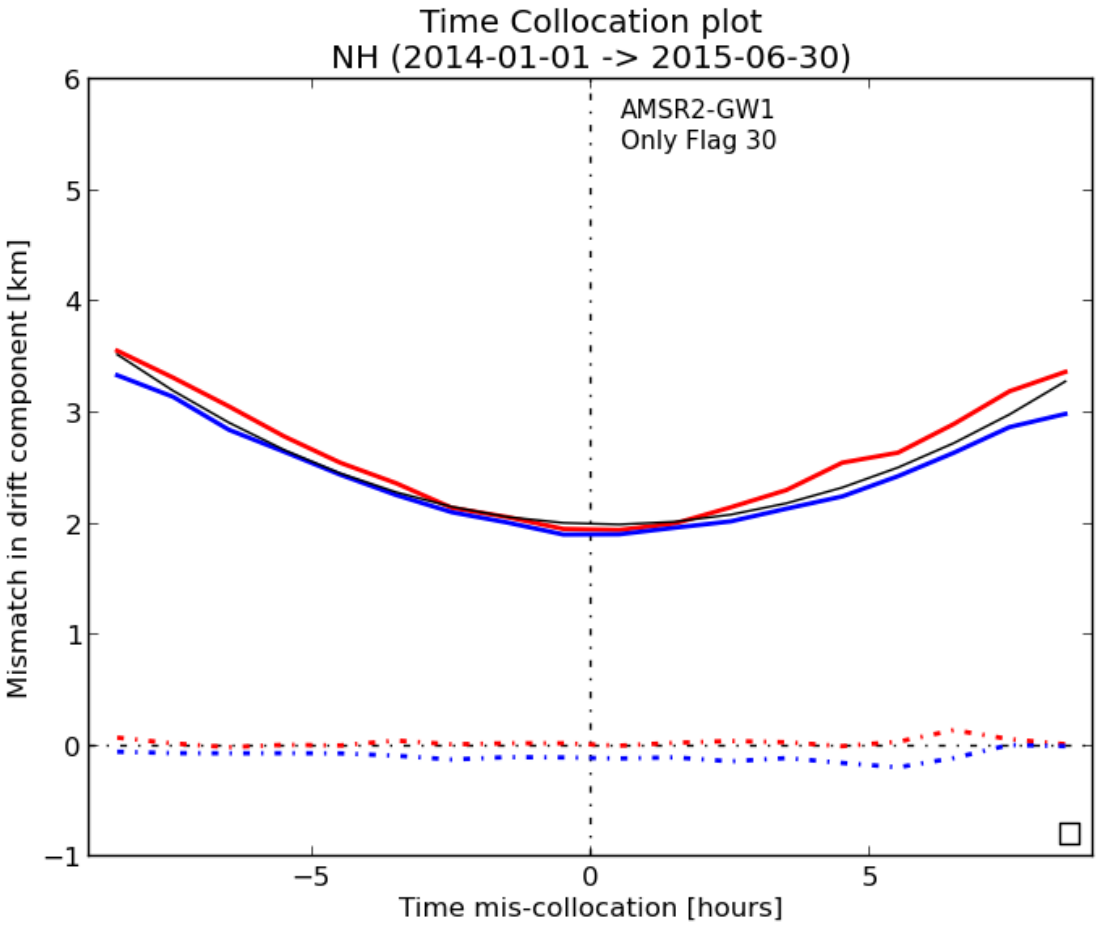


Figure 6: Variation of the mismatch (one standard deviation) of the AMSR-2 GW1 single-sensor product as a function of the temporal mis-registration during the collocation process. The solid blue ($\mathrm{d}X$) and red ($\mathrm{d}Y$) curves are from the validation data, and the black solid line is the fitted 2nd order polynomial function. Dotted blue and red lines are for the bias of the validation mismatch.

