Image Processing for Identification of Sea-Ice Floes and the Floe Size Distributions

Qin Zhang and Roger Skjetne, *Member, IEEE*

Abstract—An unmanned aerial vehicle was used as a mobile sensor platform to collect sea-ice features at Ny-Ålesund in early May 2011, and several image processing algorithms have been applied to samples of sea-ice images to extract useful information about sea ice. The sea-ice statistics given by the floe size distribution, being an important parameter for climate and wave- and structure-ice analysis, is challenging to calculate due to difficulties in ice floe identification, particularly the separation of seemingly connected ice floes. In this paper, the gradient vector flow (GVF) snake algorithm is applied to solve this problem. To evolve the GVF snake algorithm automatically, an initialization based on the distance transform is proposed to detect individual ice floes, and the morphological cleaning is afterward applied to smoothen the shape of each identified ice floe. Based on the identification result, the image is separated into four different layers: ice floes, brash pieces, slush, and water. This makes it further possible to present a color map of the ice floes and brash pieces based on sizes, and the corresponding ice floe size distribution histogram. The proposed algorithm yields an acceptable identification result, and its effectiveness is demonstrated in a case study. A discussion on the methods and results concludes the paper.

Index Terms—Floe size distribution, ice floe identification, image processing, marginal ice zone, remote sensing.

I. INTRODUCTION

SEA ICE, which is defined as any form of ice that forms as a result of seawater freezing [1], covers approximately 7% of the total area of the world's oceans [2]. It is turbulent because of wind, wave, and temperature fluctuations. Various types of sea ice can be found in ice-covered regions. Ice floe, which is the flat pieces of sea ice, can range from meters to kilometers in size. The floe size distribution is a basic parameter of sea ice that affects the behavior of sea-ice extent, both dynamically and thermodynamically. Particularly for relatively small ice floes, it is critical to the estimation of melting rate [3]. Hence, estimating floe size distributions contributes to the understanding of the behavior of the sea-ice extent on a global scale. In addition to this, the floe size distribution is also important

in ice management for Arctic offshore operations [4], [5], for example:

- Quantify the efficiency of ice management for Arctic offshore operations and automatically detect hazardous conditions, for example, by identifying large floes that escape the icebreakers operating upstream of a protected structure. The size and shape of managed floes can be identified by the image processing system, compared with limit values, and further processed by the risk management system.
- Estimate the ice loads on stationary Arctic offshore structures by empirical formulas [6], [7].
- Initialize high-fidelity numerical models [8]–[13] and validate these at various moments in time by matching the simulated ice fields with the actual ones.
- Provide an early warning of an ice compaction event, which can be dangerous if the ice-structure interaction mode changes from a "slurry flow" type to a "pressured ice" type [7], [14].

Despite its importance, however, the useful information retrieved from observational data sets are still limited because of difficulty in analysis and lack of efficient post-processing tools for such data sets. Therefore, the development of temporally and spatially continuous field observations of sea-ice conditions and determining corresponding ice floe size distributions are necessary.

One of the best ways to observe ice conditions in the oceans is by using aerial imagery and applying digital image processing techniques to the observations. This method can reduce or suppress ambiguities, incompleteness, uncertainties, and errors regarding an object and its environment, yielding more accurate and reliable information [15]. Cameras are typically used as sensors on mobile sensor platforms in ice-covered regions to characterize ice conditions [16], [17]. Cameras can collect precise spatially continuous measurements, which are particularly suitable for providing detailed localized information of sea ice. However, an important prerequisite is a clear sky and sight during missions.

A remote sensing mission to determine ice conditions was performed by the Northern Research Institute (NORUT) at 78°55′ N 11°56′ E, Hamnerabben, Ny-Ålesund, from May 6 to 8, 2011. An unmanned aerial vehicle (UAV) was used as a mobile sensor platform because of its flexibility in coverage and in spatial and temporal resolution, which are three important sensor-platform attributes. The use of cameras as sensors on a UAV was explored to measure ice statistics and properties. The

Fig. 1.   CryoWing UAV operation at Ny-Ålesund (photographer: Qin Zhang).

TABLE I
CryoWing Technical Specification

Weight	30 kg max take off weight.
Wingspan	3.8 m.
Cruise speed	100 - 120 km/h.
Range/endurance	500 km / 5 h.
Max altitude	2500 m dynamic range, 5000 m absolute.
Payload capacity	Max 15 kg including fuel load.
Engine	Two stroke gasoline.
Navigation	GPS.
Ground equipment	PC with modem, RC control.
Flight	Autonomous, but under ground control.
Communication	GSM or Iridium satellite modem.

TABLE II
Visible Spectrum Camera Specifications

Camera modal	Canon EOS 450D.
Lens type	Canon EF 28mm f/2.8.
Aperture value	11.00.
Sensor	22.214.8mm CMOS.
ISO	200.
Dimensions	42902856.
Resolution	960 dpi.
Exposure time	1/250 sec.
Sampling frequency	0.66Hz.

objective of the mission was to gather information about the ice conditions in the Arctic. The further goal was to develop tools based on the processed ice data that can be applied for decision support in Arctic offshore operations.

A CryoWing [18] UAV, as shown in Fig. 1, was used for the mission. This UAV was designed for cryospheric measurements and environmental monitoring, and its technical specification is found in Table I. The basic instrumentation of the CryoWing is an onboard computer that controls the different payload instruments, stores data to a solid-state disk, and relays data to the ground. The onboard payload system has a GPS receiver and a three-axis orientation sensor that is independent of the avionics system. The sensor device used in this analysis is a digital visual camera with specifications found in Table II. The UAV flew in the inner part of Kongsfjorden to collect high-resolution images of sea ice. Several image-processing algorithms have then been applied to these images to extract useful information of the sea ice, such as ice concentration, ice floe boundaries, and ice types [17].

Automatic identification of individual floe edges is a key tool for extracting information of floe size distribution from aerial images. In an actual ice-covered environment, ice floes typically touch each other, and the junctions may be difficult to identify in digital images. This issue challenges the boundary detection of individual ice floes and significantly affects ice floe size analysis. Several researchers have tried to mitigate this issue. In [19] and [20], the authors separated closely distributed ice floes by setting a threshold higher than the ice-water segmentation threshold and separated the connected ice floes manually when the threshold did not work well. In [17] and [21], the authors applied and compared derivative and morphology boundary detection algorithms in both model ice and sea-ice images. However, nonclosed boundaries are often produced by traditional derivative boundary detection, while some boundary information is often lost by morphology boundary detection. To separate connected sea-ice floes into individual floes, the watershed transform (widely used in connected object segmentation) was adopted in [22] and [23]. Due to an ineluctable oversegmentation problem of the watershed-based method, the authors in [22] manually removed these oversegmented lines, while those in [23] automatically removed the oversegmented lines whose endpoints were both convex. However, over- and undersegmentation still affected the ice floe detection results. In [24] and [25], the authors introduced a mathematical morphology together with principal curve clustering to identify ice floes and their boundaries in an almost fully automated manner. Their method operated on the binary images and focused on the morphological characteristics of ice floes rather than on the real boundaries. It was limited by crowded ice floe images, in which the ice floes in the mass were connected to each other, and no "hole" or concave regions could be found after binarization. Those methods are not applicable in our research because of these limitations.

To separate seemingly connected floes into individual ones, a gradient vector flow (GVF) snake algorithm [26] is applied in this research. However, to start the algorithm, a proper initial contour is required for the GVF snake to evolve correctly. Therefore, a manual initialization is typically needed, particularly in crowded floe segmentation. To solve this problem, an automatic contour initialization is proposed to avoid manual interaction and reduce the time required to run the algorithm. Once individual ice floes have been identified, the floe boundaries are obtained, and the floe size distribution can be calculated from the resulting data.

