C-Band Sea Ice SAR Classification Based on Segmentwise Edge Features

Juha Karvonen
Finnish Meteorological Institute (FMI)
Finland

1. Introduction

Sea ice is a very important issue in winter navigation in the Arctic waters and also in the Baltic Sea. The winter traffic in ice is assisted by ice breakers in ice covered areas with heavy ship traffic. In other ice covered areas, however, the vessels must have the capability to navigate in sea ice without aid, meaning that they have to be designed especially for ice conditions. Earlier e.g. the Finnish ice breakers used helicopters to find the best routes in sea ice. However, today the expensive helicopter use has been replaced by utilization of remote sensing data. And for the ships navigating in ice on their own, it is even more important to have valid and useful information of the current sea ice around the ship to find its route through ice. Especially Synthetic Aperture Radar (SAR) data are useful during the dark and often cloudy northern winters. Because C-band SAR, like Radarsat (1 and 2) and Envisat ASAR, is an active instrument it also works during the dark periods and has a suitable EM wavelength to propagate through the cloud cover. Our aim is to produce automatically generated sea ice products from SAR data, such that they are informative and easy to interpret. Sea ice types in SAR images are best described by the edges present in the images. The type and amount of edges give us information on the ice types in addition to commonly used areal backscattering values and statistics. Here we present some novel features for sea ice SAR data classification, mainly derived from detected edges.

The preprocessing of the SAR data is also an important step. First the data is rectified to a map projection, Pearson (1990). We use the Mercator projection, also used in nautical charts, in the Baltic Sea and the Polar Stereographic projection for the Arctic. Also an incidence angle grid over the SAR area in appropriate projection is generated from the data included in the SAR data. After applying a land masking and incidence angle correction, Karvonen et al. (2002), using the generated incidence angle grid, we perform an edge-preserving speckle filtering. Then we apply an intensity-based segmentation, and produce a multi-resolution presentation in three resolutions by combining adjacent segments with different criteria for different resolutions. We also detect the edges and corners from the full-resolution SAR data.

After the segmentation, the detected edges can be divided into two groups: segment edges (segment boundaries) and edges within segments. The segment edge features describe the contrast and shape of the segment, and the within-segment edge features give additional geophysically relevant information on the ice properties within the segment.

The segment shape gives information on the segment, e.g. ice floes typically can have polygonal or round shape, and ice ridges and cracks or leads typically are narrow but long segments,

often they are not straight lines, however. The segment shape can be described by shape features based on the segment edges. The segment shape features in our approach are computed from the ordered edge points sampled along the edge. Then the object shape can roughly be characterized by the polygon defined by the sampled set of the edge points, and multiple useful features based on this presentation can be computed for the ice type classification.

We have collected C-band SAR data, both Radarsat and Envisat ASAR data over the Baltic Sea and also over the Arctic Sea areas for our studies. The data are Radarsat ScanSAR wide mode data and Envisat ASAR wide swath mode data. These data have a resolution of about 100m and cover an area of 400-500 km wide, and thus are suitable for operational sea ice monitoring.

2. Preprocessing

The preprocessing chain consists of the georectification to Mercator (Baltic Sea) or Polar Stereographic projection (Arctic), incidence angle correction, Karvonen et al. (2002), land masking, speckle filtering, and multi-resolution segmentation.

2.1 Speckle Filtering

We have studied two iterative algorithms performing an edge-preserving filtering. One is based on an anisotropic mean and the other on anisotropic median computation. Both the algorithms have given very good results according to our visual judgment. The algorithms are shortly described in the following two subsections. In our experiments we have used 40 iterations for both the algorithms. We have not studied the defining of a sufficient number of iterations, and 40 iterations probably is a too high number and we could use less iterations to achieve the same performance in shorter time. One way to define the number of iterations automatically would be to have a threshold for the total change between iterations, and stop after the change has reduced to a smaller value than the given threshold.

2.2 Anisotropic Mean Filtering

The anisotropic mean algorithm iteratively computes weighted means within a small window S around each pixel. We have used a 3x3 window. The weights depend on the absolute difference between the pixel value $I_t(i,j)$, $(i,j) \in S$ and the mid-pixel value $I_t(r_0,c_0)$. The number of iterations (time) is denoted by the subindex t.

$$I_t(r_0.c_0) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I_{t-1}(r_0 + i, c_0 + j) / \Delta_{i,j}, \tag{1}$$

$$\Delta_{i,j} = |I_{t-1}(r_0,c_0) - I_{t-1}(r_0 + i, c_0 + j)|, \tag{2}$$

$$if \ \Delta_{i,j} = 0, \ then \Delta_{i,j} = 1, \tag{3}$$

$$if \ \Delta_{i,j} > T, \ then \ \Delta_{i,j} = \infty. \tag{4}$$

This is iterated from the time $T = 1$ until the desired amount of iterations, in our case 40, has been reached. At $t = 0$, then input is the original SAR image.

2.3 Anisotropic Median Filtering

The anisotropic median is also computed iteratively either using only non-edge points or edge points depending on whether the mid-point is non-edge or edge, respectively. This is also computed iteratively in a 3x3 window, S. So it is computed as

$$I_t = Median(I_{t-1}(i,j)), \ (i,j) \in S \backslash E \tag{5}$$

for the non-edge mid-points and

$$I_t = Median(I_{t-1}(i,j)), \; (i,j) \in S \cap E \qquad (6)$$

for the edge points. E is the set of the edge points. This algorithm naturally requires an edge detection to be performed before running it. This is also iterated from the time $T = 1$ until the desired amount of iterations (40) has been reached. At $t = 0$, then input is again the original SAR image.
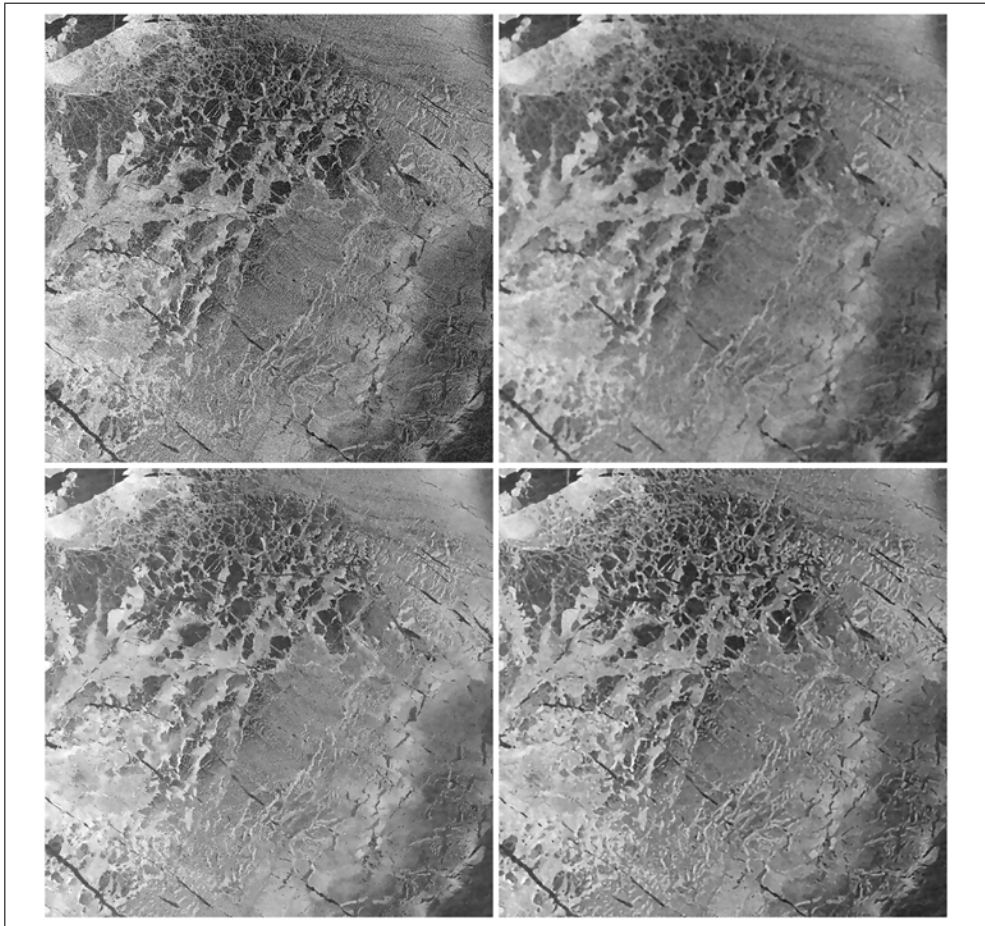