Satellite sensor	a_k	b_k
AMSR2 (GW1)	0.0195	-0.0143
SSMIS (F17)	0.0142	-0.0054
ASCAT (METOP-A)	0.0103	0.0052

Table 4: Values of a_k and b_k (both with unit km/hours) used for raising the single-sensor product uncertainty with equation 3.7 when start and stop time of the drift vectors are approximated to *12 UTC*.

4. Multi-sensor merging algorithm

The algorithms described in chapter 2 and chapter 3 are the baseline for the various single-sensor ice drift products (*e.g.* one from ASCAT instrument only, one from AMSR2 only, one from SSMIS F18 only, etc...).

The suite of OSI SAF low-resolution sea ice drift products is complemented by a multi-sensor product, that aims at presenting daily-complete maps of motion vectors. The present chapter describes the processing steps involved in producing the multi-sensor daily ice drift product.

4.1 Input single-sensor products

The input to the multi-sensor ice drift algorithms are a series of single-sensor products, obtained from the algorithms described earlier. Typically, the multi-sensor analysis ingest 3 single-sensor products, e.g. from AMSR2 (GCOM-W1), SSMIS (F18) and ASCAT (METOP-B).

4.2 Algorithm

The merging algorithm has two steps:

1. "optimal" merging at sea ice grid locations with vectors from at least one single-sensor drift vector;
2. interpolation from neighbours at sea ice grid locations with no single-sensor drift vector.

4.2.1 Optimal merging of single-sensor drift vectors

In here, we terminology *optimal* relates to the use of uncertainty estimates in terms of variance, as weights in the merging formula. Let $(u,v)_1$, $(u,v)_2$,... be the S available single-sensor drift vectors at a given grid location (typically $S = 3$). The multi-sensor drift vector $(u,v)_m$ at this same location is computed as:

$$(u,v)_m = \sum_{k=1}^{S} (\sigma_k^{12})^2 \times \sum_{k=1}^{S} \frac{1}{(\sigma_k^{12})^2} \times (u,v)_k \tag{4.1}$$

Equation 4.1 is a simplification of the full equation for combining several multi-dimensional Gaussian estimates into an *optimal* (*aka* Maximum Likelihood) estimate. The simplifications are:

1. Both u and v component use the same value of σ_k^{12};

2. We do not consider correlation between the uncertainties of u and v components;
3. We do not consider correlation between the uncertatinies of the S single-sensor products;
4. We do not consider correlation between the uncertainties of neighbouring vectors in the single-sensor products;

The simplifications above are used as pragmatic solution to decrease computation time, as fully accounting for correlations would require more advanced matrix computations. The uncertainties we use as weights in equation 4.1 are those documented in section 3.6 for the single-sensor products, once corrected for the temporal mismatch to *12 UTC*.

Very high-latitute Arctic ice drift vectors (latitude larger than 87.5N) are not used in the merging if they origin from ASCAT instrument, or if they have a `status_flag` different from `nominal` (30). This is because the single-sensor products (and particularly ASCAT ones) can be too noisy there, and this uncertainty is not reflected enough in the σ_k values.

Applying equation 4.1 at all sea ice grid locations with at least one single-sensor drift vector provides a new map of multi-sensor vectors $(u, v)_m$. This new map might still have data gaps, which are filled by spatial interpolation, as described in the next section.

4.3 Uncertainty associated to the multi-sensor product

From equation 4.1, the resulting uncertainty for the multi-sensor products are given by:

$$\sigma_u = \alpha \times \sqrt{\sum_{k=1}^{S} (\sigma_k^{12})^2} \tag{4.2}$$

where α is a scaling factor.

As noted in section 4.2.1, the transfer of uncertainty into the multi-sensor does not explicitely take into account a series of correlations that might not be negligible. The cumulated effects of these approximations lead to scaline the standard error with $\alpha = 1.5$. The value of this factor is tuned against validation statistics against buoy trajectories. Equation 4.2 is a simple but robust and cost-effective solution. Future research will allow better formulation, but -as shown in (VALR)- it already performs quite well for our purposes.