II. Ice Image Processing Methods

A. Ice Pixel Extraction

Due to the fact that sea ice is whiter than open water, the pixel values differ under normal conditions. See Fig. 2, for example, ice pixels have higher intensity values than those belonging to water in a uniform illumination ice image. Therefore, ice pixels can be extracted by using the thresholding method [27]. Most of the ice can then be identified, as shown in Fig. 3 based on Fig. 2. Of the ice pixels identified, however, only "light ice" have larger pixel intensity values than the threshold. "Dark ice," with pixel intensity values between the threshold and water, such as ice pieces under the water surface, may not be identified and thus considered to be water, according to the thresholding
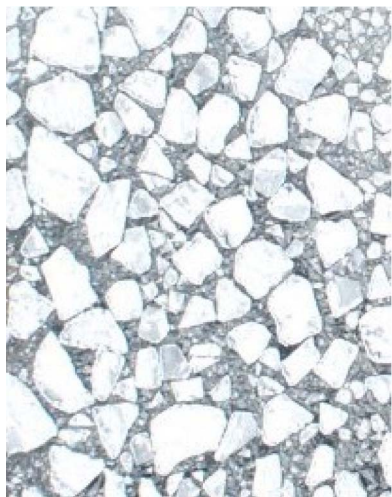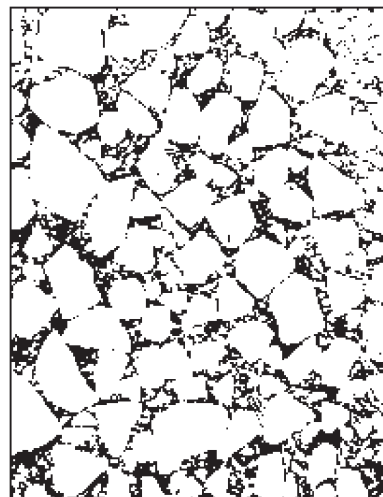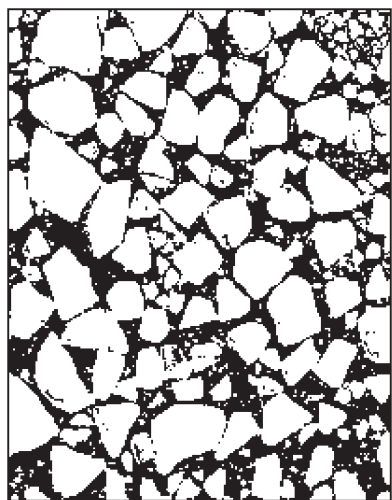
Fig. 2.   Original sea-ice image.

Fig. 4.   Ice extraction using the k-means method.

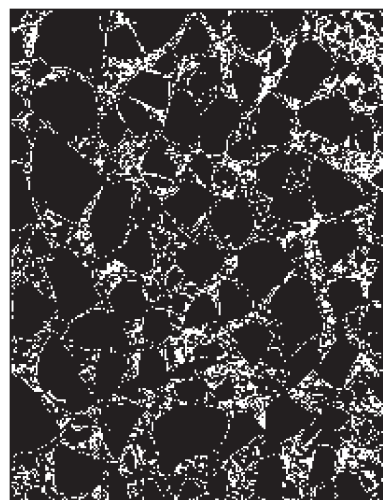Fig. 3.   "Light ice" extracted by the thresholding method.

Fig. 5.   "Dark ice" found by subtracting Fig. 3 from Fig. 4.

method. Both "light ice" and "dark ice" pixels are required for an accurate analysis. To distinguish "dark ice" from open water, the k-means clustering algorithm [28] can be applied. This minimizes the within-cluster summed distance to partition a set of data into k clusters.

The image is then divided into three or more clusters, using the k-means algorithm. The cluster with the lowest average intensity value is considered to be water, while the other clusters are considered ice, as shown in Fig. 4. The "dark ice" is then obtained by comparing the difference between Figs. 3 and 4, as shown in Fig. 5.

B. Ice Edge Detection

The most challenging task is to identify individual ice floes in the sea-ice image, in particular separating the floes that are very close or connected to each other. The boundaries between apparently connected floes have a similar brightness to the floes themselves. The boundaries are too weak to be detected directly, which significantly affects the ice floe statistical result. Therefore, the GVF snake algorithm is proposed to solve this problem.

The GVF snake operates on the grayscale image in which the real boundary information, particularly "weak" boundaries, is better preserved. The GVF snake algorithm is an extension of the snake or active contours algorithm [29]. In the classic snake algorithm, a given initial curve can move under the influence of internal forces from the curve itself and external forces computed from the image data. The algorithm stops when the internal and external forces reach equilibrium. The internal and external forces are defined such that the snake will conform to an object boundary or other desired features within an image. The classic snake algorithm can solve a number of image segmentation problems effectively, particularly in "weak" boundary detections. The capture range of the external force fields, however, is limited with a difficulty to progress into boundary concavities. It is, therefore, sensitive to the initial contour, which should be somewhat close to the true boundary. Based on these limitations, the authors in [26] introduced the dense vector field (GVF), which was derived from images by minimizing a certain energy functional in a variational framework, to expand the capture range of external force fields from boundary regions to homogeneous regions. The GVF snake is faster and less restricted by the initial contour.
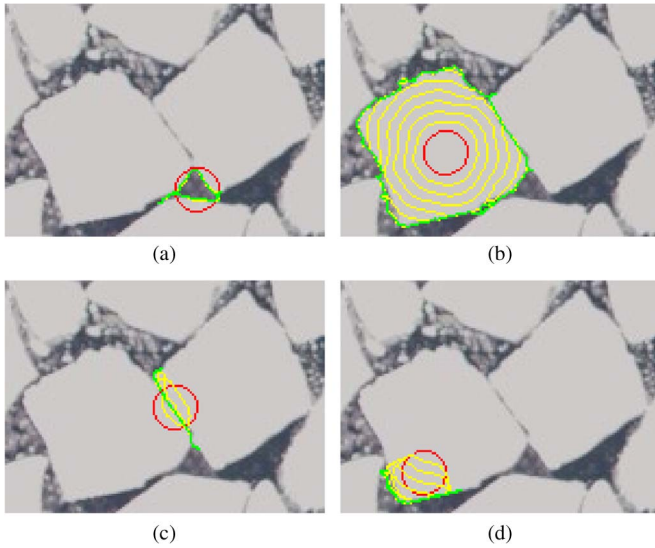
Fig. 6. Initial contours located at different positions and their corresponding curve evolutions. The red curves are the initial contours, the yellow curves are iterative runs of the GVF snake algorithm, and the green curves are the final detected boundaries. (a) Initial contour 1 located at the water, and the water region boundary is found. (b) Initial contour 2 located at the center of an ice floe, and the whole floe boundary is found. (c) Initial contour 3 located at a weak connection, and the weak connection is found. (d) Initial contour 4 located near the floe boundary inside the floe, and only a part of floe boundary is found.

The GVF snake algorithm [26] is able to detect the weak connections between floes and ensure that the detected boundary is closed. As an example, shown in Fig. 6(b), given an initial contour (red curve), the snake finds the floe boundary (green curve) after a few iterations (yellow curves). The GVF snake algorithm relaxes the requirements of the initial contour. However, a proper initial contour for an object is still necessary, particularly to identify the mass of ice floes in an ice image. Many initial contours are required when performing the GVF snake algorithm to identify all individual ice floes, and these should have proper locations and shapes. An automatic contour installation algorithm is therefore devised to increase the efficiency of the ice floe segmentation method based on the GVF snake algorithm.