Fig. 1. An example of speckle filtering, the original Radarsat-1 image ($\approx$ 75x75 km, upper left), iterative 3x3 median (40 iterations, upper right), anisotropic mean ($T = 15$, 40 iterations, lower left), and anisotropic 3x3 median (40 iterations, lower right). The (isotropic) iterative median clearly blurs edges.

2.4 Segmentation

The segmentation algorithm we use is a K-means algorithm, Linde et al. (1980), applied to the pixel intensity values of the speckle-filtered SAR images. The values of K are typically in the range 4–8 for our SAR data. In the beginning the class means are initialized based on a cumulative data histogram computed from the image. Then the upper limits for K clusters are computed to produce K bins of equal amounts of samples and the initial class means are set to be in the middle of two adjacent limits, i.e. $m_k = 0.5(L_{k-1} + L_k)$, where L_i's are the limits between two adjacent data bins. After this initialization step, the K-means algorithm is iterated using only the image pixel values at the cluster (or segment) edges, in the sense of 8-neighborhood, from the previous iteration in the iterative computation. The iteration is repeated until no changes occur or a maximum number of iterations has been reached (to guarantee stopping).

A more sophisticated segmentation result could be achieved by adding more (texture) features. We are studying the inclusion of autocorrelation to the segmentation, but then we also need to exclude the values at the segment boundaries, because large changes at the segment edges cause high autocorrelation. Instead we should first perform an intensity-based segmentation and only after that divide the segments, if necessary, based on the texture feature.

2.5 Multi-resolution Approach

Because we here are using small scale-segments in the SAR images as features, it is necessary to have a multi-resolution presentation of the data. Then we can compute statistics of smaller-scale features over the larger scale segments, such that the results are statistically relevant.

The traditional multi-resolution approaches typically use some low pass filtering and builds a multi-resolution pyramid of the data. This naturally also reduces the accuracy of segment boundaries at the low resolutions. On the other hand, processing at the low resolutions is faster and less memory is required. However, we have here adopted a multi-resolution approach based on segmentation, segment sizes and contrasts between segments. We use three resolution layers generated by an algorithm which starts from the K-means segmentation results and then combines the adjacent segments up to a given size limit T_s (in pixels) to their neighbor segments which are larger than T_s (if they exist) if the edge contrast between the segments at the edge boundary is less than a contrast threshold T_c. The contrast threshold depends linearly on the segment area and varies between given values for the minimum and maximum segment sizes. At each iteration the smaller segments are joined to the segments larger than T_s, and after each iteration the values T_s and T_c are increased (T_s) and decreased (T_c) linearly starting from given parameter start values and ending up to given parameter end values. Finally we perform a joining of the small segments to larger segments such that all the segments smaller than a given threshold T_{tot} are joined to their neighbors. A sophisticated way of doing this is again to use an iterative method such that first the smaller segments are joined and finally the larger segments. The thresholds depend on the desired resolution level and on the image resolution. Higher size thresholds are used for the lower resolutions. A suitable value for the contrast start threshold is around 30–50 for our data, and the end value in the range 0–10.

The pseudocode of the joining algorithm looks this:

```
# Initialization of the thresholds, Tsz is segment size threshold
# and its initial value Tsz(0) is a smaller value than the final value Tsz(1).
# Tc is a inter-segment contrast threshold.
# Its initial value Tc(0) is a larger  value than its final value Tc(1).
Tsz = Tsz(0); Tc=Tc(0);
```

```
Sstep=(Tsz(1)-Tsz(0))/(Niterations);
Cstep=(Tc(0)-Tc(1))/(Niterations);

# This iteration joins the segments starting from the smaller segments
for (Niterations) do
  for (each segment)
    if ((segment_size < Tsz) AND (segment_contrast < Tc) AND
        (some_neighbor_segment_size >= Tsz)) then
      Join the segment to the closest larger segment (minimum edge contrast)
      Tsz = Tsz + Sstep; Tc = Tc - Cstep;
    endif
  end
end

# This iteration is just to guarantee that all the segments are joined
# It typically only has a very small affect (if it has).
while (no changes occur OR maximum count reached) do
  for (each segment)
    if ((segment_size < Tsz) AND (segment_contrast < Tc) AND
        (some_neighbor_segment_size >= Tsz)) then
      Join the segment to the closest larger segment (minimum edge contrast)
    endif
  end
end
```

One way to reduce resolution would also be to reduce the number of clusters (K) in the K-means clustering, i.e. to use less clusters for lower resolutions. We have made some studies of this approach also, but the work for finding optimal parametrization and integrating this with the current algorithm is still under construction.
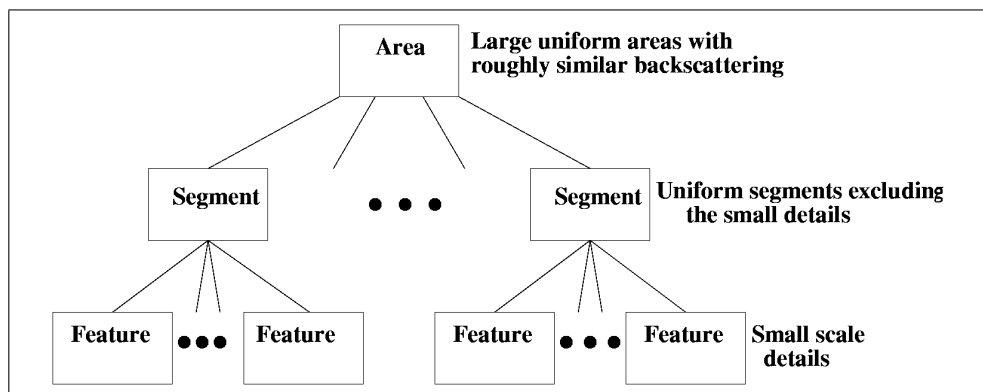

Fig. 2. The multi-resolution concept.