4.3.1 Spatial interpolation for gap filling

The gap filling is handled with spatial interpolation. Classically, the interpolation weight is function of the distance to neighbouring grid cells, is gaussian-shaped, with a reference length of 200 km (standard deviation). The interpolation is limited to a $[-4 : +4] \times [-4 : +4]$ neighbourhood.

All ice drift vectors which are computed as a spatial interpolation are accordingly flagged in the `status_flag` dataset (PUM).

Both the ice drift vector fields and the uncertainties are interpolated.

5. Conclusion

This Algorithm Theoretical Basis Document (ATBD) is a reference for the algorithms implemented in the OSISAF Low-Resolution Sea Ice Drift product (OSI-405). All processing steps are described: satellite image pre-processing, single-sensor motion tracking with the Continuous Maximum Cross-Correlation (CMCC), multi-sensor merging, and uncertainties.

A. Grids and projections

In this section, we introduce the formulas and equations entering the reprojection steps during both the preprocessing and the motion tracking.

A.1 Geographical Mapping

In this processing, we only deal with Polar Stereographic projections (WWW-PROJ-POLSTERE). Values for the latitude at natural origin (*aka* latitude at true scale) and longitude at natural origin are given in section 1.2.2, along with the specifications of the Geographic Datum. Those parameters define the projection plane, whose origin is North Pole.

Let (x, y) be a point in the projection plane. To each (x, y) corresponds a pair of geographic positions (λ, ϕ), λ the latitude and ϕ the longitude. (x, y) have unit *meter from origin* while (λ, ϕ) have unit degrees. Since the origin of the plane is North Pole, $(x, y) \equiv (0, 0)$ is mapped into $(\lambda, \phi) \equiv (90, 0)$. Formulas for the *forward* $((\lambda, \phi) \mapsto (x, y))$ and *inverse* $((x, y) \mapsto (\lambda, \phi))$ mappings are given in WWW-PROJ-POLSTERE. For all practical purposes, the PROJ4 library was used (WWW-PROJ4).

A.2 Gridding

If inside the grid limits, a (x, y) point is enclosed in a grid cell $[i, j]$. Grids are defined by number of lines and columns, spatial resolution and offsets (see table 1). The *gridding* process is controlled by following equations:

$$i = \text{round}\left(\frac{x * 0.001 - \mathbf{B}_x}{\mathbf{A}_x}\right) \quad \text{(A.1)}$$

$$j = \text{round}\left(-\frac{y * 0.001 - \mathbf{B}_y}{\mathbf{A}_y}\right) \quad \text{(A.2)}$$

$$(i, j) \in [0, n_x - 1] \times [0, n_y - 1]$$

The asymmetry between equation A.1 and A.2 is because point $(x, y) \equiv (\mathbf{B}_x, \mathbf{B}_y)$ defines the *upper*-right corner of the grid.

The inverse relationships to equation A.1 and A.2 are thus:

$$x = (i * \mathbf{A}_x + \mathbf{B}_x) \times 1000$$
$$y = (-j * \mathbf{A}_y + \mathbf{B}_y) \times 1000$$

A.3 Other quantities

The distance L along the Earth shape between two (λ_0, ϕ_0) and (λ_1, ϕ_1) points is given by:

$$\begin{aligned} \Delta_\lambda &= \frac{1}{2}(\lambda_1 - \lambda_0) \\ \Delta_\phi &= \frac{1}{2}(\phi_1 - \phi_0) \\ \mathbf{a} &= \sin^2(\Delta_\lambda) + \cos(\lambda_0)\cos(\lambda_1) * \sin^2(\Delta_\phi) \\ \mathbf{c} &= 2\,\mathrm{atan2}\left(\sqrt{\mathbf{a}}, \sqrt{\mathbf{1}-\mathbf{a}}\right) \\ L &= \mathsf{R}_{\mathsf{Earth}} * \mathbf{c} \end{aligned} \tag{A.3}$$

Equation A.3 is valid for a spherical Earth only, which is not exactly compatible with our choice of datum, but is a good enough approximation for all practical purposes.