Fig. 6 illustrates the floe boundary detection results affected by initializing the contour at different locations. In Fig. 6(a), the initial contour is located at the water, close to the ice boundaries. The snake rapidly detects the boundaries, however, not the ice but the boundaries of the water region. When initializing the contour at the center of an ice floe, as shown in Fig. 6(b), the snake accurately finds the boundary after a few iterations. A weak connection will also be detected if the initial contour is located on it, as shown in Fig. 6(c). However, when the initial contour is located near the floe boundary inside the floe, as shown in Fig. 6(d), the snake may only find a part of the floe boundary near the initial contour. It should be noted that the curve is always closed, regardless of how it deforms, even in the cases of Fig. 6(c) and (d), which appear to be nonclosed curves. This behavior occurs because the area bounded by the closed curve tends toward zero.

Fig. 6 illustrates that, with proper parameters, the snake will find a boundary, regardless of where the initial contour is
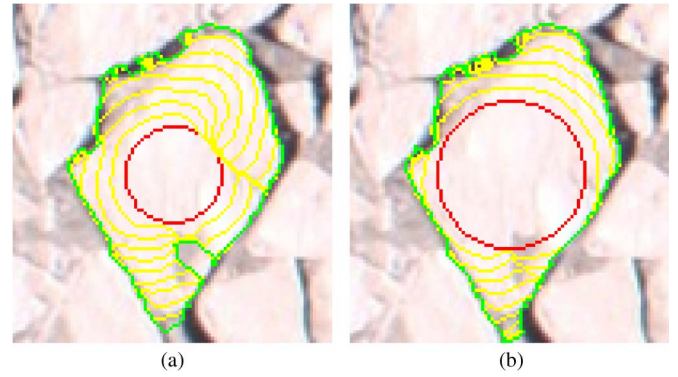

Fig. 7. Initial contours with different sizes and their curve evolutions. (a) Small contour initialized at the floe center, giving erroneous evolutions of the snake. (b) Large contour initialized at the floe center, giving convergence of the snake to the correct boundary.

located. This fact is beneficial for connected floe segmentation. As shown in Fig. 6, the initial contours should be located inside the floes and centered as close to the ice floe center as possible.

In the GVF snake algorithm, the initial contour does not need to be as close to the true boundary as for in the traditional snake algorithm. However, it will need more iterations to find the boundary if the initial contour is further distanced from the floe boundary, and it may also converge to an incorrect result [26] (particularly when the grayscale of floe is uneven). Take Fig. 7 as an example. This contains speckle inside of the ice floe, where the pixels of the speckle are darker than the other pixels of the floe. This phenomenon will affect the boundary detection, particularly when the initial contour is not close to the actual boundary. As shown in Fig. 7(a), when the initial contour is too small, the algorithm does not find the complete boundary after many steps, being blocked by the speckle. If we enlarge the initial contour, as shown in Fig. 7(b), the initial contour allows for a faster determination of the entire floe boundary. Therefore, the initial contour should still be set as close to the actual floe boundary as possible.

To accomplish the requirements of the initial contour, an automatic contour initialization algorithm based on the distance transform [30] and its local maxima is proposed. Given a binary image, whose elements only have values of "0" or "1," the pixels with a value of "0" indicate the background, whereas the pixel with a value "1" indicates the object. The distance transform of a binary image is the minimum distance from every pixel in an object to the background.

For example, Fig. 8(a) shows a small binary image matrix for a simple shape, and the matrix in Fig. 8(b) shows the corresponding distance transform (using "City Block" distance metrics). The local maximum is the pixel whose value is greater or equal to any of its neighbors, as shown by the green numerals in Fig. 8(b). A local maximum of the distance transform ideally corresponds to the center of an object, but more than one local maximum are detected in many cases. Thus, a dilation operator [31] is used to merge the local maxima within a short distance (within as a threshold T) of each other. The centers of the dilated regions [red "+" in Figs. 8(b) and 9(b)], which are called "seeds," are chosen as the locations of our initial contours.
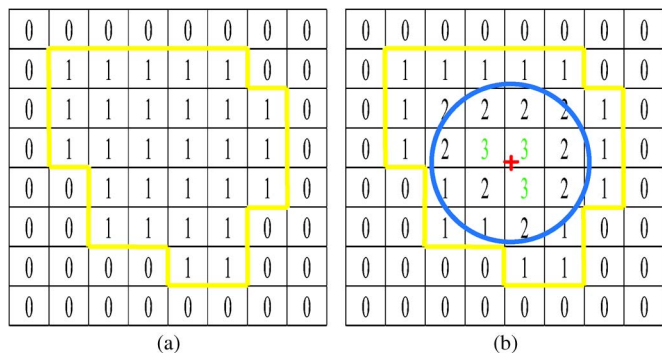
Fig. 8. Contour initialization algorithm. (a) Binary image matrix. (b) Distance transform of (a), local maximum, seed, and initial contour.
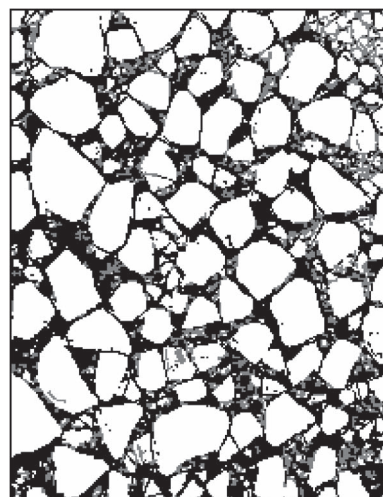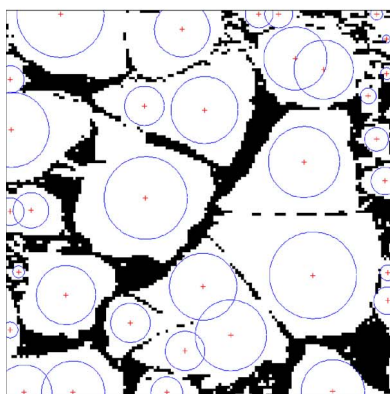

Fig. 9. Connected ice floe segmentation based on GVF snake. (a) Sea-ice floe image. (b) Binary image with initial contours. (c) Segmentation result.
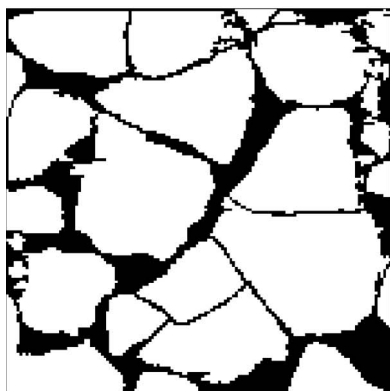

Fig. 10. Sea-ice segmentation image. The white ice is the segmentation result from the "light ice" in Fig. 3, and the gray ice is the segmentation result from the "dark ice" in Fig. 5.

Moreover, to efficiently approach the floe boundary, as discussed earlier, the initial contours are adapted to the floe size. The circular shape is chosen as the shape of the initial contour, since this shape deforms to the floe boundary more uniformly than other shapes, being unaware of the floe's irregular shape and orientation. The radius of the circle is then chosen according to the pixel value at the seed in the distance map. This ensures that the initial circle is inside the floe, as shown as blue circles in Figs. 8(b) and 9(b) (Note that, since the "City Block" distance is used to decrease the number of local maxima, the pixel values at the seeds must be divided by $\sqrt{2}$ to obtain the circle radii).

After initializing the contours, the GVF snake algorithm is run on each contour to identify the floe boundary. Superimposing all the boundaries over the binarized ice image results in separation of the connected ice floes. The final segmentation result of Fig. 9(b) is shown in Fig. 9(c). Note that the boundary pixels are specifically labeled as "residue ice" for special handling in subsequent use.