3. Edge Features

We have used the canny edge detection, Canny (1986), to detect edges in the SAR images. The Canny edge detector however only takes into account the local neighborhood in the thresholding. To get the connected edges better included we perform the Canny edge detection twice for one image, with two sets of thresholds, the high and low thresholds. If an edge resulting
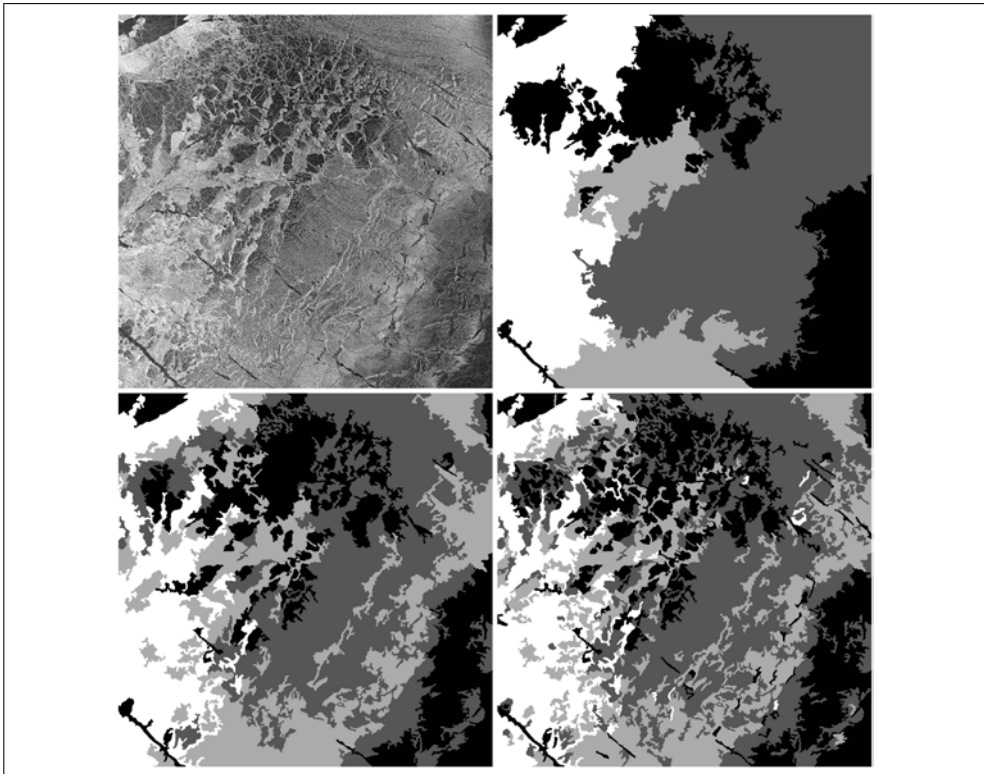
Fig. 3. A part of a Radarsat-1 SAR image (Baltic Sea, $\approx$ 75x75 km, upper left) and its segmentation in the three resolutions: low resolution (upper right), medium resolution (lower left) and high resolution (lower right).

from the Canny edge detection with the high parameter values is connected to an edge detected with the low parameters, then the edge from the detection with low parameter values is also included as an edge. We use the Canny algorithm with 5x5 pixel Gaussian smoothing and parameters $T_{lo} = 100$ and $T_{hi} = 120$ as the high Canny parameter values and $T_{lo} = 60$ and $T_{hi} = 100$ as the low Canny parameter values. The selection of these values is naturally dependent on the data scaling. These presented values seem to be a suitable selection for our SAR data. The edge detection is always computed for the SAR data before speckle filtering.

We divide the located edges into two categories, depending whether they are on a segment boundary area or inside the segment. The edge boundary area is defined as a the area of pixels which have other segments' pixels within its 8-neighborhood.

3.1 Segment Boundary Strength

The segment boundary strength can be defined in multiple ways. We can study the local gradients between the segments at the boundaries, or just simply check the amount of detected edge pixels at the segment boundary. The segment boundary strength can also give information on the segment. We utilize the segment edge contrast between adjacent segments in our
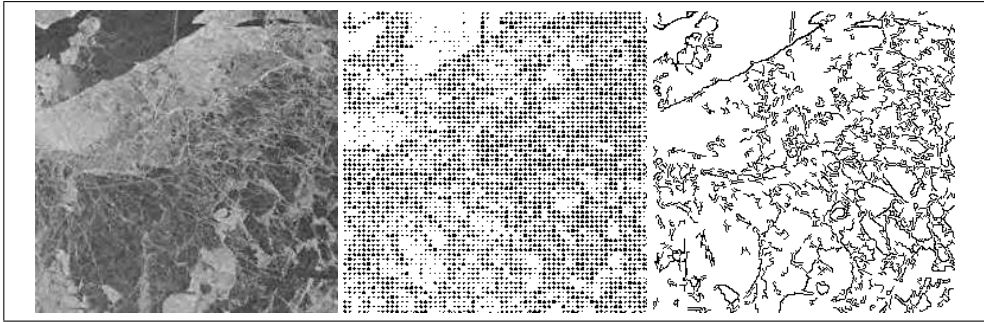
Fig. 4. A part of a SAR image ($\approx$ 25x25 km, left), detected edges (middle) and the corresponding structured edges (right), i.e. edges which are parts of larger edge segments than a given threshold, here 10.

segment joining algorithm. The segment boundary strength can also be used as a feature in segment classification, but here we mainly concentrate on the within-segment features.

3.2 Structure within Segments

The structure within segments is defined by the amount of different edge types within the segment. The edge is here said to be structured if the size of a uniform edge segment (i.e. connected edge pixels in the sense of 8-neighborhood) is larger than a given threshold T_e ($T_e > 1$), and unstructured (random edge) if the size is less or equal than T_e. If the segment size without segment boundaries is A, then we can compute three features related to the structuredness of the segment. The first is the degree of the segment random roughness or deformation

$$D_R = N_R/A, \tag{7}$$

the second is the degree of the segment structured randomness or deformation

$$D_S = N_S/A, \tag{8}$$

and the third is the relative randomness

$$D_{RS} = N_R/N_S. \tag{9}$$

N_R and N_S are the numbers of structured and random edges within the segment. The total relative number of segment edge pixels, which we here also call the segment deformation, is

$$D = (N_R + N_S)/A = N/A. \tag{10}$$

N is the total number of edge points in a segment.

3.3 Corners

We have also studied the occurrence of corner points at the segment boundaries and inside segments. To detect corners we have used a variant of the Harris (aka Harris-Stephens) edge detector, Harris & Stephens (1988). Instead of computing the Harris corner response function M_c, we have used the eigenvalues ($\lambda_1 > \lambda_2$) of the Harris matrix and thresholds T_{hi} and T_{lo} for the eigenvalues. If $\lambda_1 > T_{hi}$ at some image location (r,c), then (r,c) can be considered as an

edge point, and if additionally $\lambda_2 > T_{lo}$, then it is a corner point. The feature we use is the relative amount of corners D_c computed as:

$$D_c = \frac{N_c}{N_S + N_R} = N_c/N. \tag{11}$$

The Harris algorithm could also be used for detecting edges instead of the Canny algorithm.

3.4 Segment Shape Features Based on Segment Edges

We have also studied some shape features of the segments. The segment shape is naturally described by the segment boundary. The segment edges or boundaries are estimated as polygons. For each segment we have used a constant (20 points) with equivalent distance between the points along the segment boundary to define the polygon. This approach is basically similar to the MPEG-7 shape descriptors, Bober (2001), but our features are different and better suitable for the random shapes of ice segment features. One simple feature is the segment length, l, which in our approach is estimated as the maximum length between two edge polygon corner points along the polygon edge. The shorter distance of the two alternatives of clockwise and counter-clockwise directions is the distance between a single pair of polygon corner points. The (average) segment width, w, can then be computed as

$$w = A/l, \tag{12}$$

where A is the segment area. The segment shape ratio R_s can then be computed as

$$R_s = l/w. \tag{13}$$

This feature is a scale-independent segment shape descriptor and is high for long and narrow segments and smaller for compact segments.