Computing the distance between two (x_0, y_0) and (x_1, y_1) points only requires the prior processing through the forward mapping operator (section A.1).

B. Flow Charts

The flow charts in this appendix describe the pre-screening steps (1, 2 and 3) introduced in section 3.2.

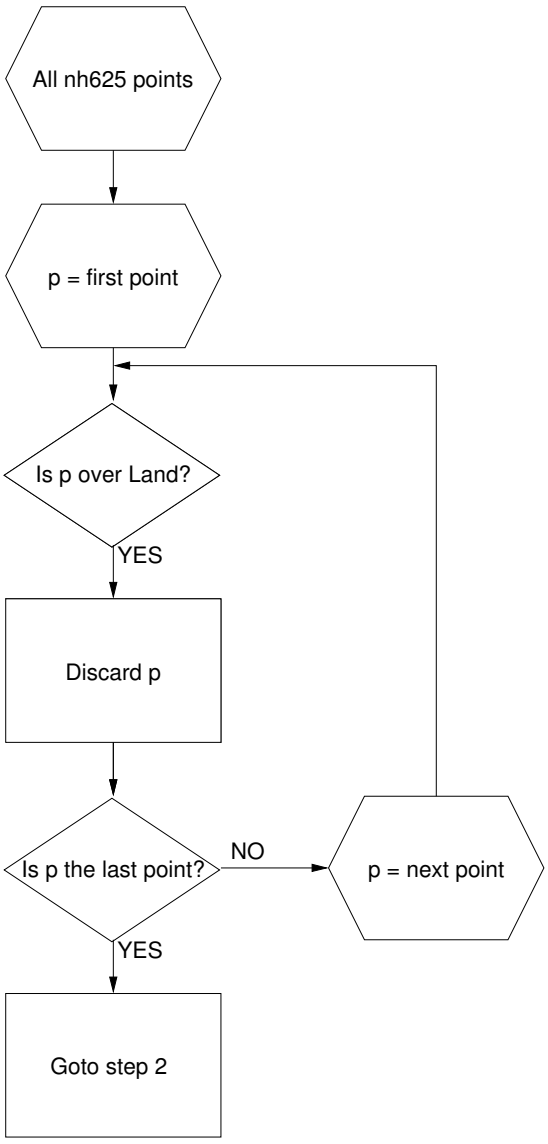


Figure B.1: Flow diagram for step 1 of the screening process which aims at discarding grid locations that are over land. Locations that pass this step are taken as input for step 2 (figure B.2)

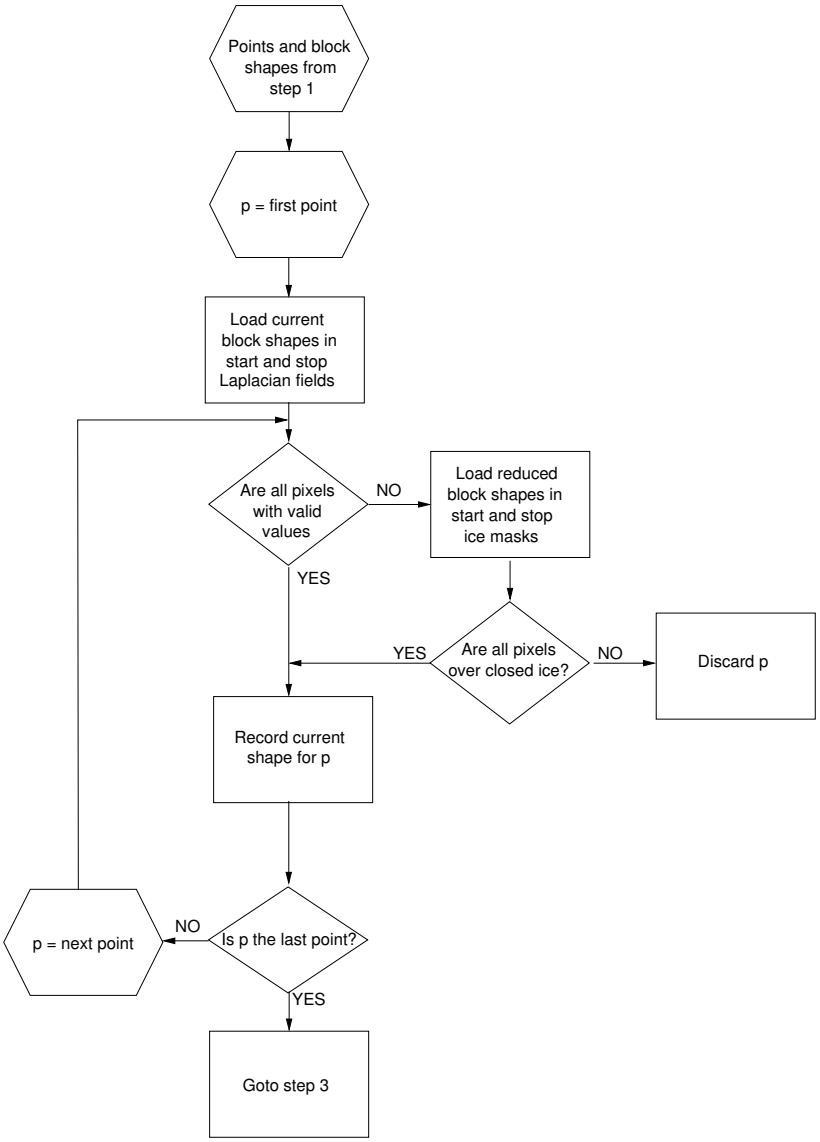


Figure B.2: Flow diagram for step 2 of the screening process which aims at discarding grid locations that are not fully over closed ice. The block shapes are used to load sub-images of the start and stop ice mask fields. Locations that pass this step are taken as input for step 3 (figure B.3)