It should be noted that the local maximum whose distance is larger than the threshold T will not be merged into one seed. This means that some floes may have more than one seed. However, two or more seeds for one ice floe will not affect its boundary detection but may increase the computational time.

When multiple ice floes are crowded together, it is difficult to initialize the contours for each ice floe. To obtain a more accurate result and reduce the computational time, the contours were initialized in both "light ice" image (see Fig. 3) and "dark ice" image (see Fig. 5). The GVF snake algorithm is then run to derive "light ice" segmentation (the white ice pieces in Fig. 10) and "dark ice" segmentation (the gray ice pieces in Fig. 10) individually. Collecting all the ice pieces in both "light ice" and "dark ice" segmentation images resulted in the segmented image, as shown in Fig. 10. It should be noted that the "light ice" and "dark ice" should be labeled differently in the segmentation result. Otherwise, some "light ice" and "dark ice" may be unable to be separated.
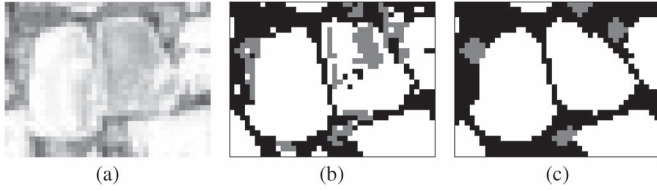
Fig. 11. Ice shape enhancement. The white pixels are "light ice" pixels, and the gray pixels are "dark ice" pixels. (a) Ice floe image with speckle. (b) Segmentation result of (a). (c) Shape enhancement result of (b).

The pseudocode of the proposed ice edge detection algorithm is concluded in Algorithm 1.

Algorithm 1 Ice Floe Boundary Detection

Input: Sea-ice image.
Start algorithm:
1: $GVF \leftarrow$ GVF derived from grayscale of input image.
2: $ICE \leftarrow$ binary ice image by the K-means method.
3: $LIGHT \leftarrow$ binary "light" ice image by the thresholding method.
4: $DARK \leftarrow ICE - LIGHT$.
5: $S_L \leftarrow$ Seeds of $LIGHT$ found by local maxima of distance transform.
6: $S_D \leftarrow$ Seeds of $DARK$ found by local maxima of distance transform.
7: **for** each seed $s_l \in S_L$ **do**
8:　　$r_l \leftarrow$ local maxima values at s_l.
9:　　$ic_l \leftarrow$ initial contours locate at s_l with its radius r_l.
10:　$B_L \leftarrow$ boundary detected by performing the snake algorithm based on GVF and ic_l.
11:　$LIGHT \leftarrow LIGHT$ with B_L superimposed.
12: **end for**
13: **for** each seed $s_d \in S_D$ **do**
14:　$r_d \leftarrow$ local maxima values at s_d.
15:　$ic_d \leftarrow$ initial contours locate at s_d with its radius r_d.
16:　$B_D \leftarrow$ boundary detected by performing the snake-algorithm based on GVF and ic_d.
17:　$DARK \leftarrow DARK$ with B_D superimposed.
18: **end for**
19: $SEGMENTATION \leftarrow LIGHT + DARK$ ($LIGHT$ and $DARK$ are marked differently in $PIECES$).
20: **return** $SEGMENTATION$.
Output: Sea-ice segmentation.

C. Ice Shape Enhancement

Because of the noise, some floes may contain holes or smaller ice pieces inside. For example, Fig. 11(a), which is extracted from Fig. 2, shows an ice floe with speckle. Because of the uneven grayscale of the ice floe, the lighter part of the floe is considered as "light ice" [the white pixels in Fig. 11(b) and (c)], whereas the darker part is considered as "dark ice" [the gray pixels in Fig. 11(b) and (c)] by the k-means and threshold method. This means that the ice floe cannot be completely identified, and the shape of the detected ice floe is rough, as shown in Fig. 11(b).
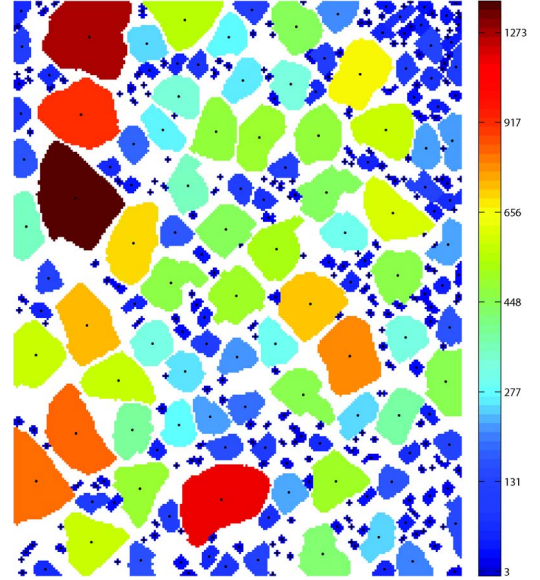

Fig. 12. Labeled and colorized ice pieces.

To smoothen the shape of the ice floe, morphological cleaning [32] is used after ice floe identification. Morphological cleaning is a combination of first morphological closing and then morphological opening [31] on an image. Morphological closing tends to smooth the contours of objects, generally joins narrow breaks, fills long thin gulfs, and fills holes smaller than the structuring element, while morphological opening removes complete regions of an object that cannot contain the structuring element, smooths object contours, breaks thin connections, and removes thin protrusions.

Using this method, all the "light ice" and "dark ice" pieces in the sea-ice segmentation image are first arranged from small to large. Then, morphological cleaning with disk-shaped structuring element [31] is performed to the arranged ice pieces in sequence, and the radius of disk-shaped structuring element is adapted to the size of each ice piece automatically. This process will ensure the completeness of the ice floe, and smaller ice floes or brash pieces contained in a lager ice floe are removed. The shape enhancement result is shown in Fig. 11(c).

After shape enhancement, collecting and labeling all the cleaned ice pieces results in Fig. 12. To present a better visualization of the sizes (defined as the pixel number of each ice piece), the ice pieces are labeled in different colors based on the formula

$$\mathbf{Color}(p) = \begin{cases} 0, & \text{if } p \notin ICE \\ \left(1 - \exp\left(-\frac{size(i)}{1000}\right)\right) \cdot 10000, & \text{if } p \in ice(i) \end{cases} \tag{1}$$

where p is the pixel, $ICE = \{ice(1), ice(2), ice(3), \ldots\}$ is a set of identified ice pieces, $ice(i) \in ICE$, and $size(i)$ is the pixel number of $ice(i)$. Smaller ice pieces are blue, and larger ice pieces are red. Ice positions, found by averaging the positions of the pixels of each ice piece, are denoted using black dots.

The pseudocode of the proposed ice shape enhancement algorithm is concluded in Algorithm 2.
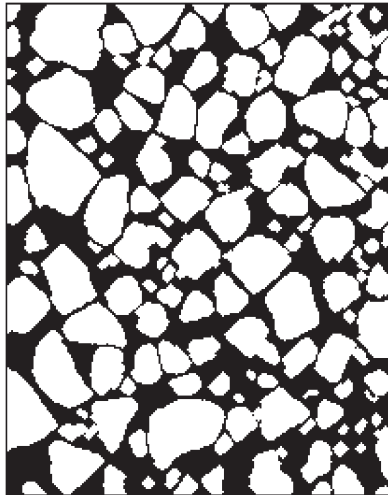
Fig. 13. Layer showing the "ice floes" in Fig. 2.

Fig. 14. Layer showing the "brash ice" in Fig. 2.