We also compute the segment edge contrast, C_e, i.e. the mean difference between the inside-segment edge points and outside-segment edge points

$$C_e = \sum_{k\in in} I_k/N_{in} - \sum_{k\in out} I_k/N_{out}. \tag{14}$$

The sums are computed along the segment edge, N_{in} and N_{out} are the numbers of the edge pixels inside and outside of the segment along the segment boundary, respectively. One more feature describing the curvature of a segment is computed as a count of those pairs of two adjacent polygon line segments for which the angle between the line segments exceeds a given angle α. If the coordinates of the three polygon corners defining the two adjacent polygon edge segments are (r_{k-1}, c_{k-1}), (r_k, c_k) and (r_{k+1}, c_{k+1}), the vectors to be compared are $p_1 = (\Delta r_1, \Delta c_1)$ and $p_2 = (\Delta r_2, \Delta c_2)$. The index k is computed in modulo N_p (circular) arithmetic such that no over or underflow occur. N_p is the number of polygon corners. The vector components are

$$\Delta r_1 = \quad r_k - r_{k-1} \tag{15}$$
$$\Delta c_1 = \quad c_k - c_{k-1} \tag{16}$$
$$\Delta r_2 = \quad r_{k+1} - r_k \tag{17}$$
$$\Delta c_2 = \quad c_{k+1} - c_k, \tag{18}$$

and the corresponding vector lengths l_1 and l_2

$$l_1 = \sqrt{\Delta r_1^2 + \Delta c_1^2} \tag{19}$$

$$l_2 = \sqrt{\Delta r_2^2 + \Delta c_2^2}. \tag{20}$$

The cosine of the angle between the polygon edge segments p_1 and p_2 is

$$\cos(\alpha) = \frac{< p_1, p_2 >}{l_1 l_2}. \tag{21}$$

We have set a threshold angle, T_α, for curvature i.e. the polygon is curved at the location (r_k, c_k) if $\alpha > T_\alpha$, and the total curvature R_c for a edge polygon is defined as the relation of the number of the curved polygon corner point locations N_c to the total number of the polygon corner points N_p:

$$R_c = N_c / N_p. \tag{22}$$

We have used the value $T_\alpha = \pi/3$ in our studies.

In figure 5 we show two artificial segments and their 20-point boundary polygons, and in table 1 the features based on the boundary polygons of these two segments are computed.
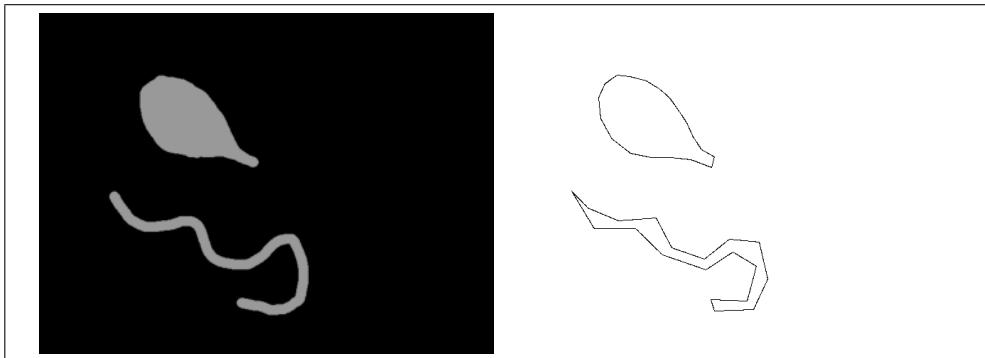
Fig. 5. An example of two artificial segments and their 20-point bounding polygons.

#	r_0	c_0	I_{in}	I_{out}	C_e	A	L	W	R_s	R_c
1	88	168	153	0	153	11783	229.27	51.39	4.46	2/20 = 0.1
2	251	104	153	0	153	7174	455.48	15.75	28.92	8/20 = 0.4

Table 1. Computed features for the artificial segments of Fig. 5.

3.5 Shape Features for the Small Segments

These features are not related to the edges, because the polygon estimation of the edge for small segments is not a very useful approach. We have used two measures of compactness instead. The first measure (C_{S1}) compares the overlapping of the actual segment and a sphere of the same size as the segment, with its center at the center of mass of the segment. The other

measure (C_{S2}) finds the bounding sphere of the segment and the feature is the segment area divided by the bounding sphere area, A_{out}.

$$C_{S1} = A_{in}/A. \tag{23}$$

$$C_{S2} = A/A_{out}. \tag{24}$$

Both the features actually give similar information and we have used only the feature C_{S1} in our classification experiments. The interpretation is straightforward: If the feature values are close to one, the segment is compact and if they are close to zero, the segments shape is not compact. Thus we have used two thresholds, $T_{c1} < T_{c2}$. If $C_{S1} < T_{c1}$, the segment is classified to a long segment and if $C_{S1} > T_{c2}$ it is classified to a compact segment.

3.6 Other studied Edge Features

We also studied the directional edge strengths using the MPEG-7 edge filters, Manjunath et al. (2001), and the local direction distributions of the edges. The orientation of the SAR edges can not be used in the same way as for typical textures, i.e. by dividing the edges to vertically oriented, horizontally oriented and so on, because the SAR orientation depends on the imaging geometry and on the location, and similar ice fields can have edge direction distributions which are rotated with respect to each other. Because of this, we can not use an edge direction histogram as a SAR feature. But we can for example utilize a feature describing how oriented the edges in a SAR image are locally, i.e. whether there exist a locally dominant direction within a image window of a fixed size. Unfortunately they did not show very good classification performance for our SAR data. Only some features, like straight ship tracks or straight ice edges could be distinguished and these could also be located by other means, e.g. locating the structured edges and edge contrasts.

We have also computed edge segment size distributions withing segments and at the segment boundaries, but we have not studied their properties carefully yet. The division into structured and random edges, i.e. a two-valued distribution, is our current approach.
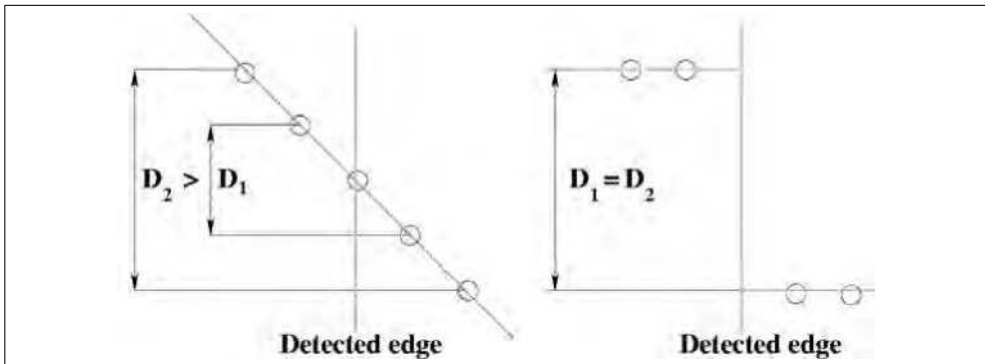

Fig. 6. A ramp edge and a sharp edge, the edge normal is horizontal in the image and the pixel value is in the vertical direction. For a sharp edge the intensity difference for both distances is about equal, and for an ideal ramp edge the intensity difference increases linearly as the distance increases.