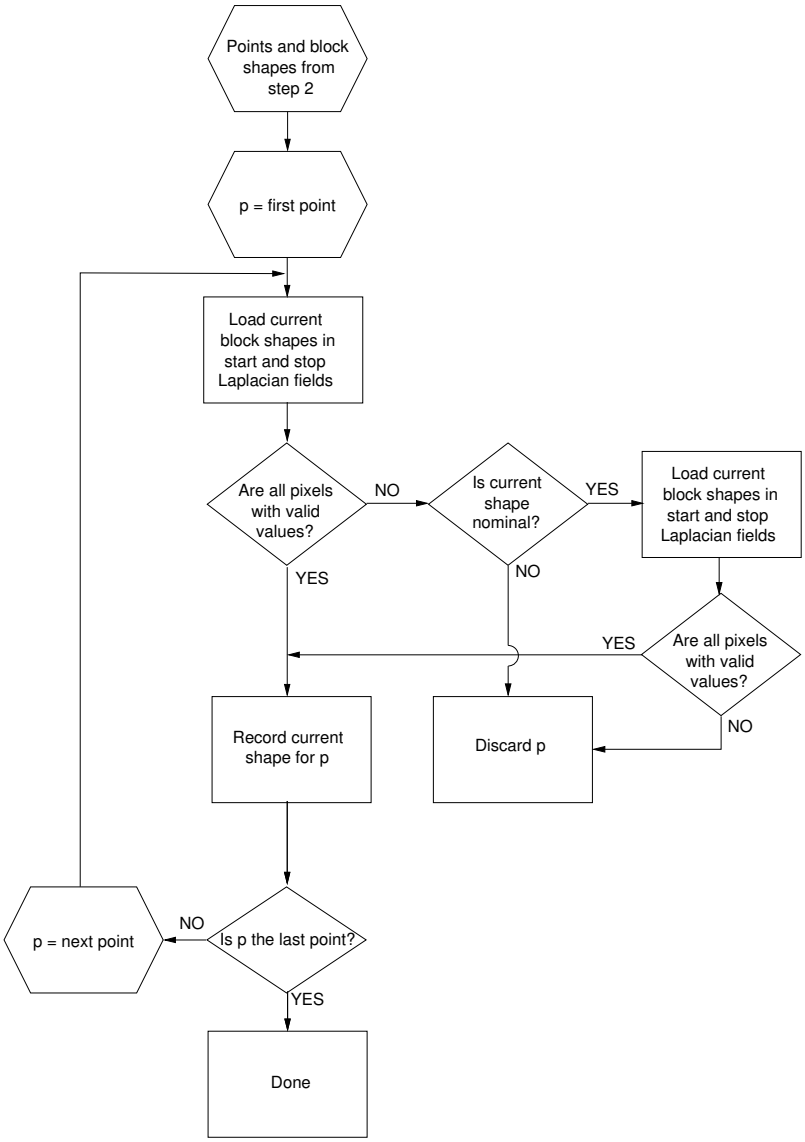


Figure B.3: Flow diagram for step 3 of the screening process which aims at discarding grid locations for which we have missing image pixels. The block shapes are used to load sub-images of the start and stop Laplacian fields. Locations that pass this step are taken as input for the motion tracking

References

S. Aaboe, L.-A. Breivik, A. Sørensen, S. Eastwood, and T. Lavergne. Product User's Manual for the OSI SAF Global Sea Ice Edge (OSI-402-c) and Type (OSI-403-c) product – v2.0. EUMETSAT OSI SAF – Ocean and Sea Ice Sattelite Application Facility, May 2016. http://osisaf.met.no/docs/.

ASCAT PG. *ASCAT Product Guide*, December 2008. URL http://oiswww.eumetsat.org/WEBOPS/eps-pg/ASCAT/ASCAT-PG-0TOC.htm.

W. J. Emery, C. W. Fowler, J. Hawkins, and R. H. Preller. Fram Strait satellite image-derived ice motions. *Journal of Geophysical Research*, 96(C3):4751–4768, March 1991.

R. Ezraty and A. Cavanié. Construction and evaluation of 12.5-km grid NSCAT backscatter maps over arctic sea ice. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3): 1685–1697, May 1999.

R. Ezraty, F. Girard-Ardhuin, and D. Croizé-Fillon. Sea ice drift in the central Arctic estimated using the 89 GHz brightness temperatures of the Advanced Microwave Scanning Radiometer – User's manual. v2.0, CERSAT, IFREMER, France, February 2007a.

R. Ezraty, F. Girard-Ardhuin, and J.-F. Piollé. Sea ice drift in the central Arctic estimated from SeaWinds/QuikSCAT backscatter maps – User's manual. v2.2, CERSAT, IFREMER, France, February 2007b.

R. Ezraty, F. Girard-Ardhuin, and J.-F. Piollé. Sea ice drift in the central Arctic combining QuikSCAT and SSM/I sea ice drift data – User's manual. v3.0, CERSAT, IFREMER, France, April 2008.

J. Haarpaintner. Arctic-wide operational sea ice drift from enhanced-resolution Quikscat/SeaWinds scatterometry and its validation. *IEEE Transactions on Geoscience and Remote Sensing*, 44(1):102–107, January 2006.

T. Hollands, S. Linow, and W. Dierking. Reliability measures for sea ice motion retrieval from synthetic aperture radar images. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 8(1):67–75, Jan 2015. ISSN 1939-1404. doi: 10.1109/JSTARS.2014.2340572.

R. Kwok, A. Schweiger, D. A. Rothrock, S. Pang, and C. Kottmeier. Sea ice motion from satellite passive microwave imagery assessed with ERS SAR and buoy motions. *Journal of Geophysical Research*, 103:8191–8214, April 1998. doi: 10.1029/97JC03334.

J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9: 112–147, 1998.

T. Lavergne. Low resolution sea ice drift Product User's Manual – v1.8. Technical Report SAF/OSI/CDOP/met.no/TEC/MA/128, EUMETSAT OSI SAF – Ocean and Sea Ice Sattelite Application Facility, May 2016a. URL http://osisaf.met.no/docs/osisaf_cdop2_ss2_pum_sea-ice-drift-lr_v1p8.pdf.

T. Lavergne. Validation and monitoring of the OSI SAF low resolution sea ice drift product – v5. Technical Report SAF/OSI/CDOP/met.no/T&V/RP/131, EUMETSAT OSI SAF – Ocean and Sea Ice Sattelite Application Facility, May 2016b. URL http://osisaf.met.no/docs/osisaf_cdop2_ss2_valrep_sea-ice-drift-lr_v5p0.pdf.

J. Maslanik, T. Agnew, M. Drinkwater, W. Emery, C. Fowler, R. Kwok, and A. Liu. Summary of ice-motion mapping using passive microwave data. Technical Report Special Publication 8, NSIDC — National Snow and Ice Data Center, November 1998. URL http://nsidc.org/pubs/special/nsidc_special_report_8.pdf.

J. A. Nelder and R. Mead. A simplex method for function minimization. *Computational Journal*, 7:308–313, 1968.

G. Notarstefano, P.-M. Poulain, and E. Mauri. Estimation of surface currents in the Adriatic sea from sequential infrared satellite images. *Journal of Atmospheric and Oceanic Technology*, 25:271–285, May 2007. doi: 10.1175/2007JTECHO527.1.

J. Schmetz, K. Holmlund, J. Hoffman, B. Strauss, B. Mason, V. Gærtner, A. Koch, and L. v. d. Berg. Operational cloud-motion winds from Meteosat infrared images. *Journal of Applied Meteorology*, 32:1206–1225, July 1993.

H. Sumata, T. Lavergne, F. Girard-Ardhuin, N. Kimura, M. A. Tschudi, F. Kauker, M. Karcher, and R. Gerdes. An intercomparison of arctic ice drift products to deduce uncertainty estimates. *Journal of Geophysical Research: Oceans*, 119(8):4887–4921, 2014. ISSN 2169-9291. doi: 10.1002/2013JC009724. URL http://dx.doi.org/10.1002/2013JC009724.

H. Sumata, R. Gerdes, F. Kauker, and M. Karcher. Empirical error functions for monthly mean arctic sea-ice drift. *Journal of Geophysical Research: Oceans*, pages n/a–n/a, 2015a. ISSN 2169-9291. doi: 10.1002/2015JC011151. URL http://dx.doi.org/10.1002/2015JC011151.

H. Sumata, R. Kwok, R. Gerdes, F. Kauker, and M. Karcher. Uncertainty of arctic summer ice drift assessed by high-resolution sar data. *Journal of Geophysical Research: Oceans*, 120(8):5285–5301, 2015b. ISSN 2169-9291. doi: 10.1002/2015JC010810. URL http://dx.doi.org/10.1002/2015JC010810.

WWW-NSIDC-PGRID. Polar stereographic projections and grids — National Sea and Ice Data Center. http://nsidc.org/data/grids/ps_grid.html.

WWW-PROJ-POLSTERE. Polar stereographic. http://www.remotesensing.org/geotiff/proj_list/polar_stereographic.html.

WWW-PROJ4. Proj.4. http://trac.osgeo.org/proj/.

WWW-WIKIP-LAPLACE. Laplace operator. http://en.wikipedia.org/wiki/Laplace_operator.