Algorithm 2 Ice Shape Enhancement

Input: Sea-ice segmentation from Algorithm 1.
Start algorithm:
  1: $PIECES \leftarrow$ labeled regions in $SEGMENTATION$ arranged from small to large.
  2: $BW \leftarrow$ empty black image.
  3: **for** each labeled region $piece \in PIECES$ **do**
  4:     $piece \leftarrow$ morphological clean and fill hole.
  5:     $BW \leftarrow BW$ with $piece$ superimposed and labeled.
  6: **end for**
  7: $IDENTIFICATION \leftarrow$ labeled regions in BW.
  8: **return** $IDENTIFICATION$.
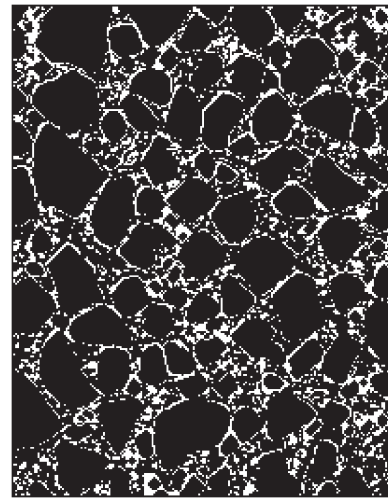**Output**: Individual ice piece identification.

Fig. 15. Layer showing the "slush" in Fig. 2.

D. Ice Type Classification and Floe Size Distribution

According to [33], brash ice is considered as floating ice fragments no more than 2 m across. To distinguish brash ice from ice floes in our algorithm, we define a brash-ice threshold parameter (pixel number or area) that can be tuned for each application. The ice pieces with size larger than the threshold are considered to be ice floes, while smaller pieces are considered to be brash ice. The remaining ice pixels in Fig. 4 were are labeled as slush. The result is four layers of a sea-ice image (using Fig. 2 as an example): ice floe (see Fig. 13), brash ice (see Fig. 14), slush (see Fig. 15), and water (see Fig. 16). Based on the four layers, a total of 154 ice floes and 189 brash ice pieces are identified in Fig. 2. The coverage percentages are 60.52% ice floe, 3.34% brash ice, 16.03% slush, and 20.11% water.

The ice floe (brash) size can be determined by the number of pixels in the identified floe (brash). If the focal length f and camera height are available, the actual size in SI unit of the ice floes and floe size distribution can be also calculated [34] by converting the image pixel size to its SI unit size. The ice floe size (calculated by counting the pixel number of the floe) distribution histogram is shown in Fig. 17.

The residue ice, which is the detected boundary pixels between the connected floes, was previously considered as slush
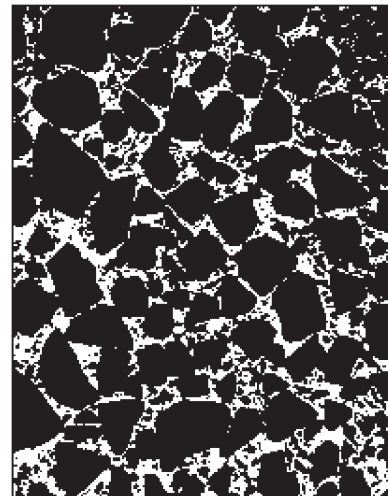

Fig. 16. Layer showing the "water" in Fig. 2.

(since there often is a boundary layer of slush ice between two ice floes) and included in Fig. 15. However, the residue ice, as shown in Fig. 18, can be also handled specifically according to the applied subsequent processing by the user.
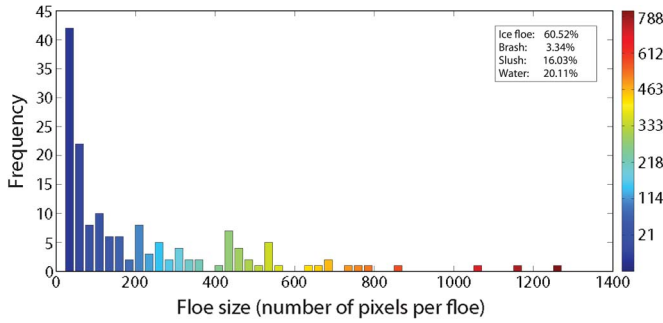
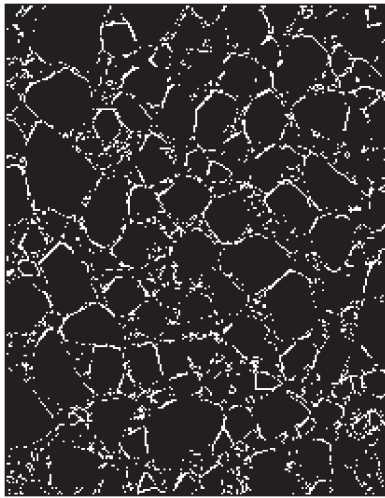Fig. 17.   Floe size distribution histogram of Fig. 13.

Fig. 18.   Residue ice (boundary pixels).

III. Case Study and Discussion

The remotely sensed images of sea ice were acquired in the NORUT mission to Ny-Ålesund. The aerial sea-ice images, collected by the UAV, usually cover a large area, and the illumination of the images is often nonuniform. In addition to this, the perspective distortion may also exist in the image data because of the shooting angle of the camera. Perspective distortion usually exists when an aerial vehicle orbits the observation field. Both of these two issues will affect the final ice floe identification and size distribution results. This section illustrates how to apply the proposed algorithm to process the sea-ice image, shown in Fig. 19, covering a large area with perspective distortion.

A. Local Processing

The image is first divided into smaller overlapping regions, such that each region can be analyzed locally [32]. The ice edge detection algorithm (Algorithm 1) is performed on each region to obtain a subsegmentation image. Then, we remove the overlapping part and superimpose the subsegmentation images into their locations (by stitching the sub segmentation images), resulting in an overall segmentation image. This procedure is shown in Fig. 20.
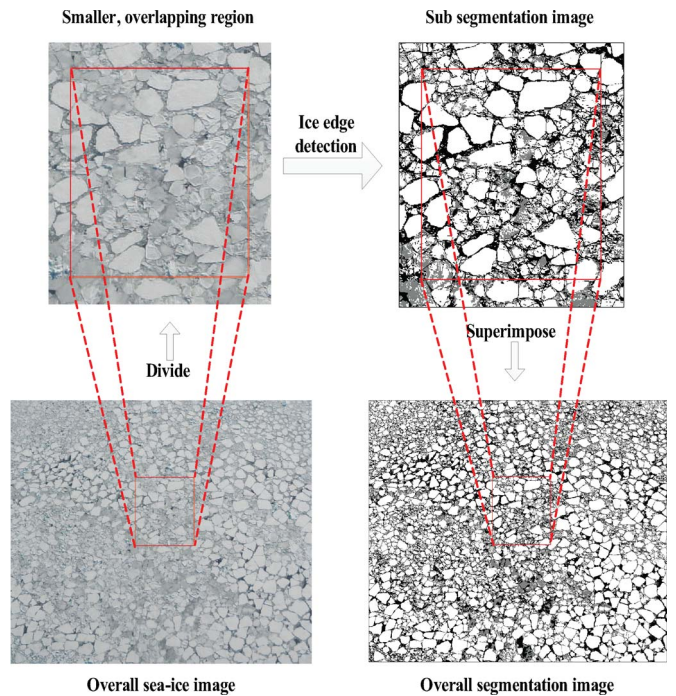

Fig. 19.   Sea-ice image.

Fig. 20.   Local segmentation procedure. The white pixels are "light" ice pixels, and the gray pixels are "dark" ice pixels.