We have also studied the division of segment and within-segment edges into sharp edges and ramp edges (smooth edges). The edge is considered as a sharp edge if at the edge $D_1 \approx D_2$,

$D_1 = I_1 - I_{-1}$, $D_2 = I_2 - I_{-2}$, i.e. the pixel values in the speckle filtered image at two distances, $l_1 < l_2$, along the edge normal on opposite sides of the edge are almost equal, and as a ramp edge if $aD_1 < D_2$, $a > 1.0$ is a given factor, see Fig. 6. The distribution of edge type to these two categories was also studied within the segments. The relation of the amounts of these two edge types can also be used to classify the segments, but the geophysical interpretation is still missing. At least it can be used to distinguish between smooth ice segments (like open water and fast ice) and deformed ice segments, as many other edge features, but its ability to provide complementary information is still vague. Intuitively it could be useful in distinguishing e.g. areas with (widely spaced) clear ridges from areas of rubble fields.

4. Some Classification Results

4.1 Open Water Detection

We have earlier used the segment-wise autocorrelation as an open water detector, see Karvonen et al. (2005). Our recent studies have shown that also edge information can be utilized in open water detection.

The relative amount of edges within segment D can be used to locate most of the open water area, but even better indicator for open water is the relative amount of structured edges D_S. In some cases open water can be mixed with level ice or fast ice areas. The classification can be further improved in some cases by using the relative amount of corners D_c as an additional feature. In general we can say that segment-wise D_S is a good open water detector, such that open water has very low values of D_S. Performed tests show that it works well for both the Baltic sea ice and for the Arctic Sea ice. We have two examples of this shown in Figs. 7 and 8. The ASAR mosaic of Fig. 8 has been composed by overlaying all the available ASAR data over the Kara sea area starting from November 2008. Multiple daily images were typically acquired, and this mosaic image describes the ice situation on January 23rd 2009.

4.2 Ice Classification Based on the Inside-Segment Edges

We have made studies with several different sets of edge features. The ratio of the total number of edges within segment and the segment area (D) represents the degree of deformation of the segment. However, this only feature can not always e.g. very well distinguish between open water and deformed ice areas. But including the relative amount of structured edges (D_S) and the relative number corners (D_C), the ice types can be rather well distinguished, see Fig. 9. This figure is a three channel image of the three features suitably scaled for visual inspection. In this figure over the the Gulf of Bothnia, Baltic Sea, the open water areas appear as brown areas and fast ice areas have more red color, indicating that these areas have relatively more corner points than the open water areas. The other ice areas mostly have different tones of green, the more deformed areas being brighter. This example shows the potential of using these three features together for sea ice SAR classification.

4.3 Ice Classification Based on the Segment Shape Features

More information from the data can be extracted by the segment shape classification. Here we only show one example of segment shape classification for one SAR window. The segments smaller than a given size threshold ($T_A = 3000$) have been located and classified to compact segments and non-compact ("long") segments and indicated with different colors in Fig. 10. The relative amounts, with respect to the segment area, of different types of these smaller segments withing medium-scale or large-scale segments (or areas) can then be computed, and we can then get information on the relative amount of cracks, ridges and other ice structures
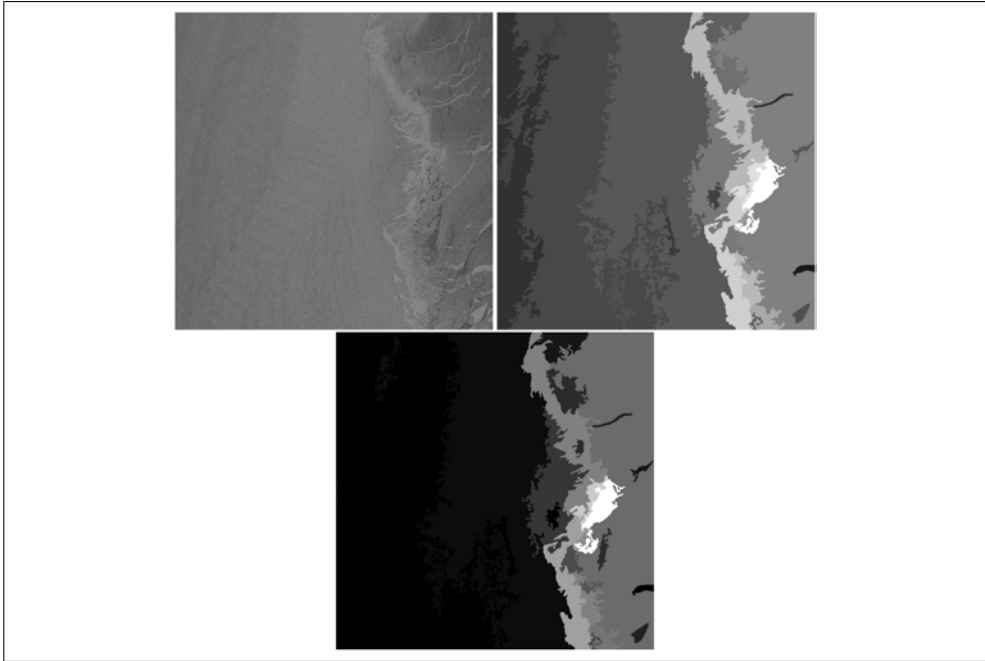
Fig. 7. A Radarsat-1 window over the ice edge, open water area is on the left side of the image (upper left), its D (upper right) and D_S images (lower middle). The open water areas appear as dark areas, especially in the D_S image, and the brash ice area at the ice edge appears bright in both edge images, indicating that it has relative much edge points.

(smooth or rough/ridged compact segments) within the larger areas. We have used an experimental set of parameters for the different segment classes as follows: for compact segments $R_s < 7$ and $R_c < 0.3$, for "long" segments $R_s > 11$ and $R_c < 0.4$. The edge contrast threshold applied was 5 for the dark segments and 15 for the bright segments, i.e. the contrast must exceed these values to be classified. These parameters are also experimental, and studying of ways to find better parameters is under construction.

Some examples of this classification are also given in Figs. 11 and 12. They show the relative amount of different features with different gray tones, the brighter values indicating higher occurrence of the specific feature type.

The relation of amount the edge types (sharp and ramp edges) can also be used as a feature, it is high in the areas of prominent features, e.g. ice floes, ridges with large enough spacing (depending on the SAR resolution) or cracks. This ratio can be used as an additional feature for refining the segment-wise classification. Here we show one example of this feature in Fig. 13 for the ASAR mosaic shown in Fig. 8
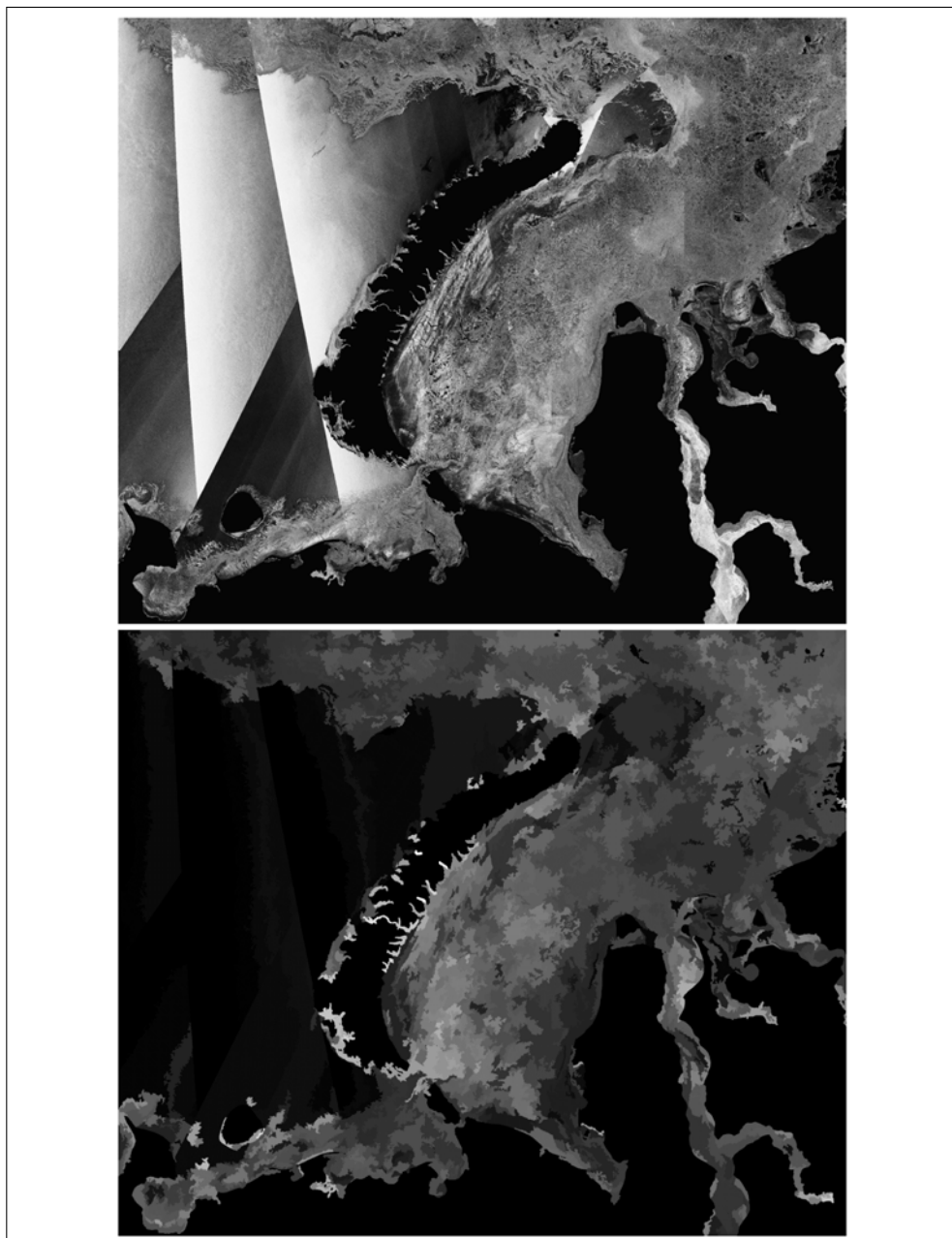
Fig. 8. A SAR image mosaic over the Kara Sea (Jan 23rd 2009, upper image) and the values of D_S over the area (lower), The areas of open water, mainly on the left side of the image have very low value of D_S.
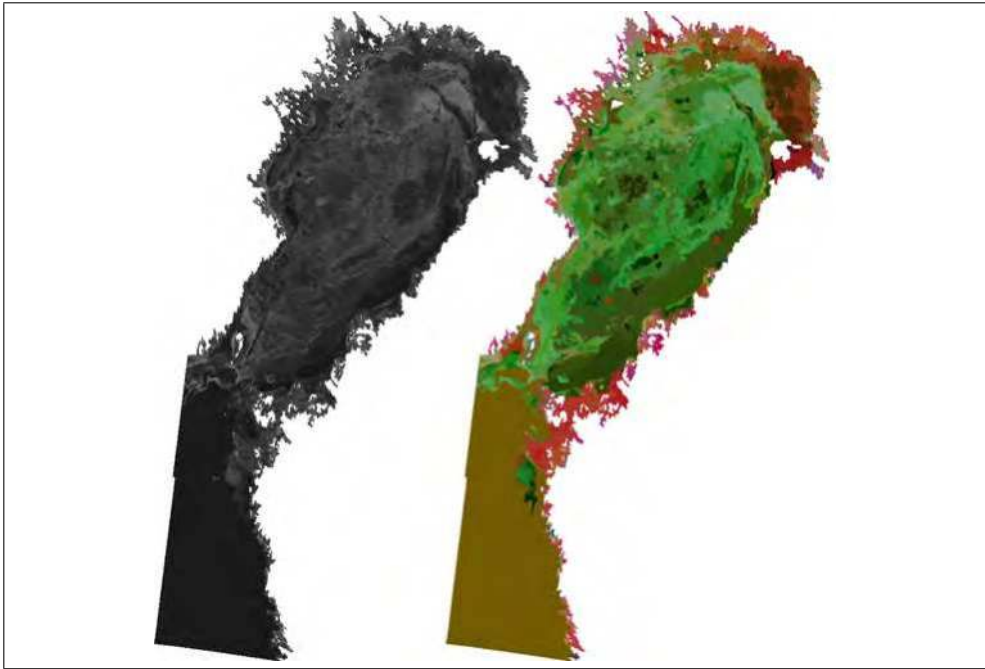
Fig. 9. A Baltic Sea Radarsat-2 image (left) and a 3-feature classification result (RGB three channel presentation) in medium resolution (right), the used features are the relative number of corners (red), relative amount of edges (green), and relative amount of structured edges (blue), the total area covered by the SAR image is about 500x300 km.
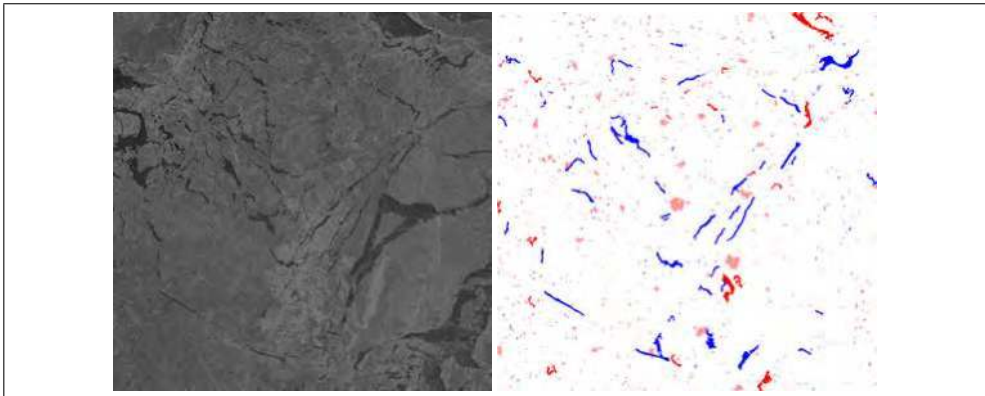
Fig. 10. A part of a Radarsat-1 SAR image (Baltic Sea, left), and the the classified features (for segments smaller than a threshold, i.e $A < T_A$, $T_A = 3000$ pixels in this example, right). The red segments have the edge contrast $C > T_{ctr2}$ and the blue segments $C < T_{ctr1}$, the segments drawn with lighter red and blue are classified based on the small segment algorithm. The total area covered by the image is about 75x75 km.