Processing the local subimages of the overall sea-ice image is recommended to obtain an accurate segmentation result (but at the expense of more processing time and manual intervention). Some ice information can be lost when globally extracting "light ice" and "dark ice" from a sea-ice image, when nonuniform illumination or shadow problems exist in the sea-ice image. Moreover, a sea-ice image typically contains multiple ice floes that crowd together, as shown in Fig. 19, where parts of floe boundaries become weaker than others. When using the same GVF parameter, which controls the capture range of the GVF, the external forces near weak connections are weaker than those near strong boundaries. If the GVF capture range is too strong, the capture range of the strong boundaries will dominate the entire external force field, while the external force
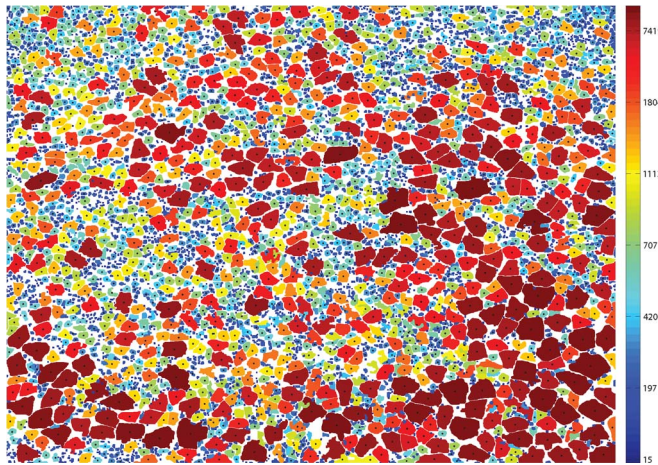
Fig. 21.   Ice floe and brash size distribution without orthorectification.

Fig. 22.   Ice floe and brash size distribution after orthorectification.

Fig. 23.   Ice floe size distribution histogram of Fig. 22.

near the weak connections will be too weak to pull the snake toward the desired boundaries. Usually, weak connections tend to be more difficult to detect when increasing the GVF capture range, which results in under-segmentation. If the capture range is decreased, however, the noise is enhanced and leads to oversegmentation. Therefore, the GVF capture range under the same GVF parameter cannot represent an overall sea-ice image. To identify all of the boundaries, it should be adjusted according to each subimage.

B. Geometric Calibration

When the perspective distortion exists in the image data, the final identification result, as illustrated in Fig. 21, is not adequate for the calculation of size distribution statistics. The ice floes in the far range of the image will seem smaller than those in the near range. This distortion will therefore induce errors in further analyses.

According to [34], [35], and the Appendix, the image can be orthorectified when the values of the shooting angle and the field of view of the camera are known, thus needing a sensor to measure the camera's shooting angle. However, the actual parameters of the camera were not measured in this mission. Hence, to give an example to illustrate the overall algorithm, we have estimated the shooting angle to be approximately 20° and the field of view to be 46° (using to the statistical similarities between the size distributions of near and far range of the image). Using this, we can orthorectify the overall segmentation image.

For a nonridged and nonshielded image, the geometric calibration should be performed on the segmented sea-ice image (after Algorithm 1) before ice shape enhancement (Algorithm 2). Otherwise, the small ice floes located at the far end of the image could be still considered to be brash ice. Furthermore, to reduce the visual distortion caused by the fractional zoom calculation, the image will be enlarged, and the total number of pixels will increase after orthorectification. The points between the pixels in the orthorectification coordinates that are mapped from the image coordinates must be interpolated. Each pixel holds quantized values that represent the color or gray level of the image at a particular point. Image
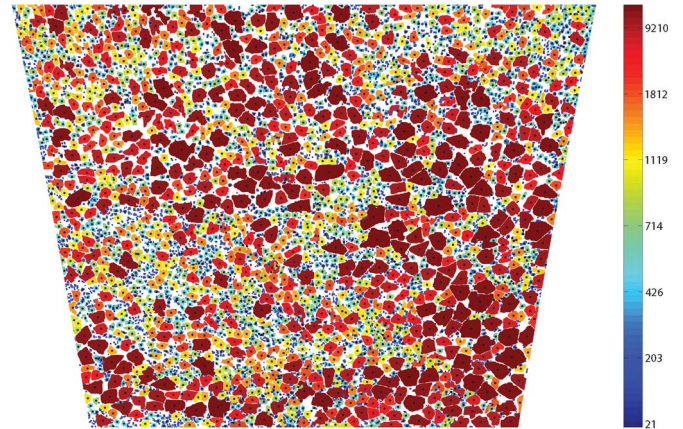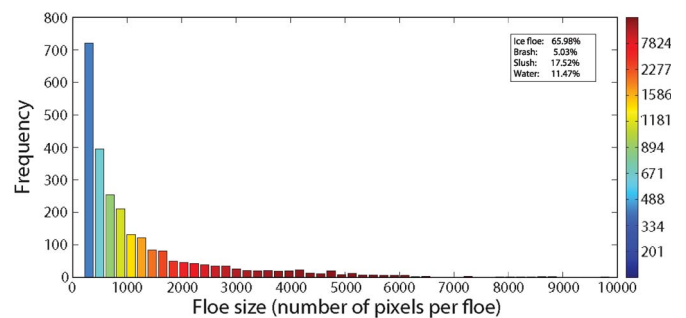
interpolation [36] therefore plays an important role in filling the values in those interpolated pixels by using known data to estimate values at unknown points. The calibrated image, however, may be blurred because the values of the interpolated pixels are not the real values captured from the objects; as the number of interpolated pixels increases, the objects in the calibrated image become more blurry. The ice floe boundaries may become weaker or even be lost. The floe boundaries will become more difficult to detect. If the geometric calibration is performed before the ice floe identification, the proposed algorithm may fail to detect the ice floes in the far range of the image because of their blurred boundaries.

C. Results

After orthorectification, we enhance the shapes of all the ice pieces (Algorithm 2), and finally, we obtain the ice floe and brash ice size distribution, as shown in Fig. 22. Brash ice is dark blue, smaller floes are light blue, and larger floes are red. Brash positions are not shown, whereas the floe positions are denoted using a black dot.

A total of 2511 ice floes and 2624 brash ice are identified in Fig. 19. The coverage percentages are 65.98% ice floe, 5.03% brash ice, 17.52% slush, and 11.47% water. Instead of the actual size of ice floe and brash (since we do not have the height above sea level for the camera), the ice floe (brash) size is calculated by the number of pixels in the identified floe (brash). The relative ice floe distribution histogram is derived and shown in Fig. 23, and the overall algorithm of the case study is concluded in Algorithm 3.

Algorithm 3 Overall Ice Floe and Brash Identification Algorithm

---

Input: Sea-ice image.
Start algorithm
1: $SUB \leftarrow$ sub-images divided from the input image.
2: **for** each sub-image $sub \in SUB$ **do**
3:     $seg \leftarrow$ ice edge detection (Algorithm 1) on sub.
4: **end for**
5: $SEG \leftarrow$ overall segmentation image with all seg stitched.
6: $SEG \leftarrow$ geometric calibrated SEG.
7: $IDENTIFICATION \leftarrow$ ice shape enhancement (Algorithm 2) on SEG.
8: $FLOE \leftarrow$ labeled regions in $IDENTIFICATION$ with large sizes.
9: $BRASH \leftarrow$ labeled regions in $IDENTIFICATION$ with small sizes.
10: $DISTRIBUTION \leftarrow$ floe size distribution.
11: **return** $FLOE, BRASH, DISTRIBUTION$.
Output: Segmented layers and Floe size distribution.

IV. CONCLUSION

A remote sensing mission yielded experience in data acquisition using a UAV. Various image processing methods were applied to a few samples of the collected sea-ice image data for analysis to retrieve important sea-ice information.

Focusing on identifying the nonridged ice floe in the marginal ice zone, and the managed ice resulting from offshore operations in sea ice, we proposed an algorithm to identify the individual ice floes in a sea-ice image using the GVF snake algorithm. To evolve the GVF snake automatically, "light ice" and "dark ice" were first obtained using the thresholding and k-means algorithms. The initial contours of both "light ice" and "dark ice" with proper locations and radii were then derived based on the local maxima from the distance transform. After ice edge detection, morphological cleaning was used to enhance floe shapes. The implementation on the sea-ice images, which contained multiple ice floes crowded together, is shown to give acceptable segmentation results.

APPENDIX
GEOMETRIC ORTHORECTIFICATION

The location of any object in the image is a function of the spatial orientation of the camera in relation to ground topography. Fig. 24 illustrates the relationship between the image coordinates and the orthorectification coordinates. The image coordinates lay in the focal plane and are denoted with small letters (x, y). The orthorectification plane coordinates are parallel to the ground and are denoted with capital letters (X, Y). The optic center of the camera is S. The camera *nadir line* intersects the orthorectification plane at the nadir T. The optic axis is perpendicular to the focal plane and intersects the center of the focal plane at principal point r, forming the shooting angle φ with the vertical nadir line. The optic axis extends to the orthorectification plane at point R. The principal line
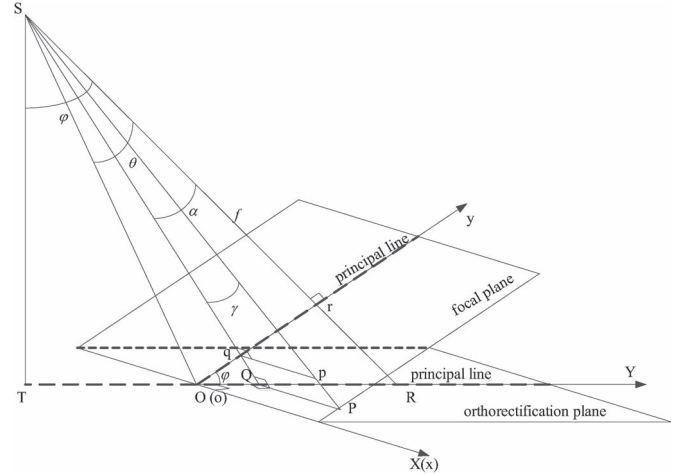

Fig. 24.   Geometric orthorectification.

passes through point $R(r)$ and the image border at O (or o), and it bisects the orthorectification (focal) plane. The point O (or o) acts as the origin for the image coordinate system, with the y-axis as the principal line, and the origin for the orthorectification coordinate system with the principal line in the orthorectification plane defining the positive Y-axis. $p(x, y)$ is any point on the image plane, and $P(X, Y)$ is the corresponding point on the orthorectification plane.

From Fig. 24, we obtain

$$TO = SO \sin(\varphi - \theta) = f \sec \theta \sin(\varphi - \theta) \tag{2}$$
$$TQ = ST \tan(\varphi + \vec{\alpha}) = f \sec \theta \cos(\varphi - \theta) \tan(\varphi + \vec{\alpha}) \tag{3}$$
$$SQ = ST \sec(\varphi + \vec{\alpha}) = f \sec \theta \cos(\varphi - \theta) \sec(\varphi + \vec{\alpha}) \tag{4}$$
$$\vec{\gamma} = \arctan(qp/Sq) = \arctan \frac{qp \cos \vec{\alpha}}{f} \tag{5}$$

where θ is half of the len's vertical field of view angle, f is the focal length, and $\vec{\alpha} = \arctan(\vec{qr}/f)$.

According to (2)–(5), we derive

$$OQ = TQ - TO$$
$$= f \sec \theta \left[\cos(\varphi - \theta) \tan\left(\varphi + \arctan \frac{\vec{qr}}{f}\right) - \sin(\varphi - \theta) \right] \tag{6}$$

$$QP = SQ \tan \gamma = qp \frac{\cos \vec{\alpha} \cos(\varphi - \theta)}{\cos \theta \cos(\varphi + \vec{\alpha})}$$
$$= qp \frac{f}{\sqrt{f^2 + \vec{qr}^2}} \frac{\cos(\varphi - \theta)}{\cos \theta \cos\left(\varphi + \arctan \frac{\vec{qr}}{f}\right)}. \tag{7}$$

A digital image is a numeric representation of a two-dimensional picture and is composed of pixels that are the smallest individual elements of the image. We assume that the pixel magnification of the image is μ, yielding

$$\begin{cases} oq = y \cdot \mu \\ qp = x \cdot \mu \end{cases} \tag{8}$$
$$\begin{cases} OQ = Y \cdot \mu \\ QP = X \cdot \mu. \end{cases} \tag{9}$$

Therefore

$$\vec{qr} = \vec{(oq)} - \vec{(or)} = \left(y - \frac{n_y}{2}\right) \cdot \mu \qquad (10)$$

where n_y is the number of pixels in image length. Instead of working from a $1:1$ positive, we counted the image pixels on a computer screen. Hence, the apparent focal length of the image is altered by

$$f' = \frac{y_e}{\tan\theta} = \frac{\mu \cdot n_y/2}{\tan\theta} \qquad (11)$$

where y_e is the half-length of the image. Substituting equations (8)–(11) into (6) and (7), the location of any point $P(X, Y)$ in the orthorectification coordinates can be determined from its image coordinates $p(x, y)$ by

$$\begin{cases} Y = \frac{n_y}{2}\csc\theta \left\{ \cos(\varphi-\theta)\tan\left[\varphi + \arctan\left(\frac{y-n_y/2}{n_y/2}\tan\theta\right)\right] \right. \\ \qquad\qquad \left. -\sin(\varphi-\theta) \right\} \\ X = x \cdot \dfrac{\frac{n_y}{2}\csc\theta\cos(\varphi-\theta)\sec\left[\varphi+\arctan\left(\frac{y-n_y/2}{n_y/2}\tan\theta\right)\right]}{\sqrt{\left(\frac{n_y/2}{\tan\theta}\right)^2 + \left(y-\frac{n_y}{2}\right)^2}} \end{cases}$$
$$(12)$$

and the image is orthorectified.

From (12), we can find that, although μ varies with the magnification ratio of the image, the relationship between different coordinates will not change. This is because the influence of μ is counteracted based on equations (8)–(11). Consequently, the relationship between the image coordinates and the orthorectification coordinates is a function of the shooting angle φ and the camera's (vertical) field of view 2θ.

ACKNOWLEDGMENT

REFERENCES

[1] S. Løset, K. N. Shkhinek, O. T. Gudmestad, and K. V. Hyland, *Actions from Ice on Arctic Offshore and Coastal Structures*. St. Petersburg, Russia: Lan Publishing House, 2006.

[2] W. Peter, *Ice in the Ocean*. New York, NY, USA: Taylor & Francis, 2000.

[3] T. Toyota and H. Enomoto, "Analysis of sea ice floes in the sea of Okhotsk using ADEOS/AVNIR images," in *Proc. 16th IAHR Int. Symp. Ice*, Dunedin, New Zealand, 2002, pp. 211–217.

[4] A. Keinonen, "Ice management for ice offshore operations," in *Proc. Offshore Technol. Conf.*, Houston, TX, USA, 2008, pp. 690–704.

[5] J. Hamilton *et al.*, "Ice management for support of arctic floating operations," in *Proc. OTC Arctic Technol. Conf.*, Houston, TX, USA, 2011, pp. 615–626.

[6] A. Keinonen and I. Robbins, "Icebreaker characteristics synthesis, icebreaker performance models, seakeeping, icebreaker escort," in *Icebreaker Escort Model User's Guide: Report Prepared for Transport Development Centre Canada (TP12812E)*, vol. 3. Calgary, AB, Canada: AKAC, 1998, p. 49.

[7] A. Palmer and K. Croasdale, *Arctic Offshore Engineering*. Singapore: World Scientific, 2012.

[8] C. Daley, S. Alawneh, D. Peters, B. Quinton, and B. Colbourne, "GPU modeling of ship operations in pack ice," in *Proc. Int. Conf. ICETECH Exhib. Perform. Ships Struct.*, Banff, AB, Canada, 2012, pp. 122–127.

[9] G. Vachon, M. Sayed, and I. Kubat, "Methodology for determination of ice management efficiency," in *Proc. Int. Conf. ICETECH Exhib. Perform. Ships Struct.*, Banff, AB, Canada, 2012, pp. 346–353.

[10] M. Sayed *et al.*, "Numerical simulations of ice interaction with a moored structure," in *Proc. Int. Conf. ICETECH Exhib. Perform. Ships Struct.*, Banff, AB, Canada, 2012, pp. 159–166.

[11] M. Sayed, I. K. Kubat, and B. Wright, "Numerical simulations of ice forces on the Kulluk: The role of ice confinement, ice pressure and ice management," in *Proc. OTC Arctic Technol. Conf.*, Houston, TX, USA, 2012, pp. 965–973.

[12] A. Gürtner, B. Bjørnsen, T. H. Amdahl, S. R. Sberg, and S. H. Teigen, "Numerical simulations of managed ice loads on a moored Arctic drillship," in *Proc. OTC Arctic Technol. Conf.*, Houston, TX, USA, 2012, pp. 914–920.

[13] I. Metrikin, S. Løset, N. A. Jenssen, and S. Kerkeni, "Numerical simulation of dynamic positioning in ice," *Marine Technol. Soc. J.*, vol. 47, no. 2, pp. 14–30, Mar./Apr. 2013.

[14] B. Wright *et al.*, "Evaluation of full scale data for moored vessel stationkeeping in pack ice," B. Wright & Associates Ltd., Canmore, AB, Canada, PERD/CHC Report 26-200, 1999.

[15] J. Haugen, L. Imsland, S. Løset, and R. Skjetne, "Ice observer system for ice management operations," in *Proc. 21st Int. Ocean Polar Eng. Conf.*, Maui, HI, USA, 2011, p. 1120.

[16] P. Lu, Z. Li, Z. Zhang, and X. Dong, "Aerial observations of floe size distribution in the marginal ice zone of summer Prydz Bay," *J. Geophys. Res., Oceans*, vol. 113, no. C2, pp. C02011-1–C02011-11, Feb. 2008.

[17] Q. Zhang, R. Skjetne, S. Løset, and A. Marchenko, "Digital image processing for sea ice observation in support to Arctic DP operation," presented at the Proc. 31st International Conf. Ocean, Offshore Arctic Engineering, Rio de Janeiro, Brazil, 2012, Paper OMAE2012-83860.

[18] M. Eldridge *et al.*, "Design and build a search and rescue UAV," Univ. Adelaide, Adelaide, Australia, 2009.

[19] T. Toyota, S. Takatsuji, and M. Nakayama, "Characteristics of sea ice floe size distribution in the seasonal ice zone," *Geophys. Res. Lett.*, vol. 33, no. 2, pp. L02616-1–L02616-4, Jan. 2006.

[20] T. Toyota, C. Haas, and T. Tamura, "Size distribution and shape properties of relatively small sea-ice floes in the Antarctic marginal ice zone in late winter," *Deep Sea Res. II, Top. Studies Oceanogr.*, vol. 58, no. 9/10, pp. 1182–1193, May 2011.

[21] Q. Zhang, S. van der Werff, I. Metrikin, S. Løset, and R. Skjetne, "Image processing for the analysis of an evolving broken-ice field in model testing," presented at the 31st Int. Conf. Ocean, Offshore Arctic Engineering, Rio de Janeiro, Brazil, 2012, Paper OMAE2012-84117.

[22] J. Blunt, V. Garas, D. Matskevitch, J. Hamilton, and K. Kumaran, "Image analysis techniques for high Arctic, deepwater operation support," in *Proc. OTC Arctic Technol. Conf.*, Houston, TX, USA, 2012, pp. 986–996.

[23] Q. Zhang, R. Skjetne, and B. Su, "Automatic image segmentation for boundary detection of apparently connected sea-ice floes," in *Proc. 22nd Int. Conf. Port Ocean Eng. Arctic Conditions*, Espoo, Finland, 2013, pp. 303–310.

[24] J. Banfield, "Automated tracking of ice floes: A stochastic approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 29, no. 6, pp. 905–911, Nov. 1991.

[25] J. D. Banfield and A. E. Raftery, "Ice floe identification in satellite images using mathematical morphology and clustering about principal curves," *J. Amer. Stat. Assoc.*, vol. 87, no. 417, pp. 7–16, Mar. 1992.

[26] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998.

[27] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.

[28] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, Berkeley, CA, USA, 1967, vol. 1, pp. 281–297.

[29] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.

[30] A. Rosenfeld and J. L. Pfaltz, "Distance functions on digital pictures," *Pattern Recognit.*, vol. 1, no. 1, pp. 33–61, Jul. 1968.

[31] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.

[32] L.-K. Soh, C. Tsatsoulis, and B. Holt, "Identifying ice floes and computing ice floe distributions in SAR images," in *Analysis of SAR Data of the Polar Oceans*. Berlin, Germany: Springer-Verlag, 1998, pp. 9–34.

[33] WMO SEA-ICE NOMENCLATURE, (accessed 2014-08-31). [Online]. Available: http://www.aari.ru/gdsidb/XML/volume1.php?lang1=0&arrange=4

[34] P. Lu and Z. Li, "A method of obtaining ice concentration and floe size from shipboard oblique sea ice images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2771–2780, Jul. 2010.

[35] T. Lippmann and R. A. Holman, "Quantification of sand bar morphology: A video technique based on wave dissipation," *J. Geophys. Res., Oceans*, vol. 94, no. C1, pp. 995–1011, Jan. 1989.

[36] P. Smith, "Bilinear interpolation of digital images," *Ultramicroscopy*, vol. 6, no. 2, pp. 201–204, 1981.

Qin Zhang received the M.Sc. degree in signal and information processing from the East China University of Science and Technology, Shanghai, China, in 2010. She is currently working toward the Ph.D. degree in image and sensory data processing within the Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Her research interests include remote sensing, image and sensory data processing.

Roger Skjetne (M'99) received the M.Sc. degree in control engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2000 and the Ph.D. degree in 2005 from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, for which the thesis was awarded the Exxon Mobil Prize for best Ph.D. thesis in applied research.

Prior to his studies, he was an Electrician at Aker Elektro AS, working on numerous oil installations for the North Sea. In 2004–2009, he was with Marine Cybernetics AS, working on hardware-in-the-loop simulation for testing safety-critical marine control systems. Since August 2009, he has been a Professor in marine control engineering with the Department of Marine Technology, NTNU, where he is presently the Leader of the Research Group on Marine Structures. His research interests are within Arctic stationkeeping operations and ice management systems for ships and rigs, environmentally robust control of shipboard electric power systems, and nonlinear control theory for motion control of single and groups of marine vessels.

Prof. Skjetne serves as the Project Manager for the KMB Arctic DP research project, the Leader of the ice management work package in the Center for Research-based Innovation (CRI) Sustainable Arctic Marine and Coastal Technology (SAMCoT), and an Associated Researcher in the Centre of Excellence "Centre for Ships and Ocean Structures" (CeSOS) and "Autonomous Marine Operations and Systems" (AMOS).