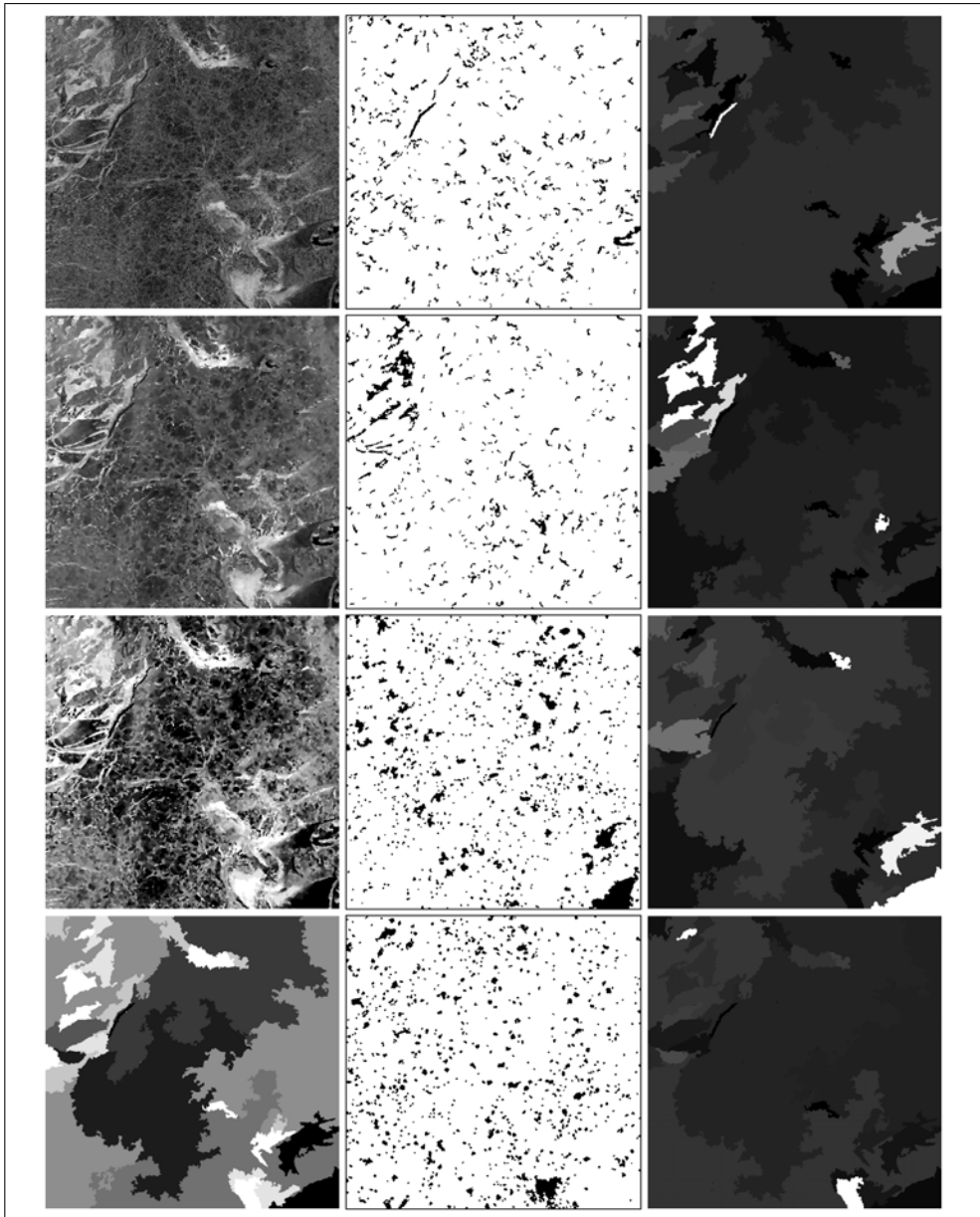
Fig. 11. Envisat ASAR image and detected class-wise features and their relative amounts in different image areas. In the first column from top towards bottom: the original SAR data, speckle-filtered (anisotropic median) data, segmentation. In the second column, the detected features from top towards bottom: dark long features, bright long features, dark compact features and bright compact features. In the third column the segment-wise (large-scale) amounts of different features corresponding to the second row features.
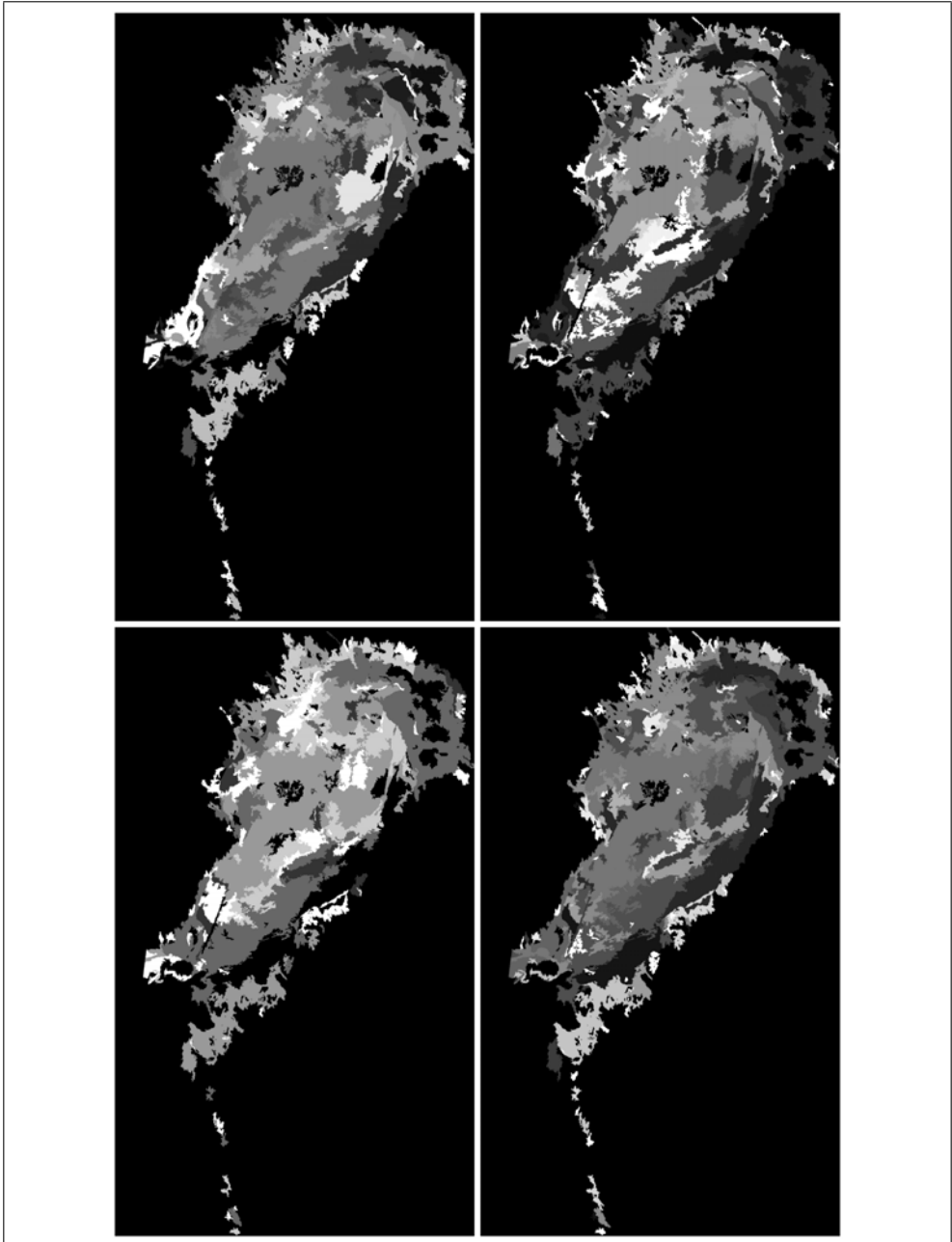
Fig. 12. Segment-wise (large scale) relative amounts of different feature types for the Radarsat-2 image (see Fig. 9): dark compact segments (upper left), bright narrow segments (upper right), dark narrow segments (lower left), bright compact segments (lower right).
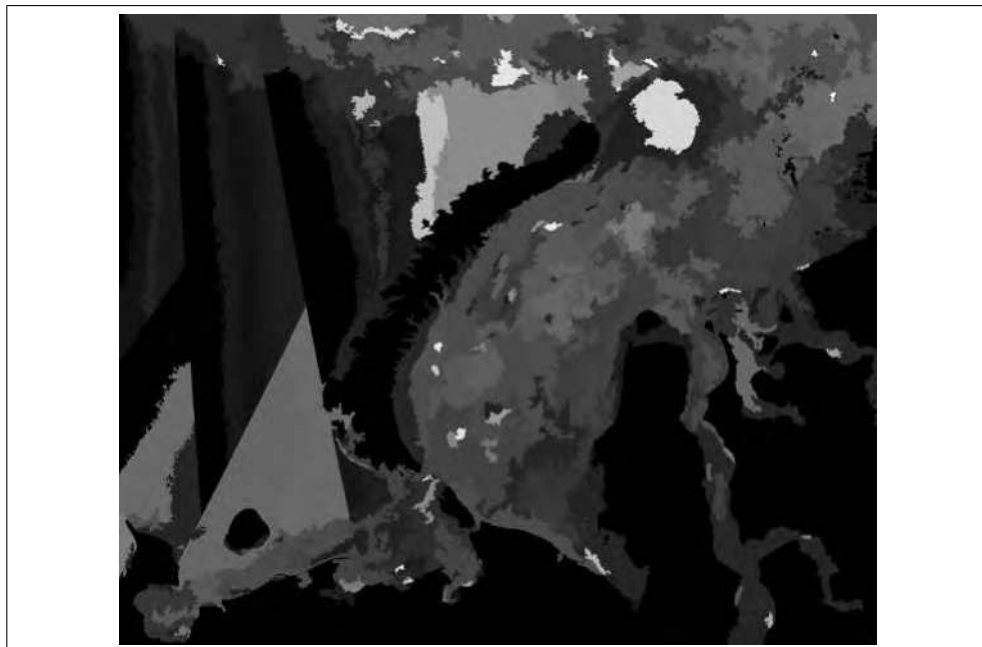
Fig. 13. The segment-wise ratio of structured to random edges for the ASAR mosaic of Fig. 8. The ice areas with many cracks, ice floes or other clearly distinguishing features have higher values. hight values and the other deformed fields, like rubble fields, have lower values. In the open water covered areas (on the left side of the image) the values can have large variations because there are only few edges in these areas, and a small change in the amount of edges of either type can cause large changes in the ratio.

5. Conclusion and Future Work

We have developed a whole sea ice SAR image processing and interpretation chain and demonstrated its usability. The basic idea is that most of the SAR information, in addition to the backscattering lies in the SAR edges. We have also found out that suitable combinations of our edge features can be used for sea ice SAR classification and they give us useful complementary information of the sea ice structure. We believe that we have not yet discovered the full potential of all the edge-related features and here only present some suitable features for SAR classification.

The speckle filtering using either anisotropic mean or median works well and the execution times are reasonable for operational SAR processing. We have not studied the optimal number of iterations, and probably still use too many iterations.

The multi-resolution approach also seems to work well, and gives reasonable segmentations and ice areas compared to visual interpretations. The selection of parameters naturally affects the results of the lower resolution images produced by the segment joining algorithm. The best results are achieved by using many iterations i.e. by increasing the joined segment size slowly, but this also increases the execution times.

The classification results have been promising. Many sea ice classes can be distinguished with very simple edge features, like the combination of amount of edges, and the relative amount of structured edges and the relative amount of corners. The methods can distinguish open water areas very well, and also different ice types and the areas with certain types of ice features (e.g. cracks or ridges) can be located. Not all the features are found, but when using large enough areas, the relative amounts of different features can be estimated.

The parametrization of the studied algorithms has been experimental and we must concentrate on better optimization of the parametrization. We are going to study an automated parameter extraction for given training data sets to reduce the work of experimental parameter definition. But even our experimental parameters have shown promising results and edge features are a very promising addition to SAR classification algorithms. These features will probably also be very useful for classification of other kinds of SAR data sets over land areas. We have studied these features only with a few images from three instruments Radarsat-1, Radarsat-2 and Envisat ASAR. In the next phase we are going to make tests for larger data sets, for example for a whole winter season in both Baltic Sea and Kara Sea, and also for other SAR instruments with different operating parameters (e.g. X- and L-band SAR).

The classification results have been evaluated against visual interpretation. Sea ice measurements are very difficult and expensive to carry out. Because the ice is typically moving, except in fast ice zones, multiple measurements should be made simultaneously (or temporally as close as possible) with the satellite passing time. Even making a few measurements is difficult and expensive, because typically a ship capable of operating in sea ice is required to get in the target area. And the ice properties can differ much in a relatively small area, less than a SAR pixel size. However, visual interpretation of the ice typing from SAR data by our sea ice experts has been very good compared to our occasional field campaign measurements and feedback from the Finnish ice breakers using this information, and we can consider it as good reference data.

6. References

Bober, M. (2001). Mpeg-7 visual shape descriptors, *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on MPEG-7* **11**(6): 716–719.

Canny, J. (1986). A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence* **8**(6): 679–698.

Harris, C. & Stephens, M. (1988). A combined corner and edge detector, *Proc. of Alvey Vision Conference*, Univ. of Manchester, pp. 147–151.

Karvonen, J., Simila, M. & Makynen, M. (2002). An iterative incidence angle normalization algorithm for sea ice sar images, *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS'02)*, Vol. III, pp. 1524–1527.

Karvonen, J., Simila, M. & Makynen, M. (2005). Open water detection from baltic sea ice radarsat-1 sar imagery, *IEEE Geoscience and Remote Sensing Letters* **2**(3): 275–279.

Linde, Y., A. Buzo, A. & Gray, R. M. (1980). An algorithm for vector quantizer design, *IEEE Trans. Communication* **28**(1): 84–95.

Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V. & Yamada, A. (2001). Color and texture descriptors, *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on MPEG-7* **11**(6): 703–715.

Pearson, F. (1990). *Map Projections: Theory and Applications*, CRC Press.

Geoscience and Remote Sensing New Achievements

Edited by Pasquale Imperatore and Daniele Riccio

Our planet is nowadays continuously monitored by powerful remote sensors operating in wide portions of the electromagnetic spectrum. Our capability of acquiring detailed information on the environment has been revolutionized by revealing its inner structure, morphology and dynamical changes. The way we now observe and study the evolution of the Earth's status has even radically influenced our perception and conception of the world we live in. The aim of this book is to bring together contributions from experts to present new research results and prospects of the future developments in the area of geosciences and remote sensing; emerging research directions are discussed. The volume consists of twenty-six chapters, encompassing both theoretical aspects and application-oriented studies. An unfolding perspective on various current trends in this extremely rich area is offered. The book chapters can be categorized along different perspectives, among others, use of active or passive sensors, employed technologies and configurations, considered scenario on the Earth, scientific research area involved in the studies.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds