# Image Processing for Ice Floe Analyses in Broken-ice Model Testing

Q. Zhang[a,*], R. Skjetne[a], I. Metrikin[b,c], S. Løset[b]

[a] *Dept. Marine Technology, Norwegian Univ. of Science and Technology (NTNU), Trondheim, Norway*
[b] *Dept. Civil and Transport Engineering, NTNU, Trondheim, Norway*
[c] *Statoil ASA, Trondheim, Norway*

## Abstract

The ice floe shape and size distribution are important ice parameters in ice-structure analyses. Before performing an analysis at full scale, the Dynamic Positioning (DP) experiments in model ice at the Hamburg Ship Model Basin (HSVA) allow for the testing of relevant image processing algorithms. A complete overview image of the ice floe distribution in the ice tank was generated from the experiments. An image processing method based on a Gradient Vector Flow (GVF) Snake and a distance transform is proposed to identify individual ice floes. Ice floe characteristics such as position, area, and size distribution are obtained. A model of the managed ice field's configuration, including identification of overlapping floes, is also proposed for further studies in ice-force numerical simulations. Finally, the proposed algorithm is applied to an ice surveillance video to further illustrate its applicability to ice management.

*Keywords:*
connected object segmentation; model ice floe; GVF Snake; distance transform

## 1. Introduction

Operations in ice-covered regions are more technically and physically challenging than those in other accessible regions due to their low temperatures, remoteness, darkness, and to the presence of sea ice and icebergs. Because the behavior of ice forces is significantly different from other environmental forces, the development of temporally and spatially continuous field observations of sea ice conditions, such as ice concentration and ice floe size distributions is very important to mitigate the risks of such operations.

The use of cameras as sensors for offshore operations in ice-covered regions is explored to characterize the ice conditions. Camera sensors have the advantage of delivering spatially continuous measurements with high precision, which can be particularly important for providing detailed localized information about the ice to ensure safe operations in ice-covered regions. Image processing can reduce or suppress the ambiguities, incompleteness, uncertainties, and errors of an object and the environment and consequently produce more accurate and reliable information of the object and environment [1]. Various types of remote sensing technologies and corresponding image processing algorithms for analysis of sea-ice statistics and ice properties have been developed. Rothrock and Thorndike [2] measured the sea ice floe size distribution by identifying the ice floes manually from aerial photographs. Ji et al. [3] determined the ice thickness, ice velocity, and ice concentration in the Bohai Sea by using the sea ice digital image collection and processing

[*] Corresponding author

*Email addresses:* qin.zhang@ntnu.no (Q. Zhang), roger.skjetne@ieee.com (R. Skjetne), imet@statoil.com (I. Metrikin), sveinung.loset@ntnu.no (S. Løset) system. In the model tests performed by Millan and Wang [4], a machine vision system based on boundary detection and thresholding was used to analyze and record the ice conditions surrounding the vessel in real time.

For an actual ice image, the boundary hidden by an apparent connection between ice floes should be identified. Traditional boundary detection algorithms cannot easily detect this boundary, which seriously affects the floe size analysis. To mitigate this issue, Toyota et al. [5, 6] separated closely distributed ice floes by setting a threshold higher than the ice-water segmentation threshold. However, this threshold did not work well when the ice floes were connected. Consequently, they separated the connected ice floes manually. Zhang et al. [7, 8] applied and compared derivative and morphology boundary detection algorithms in both model ice and sea ice images. Traditional derivative boundary detection is sensitive to weak boundaries and noise, and it often produces non-closed boundaries. In contrast, morphology boundary detection results in a good description of the object shape and generates closed boundaries, but some boundary information is still lost. Blunt et al. [9] adopted the Watershed transform, which has been widely used in connected object segmentation, to separate the connected sea ice floes into individual floes. They removed brash ice by using image opening and erosion operators literally, and then they used the Watershed transform to segment the connected ice floes. However, over- and under-segmentation of the ice floes are the major issues in watershed-based segmentation. Due to an ineluctable over-segmentation problem, Blunt et al. removed these over-segmented lines manually. Zhang et al. [10] assumed that each ice floe had a convex boundary and that the junction line between two connected ice floes had at least one concave ending point. After the Watershed transform the convexity of each pair of ending points was checked, and two neighboring floes whose junction line ending points were both convex were merged to remove the over-segmentation automatically. Banfield [11] and Raftery [12] introduced a mathematical morphology together with principal curve clustering to identify ice floes and their boundaries in a nearly fully automated manner. First, the image is binarized using the thresholding method. The erosion-propagation algorithm (EP) is then used to provide a preliminary clustering of the boundary pixels and to produce a collection of objects as floe candidates. To remove subdivisions caused by the EP algorithm, they developed a method based on an algorithm for clustering about closed principal curves to determine which floes should be merged. However, both Zhang et al. [10] and Banfield et al.[12] operated on the binary images, their methods focusing on the morphological characteristics of ice floes rather than on the real boundaries, and they were limited by crowded ice floe images where the mass of ice floes were connected to each other and no "hole" or concave regions could be found after binarization.

In this research, a Gradient Vector Flow (GVF) Snake algorithm is adopted to separate seemingly connected floes into individual ones. The GVF Snake operates on the gray scale image in which the real boundary information, particularly "weak" boundaries, has been preserved. Moreover, the GVF Snake will ensure that the detected boundary is a closed curve. To start the algorithm, an initial contour is required for the GVF Snake. This initial contour should be close to the desired contour in the image. Otherwise, the snake may evolve incorrectly. Therefore, a manual initialization is required in some cases, particularly in connected and crowded floes segmentation. To solve this problem, an automatic contour initialization is proposed to avoid user interaction and to reduce the time required to run the GVF Snake algorithm. Once individual ice floes have been extracted, the floe boundaries are obtained, and the floe size distribution and shape factor can be extracted from the images.

The main focus of this paper is on model ice images. The algorithm has been tested on different sub-images and on an overall ice tank image. This results in an identified set of broken ice floes, which can be used in several ways:

- To quantify the efficiency of ice management for Arctic offshore drilling operations and automatically detect hazardous conditions, for example, by identifying large floes that escape the icebreakers operating upstream of the stationary drilling vessel. The size and shape of those floes, as identified by the image processing system, can be compared with the maximal allowed values, and a warning signal can be sent to the risk management system. Eventually, a decision to disconnect the floater might be taken, based on the identified operational ice conditions.

- The managed ice concentration and ice floe sizes are essential parameters in the empirical formulas that estimate the ice loads on stationary Arctic offshore structures (Keinonen and Robbins [13]; Palmer and Croasdale [14]). One of the largest concerns of ice management modeling is accurately predicting not just the mean floe size resulting from an ice management system, but the floe size distribution [15].
- Individual ice floes identified by the image processing system, can be used to initialize high-fidelity numerical models, such as those in Daley et al. [16], Vachon et al. [17], Sayed et al. [18], Sayed et al. [19], Gürtner et al. [20] and Metrikin et al. [21]. Individual snapshots of identified ice floes can be used to validate the numerical models at various moments in time by matching the simulated ice fields with the actual ones.
- The ice floe size and shape distribution, calculated from an identified ice field, can be used in synthetic ice field generators. These generators draw polygons from the distribution and use packing algorithms to place the polygons on a 2D plane. Such synthetic ice fields may be used to study various packing configurations with the same ice concentrations and floe size distributions as well as the variability of the resulting ice loads on an offshore structure.
- The identification of the ice field may 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, as defined by Wright [22] and discussed in Palmer and Croasdale [14].
- Finally, the ice-drift speed and direction (velocity) can be estimated by applying an image analysis to sequential frames. The ice-drift velocity is an important parameter for ice management because it poses requirements on the speed of icebreaking vessels and may indicate an approaching ice drift reversal scenario (which usually happens when the ice drift tends to zero velocity).

In addition to the above application areas, the ice floe identification algorithm may potentially help to illuminate the momentum exchange from atmosphere to ice discussed in [23], the melting rate of ice floes discussed in [24], and possibly providing a clue to the understanding of ice-floe formation processes as discussed in [25].

## 2. Snake models

Snakes, or active contours as introduced in [26], correspond to a powerful method used to locate object boundaries. The initial curves 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. There are two types of active contour models: parametric active contours and geometric active contours. This study considered the parametric active contours due to its superior detection capability of “weak”-boundaries.

### *2.1. Parametric Snake Model*

A typical snake is a curve $\mathbf{C}(s) = (x(s), y(s)), s \in [0, 1]$ that moves through the spatial domain of an image to minimize the sum of the internal and external energy. This energy is given by

$$\mathbf{E} = \int_0^1 (\mathbf{E}_{int}(\mathbf{C}(s)) + \mathbf{E}_{ext}(\mathbf{C}(s)))\mathrm{d}s, \tag{1}$$

where $\mathbf{E}_{int}$ is the internal energy

$$\mathbf{E}_{int} = \frac{1}{2}(\alpha|\mathbf{C}^{'}(s)|^2 + \beta|\mathbf{C}^{''}(s)|^2), \tag{2}$$

where $\alpha$ and $\beta$ are weight parameters that control the snake's tension and rigidity, respectively. $\mathbf{C}^{'}(s)$ denotes the first derivatives of $\mathbf{C}(s)$ with respect to $s$, making the snake act as a membrane, and $\mathbf{C}^{''}(s)$ denotes the second derivatives, making the snake act as a thin plate.

$\mathbf{E}_{ext}$ is the external energy defined in the image domain. It attracts snakes to salient features in the image, such as boundaries. To find boundaries in a gray scale image, the image gradient is typically chosen as the external energy [26]

$$\mathbf{E}_{ext} = -|\nabla \mathbf{I}(x, y)|^2, \tag{3}$$

where $\nabla \mathbf{I}(x, y) = \left(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y}\right)$ is the image gradient that represents a directional change in the brightness of the image with the gradient angle $\theta = \arctan\left(\frac{\partial \mathbf{I}}{\partial y} / \frac{\partial \mathbf{I}}{\partial x}\right)$. When also considering the image noise, the external energy is defined as [26]

$$\mathbf{E}_{ext} = -|\nabla \mathbf{G}_\sigma(x, y) * \mathbf{I}(x, y)|^2, \tag{4}$$

where $\nabla \mathbf{G}_\sigma(x, y)$ is a two-dimensional Gaussian function with a standard deviation $\sigma$ and $*$ denotes convolution.

To minimize the energy $\mathbf{E}$, a snake must satisfy the Euler equation

$$\alpha\mathbf{C}^{''}(s) - \beta\mathbf{C}^{''''}(s) - \nabla\mathbf{E}_{ext} = 0. \tag{5}$$

Let $\mathbf{F}_{int} = \alpha\mathbf{C}^{''}(s) - \beta\mathbf{C}^{''''}(s)$ denote the internal force and $\mathbf{F}_{ext} = -\nabla\mathbf{E}_{ext}$ denote the external force. Then (5) can be written as the force balance

$$\mathbf{F}_{int} + \mathbf{F}_{ext} = 0. \tag{6}$$

The internal force $\mathbf{F}_{int}$ discourages stretching and bending, while the external potential force $\mathbf{F}_{ext}$ pulls the snake towards the desired image boundaries.

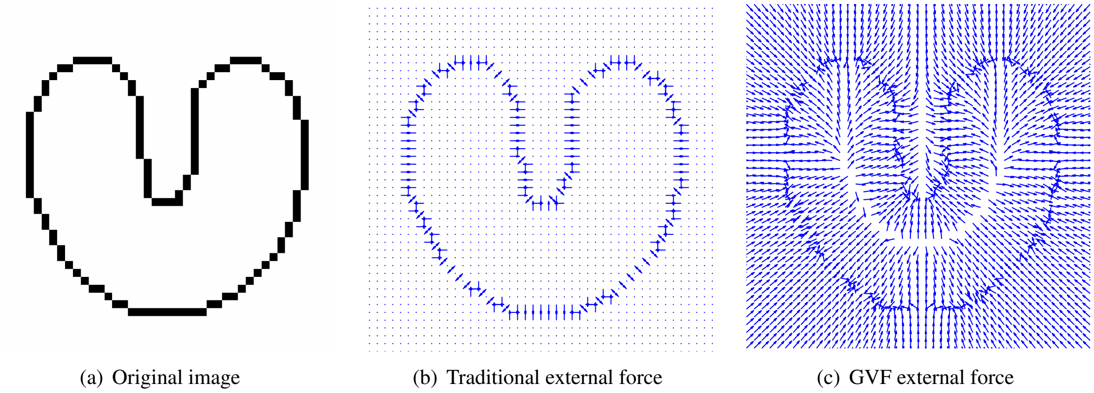
To find a solution of (5), $\mathbf{C}(s)$ is treated as a discrete system of $s$ and $t$

$$\frac{\partial \mathbf{C}(s, t)}{\partial t} = \alpha\mathbf{C}^{''}(s, t) - \beta\mathbf{C}^{''''}(s, t) - \nabla\mathbf{E}_{ext}. \tag{7}$$

When the solution $\mathbf{C}(s, t)$ becomes stationary, $\frac{\partial \mathbf{C}(s,t)}{\partial t}$ tends to zero, the energy $\mathbf{E}$ reaches a minimum, and the curve converges towards the target boundary.

### *2.2. Gradient Vector Flow Snake*

The traditional snake algorithm can solve a number of image segmentation problems effectively, particularly in detection of "weak"-boundaries. However, there are two key limitations. First, the capture range of the external force fields is limited. The external forces $\mathbf{F}_{ext}$ have large values near the boundaries and small values in the homogeneous regions. Consequently, it is difficult for a curve to converge in regions of low variations in intensity. It is therefore sensitive to the initial contour, and the initial contour should be somewhat close to the true boundary. Otherwise, the curve will likely converge to an incorrect result. Second, it is difficult to progress into boundary concavities. According to these limitation, Xu and Prince [27] introduced the Gradient Vector Flow (GVF) Snake to expand the capture range of the external force fields from the boundary regions to the homogeneous regions (see Figure 1).


(a) Original image

(b) Traditional external force

(c) GVF external force

Figure 1: External forces.

The GVF, "computed as a spatial diffusion of the gradient of an edge map derived from the image" [28], is defined to be the vector field $\mathbf{v}(x,y) = (u(x,y), v(x,y))$ that minimizes the energy functional

$$\epsilon = \iint [\mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2] \mathrm{d}x\mathrm{d}y, \tag{8}$$

where $u_x, u_y, v_x, v_y$ are the derivatives of the vector field, $\mu$ is a parameter that controls the balance between the first and second term in the integrand, and $f$ is an edge map (which could be the image gradient $|\nabla \mathbf{I}(x,y)|^2$) that is larger near the edges of objects in the image.

In (8), $|\nabla f|$ becomes large close to the object boundaries in which case the second term dominates the integrand and is minimized by $\mathbf{v} = \nabla f$. Otherwise, $|\nabla f|$ is small, and the first term dominates the integrand to ensure that the external force field vary slowly and still act in the homogeneous regions.

The GVF field can be found by solving the Euler equations

$$\mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0, \tag{9a}$$

$$\mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0. \tag{9b}$$

A solution to (9a) and (9b) can be obtained by introducing a time variable $t$ and finding the steady-state solution of the following partial differential equations

$$u_t(x,y,t) = \mu \nabla^2 u(x,y,t) - (u(x,y,t) - f_x(x,y))(f_x(x,y)^2 + f_y(x,y)^2), \tag{10a}$$

$$v_t(x,y,t) = \mu \nabla^2 v(x,y,t) - (v(x,y,t) - f_y(x,y))(f_x(x,y)^2 + f_y(x,y)^2). \tag{10b}$$

Compared to the external force field in the traditional snake model having only $f_x$ and $f_y$, the new vector fields $u$ and $v$ in the GVF are derived using an iterative method to find a solution for $f_x$ and $f_y$. The result is that the capture range is effectively enlarged and the initial contour no longer needs to be as close to the true boundary.

## 3. Contours initialization

The GVF Snake is faster and less restricted by the initial contour. However, a proper initial contour is still necessary because the snake deforms itself to conform with the nearest salient contour.

### *3.1. Locating initial contours*

Figure 2 presents the floe boundary detection results affected by initializing the contour at different locations. The red circles in the first row of Figure 2 are the initial contours, and the curve evolutions are shown by the red curves in the second row. The green curves are the final detected boundaries. In Figure 2(a), the initial circle is located at the water, and the contour is close to the ice boundaries. The snake rapidly finds the boundaries. However, the detected boundaries are not the boundaries of the ice but the boundaries of the water region. When initializing the circle at the center of an ice floe, as shown in Figure 2(b), the snake finds the boundary accurately after a few iterations even if the initial contour is a bit distance away from the floe boundary.

The weak boundary will also be detected, if the initial circle is located at a weak boundary, as shown in Figure 2(c). However, when the initial circle is near the floe boundary and inside of the floe, as shown in Figure 2(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 Figure 2(c) and Figure 2(d) which appear to be non-closed curves. This behavior occurs because the area bounded by the closed curve tends toward a small value, even zero.

Figure 2 illustrates that using the proper parameters, the snake will always find a boundary regardless of where the initial contour is located. This fact is beneficial for connected floe segmentation. By comparing the results of Figure 2, the results where the initial contours are located inside of the floes are more effective, whereas the most efficient case is the one in which the initial contour is in the center of the ice floe. Thus, the initial contour should be located as close to the floe center as possible.

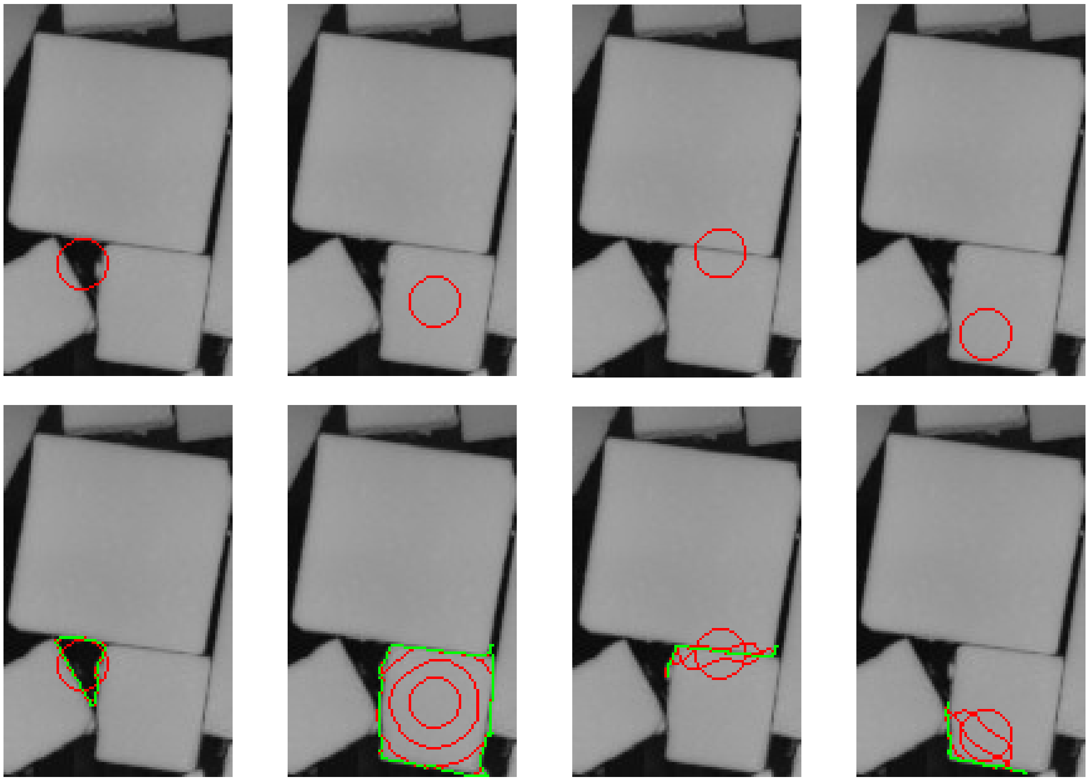
In ice image analyses, the model ice floes can be easily separated from water and converted into a binary image using a thresholding method [7]. The thresholding method makes it easy to locate the initial contours inside of the ice floes. We propose to use the distance transform [29] and its local maxima to locate the initial contours as close to the floe centers as possible.

Given a binary image $f$, whose elements only have values of '0' and '1', the pixels with a value of '0' indicate the background, while the pixels with a value of '1' indicate the object. Let $B = \{(x, y)|f(x, y) = 0\}$ be the set of background pixels and $O = \{(x, y)|f(x, y) = 1\}$ be the set of object pixels. The distance transform of a binary image $f$ is the minimum distance from every pixel in an object $O$ to the background $B$, that is

$$\mathbf{D}(p) = \begin{cases} 0 & \text{if } p \in B \\ \min_{q \in B} d(p, q) & \text{if } p \in O, \end{cases} \tag{11}$$

where $d(p, q)$ is some distance measure between $p$ and $q$ [30].

Figure 3 illustrates a distance transform for a simple shape. Figure 3(a) presents a small binary image matrix and the corresponding distance transform. Herein we are using 'City Block' distance metrics. The local



(a) Initial contour 1.

(b) Initial contour 2.

(c) Initial contour 3.

(d) Initial contour 4.

Figure 2: Initial contours located at different positions and their corresponding curve evolutions.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(a) Binary image matrix.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 2 | 2 | 2 | 1 | 0 |
| 0 | 1 | 2 | 3 | 2 | 1 | 0 |
| 0 | 1 | 2 | 2 | 2 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(b) Distance transform of (a).

Figure 3: Distance transform.

maximum is the pixel whose value is greater than any of its neighborhoods (e.g., the red numeral shown in Figure 3(b)). A local maximum of the distance transform ideally corresponds to the center of an object, but more than one local maximum is detected in many cases (see, e.g., the red crosses in Figure 4(c)). Thus, we propose to combine the local maxima within a short distance of each other using a dilation operator [31]. The centers of the dilated regions, which we call seeds, will be the locations of our initial contours. The segmentation result is illustrated in Figure 4(e), where very small ice floes, called "rubbles", were removed.

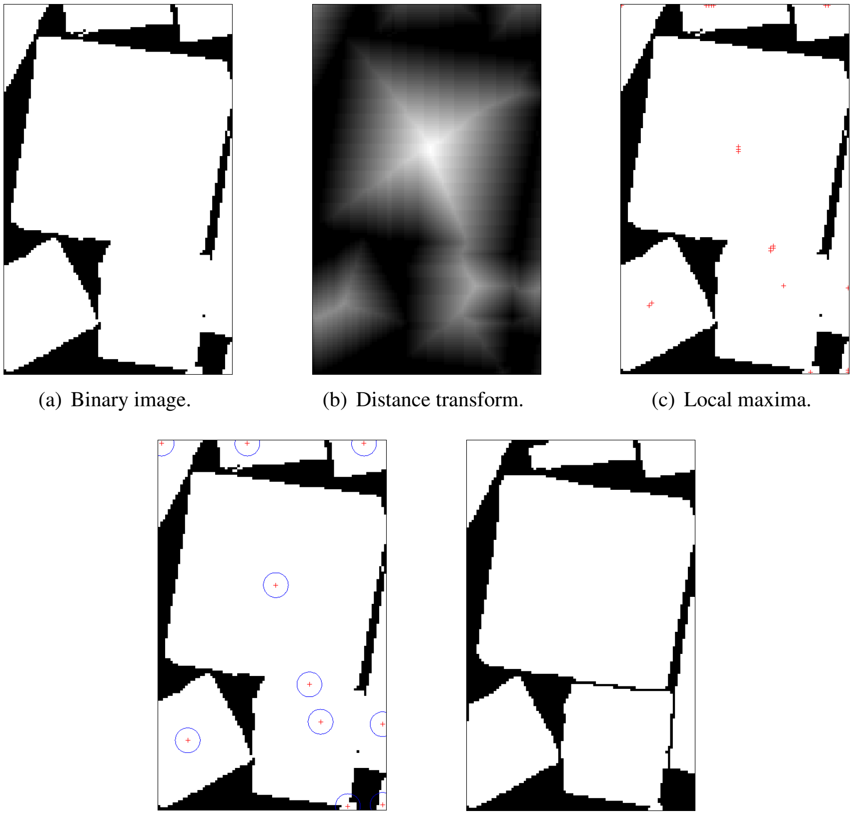
### *3.2. Crowded ice floes*

Figure 4(e) presents an acceptable result of the proposed algorithm applied in a case with only two or three connected ice floes. In a complex case (e.g., Figure 5(a)), multiple ice floes are crowded together. It then becomes more difficult to locate the seeds for each ice floe (Figure 5(b)) with the result that some connected ice floes could not be separated (Figure 5(c)). It follows that an additional round of contour initialization and segmentation is necessary.

In this model ice test, the ice floes were modeled square shapes with predefined side lengths. Hence, the largest floe has an area less than a predefined value. Although they are not perfect squares, most of the floes could be approximated as rectangles with a length-to-width ratio less than the given threshold. Based on these characteristics, we use three criteria to determine whether it is necessary to initialize the contours and conduct a second segmentation:

- The ice floe area is less than the given threshold.
- The ice floe has a convex shape (the ratio between the floe area and its minimum bounding polygon area is larger than the threshold).
- The length-to-width ratio of the minimum bounding rectangle of the ice floe is less than the threshold.

After a segmentation step, the algorithm will stop if all the identified floes satisfy these criteria. Otherwise, the algorithm must find the floes that do not satisfy any of these criteria, calculate their distances, find the new seeds, initialize new contours, and perform the segmentation again. Some boundaries may exist that are too weak to be detected, and there may be some floes that do not satisfy the criteria after a new step. However, the total number of identified floes will converge to a final solution. Therefore, the algorithm is made to stop if the total number of floes identified after step $N$ and $N + 1$ are equal, in combination with an absolute stop criterion.



(a) Binary image.

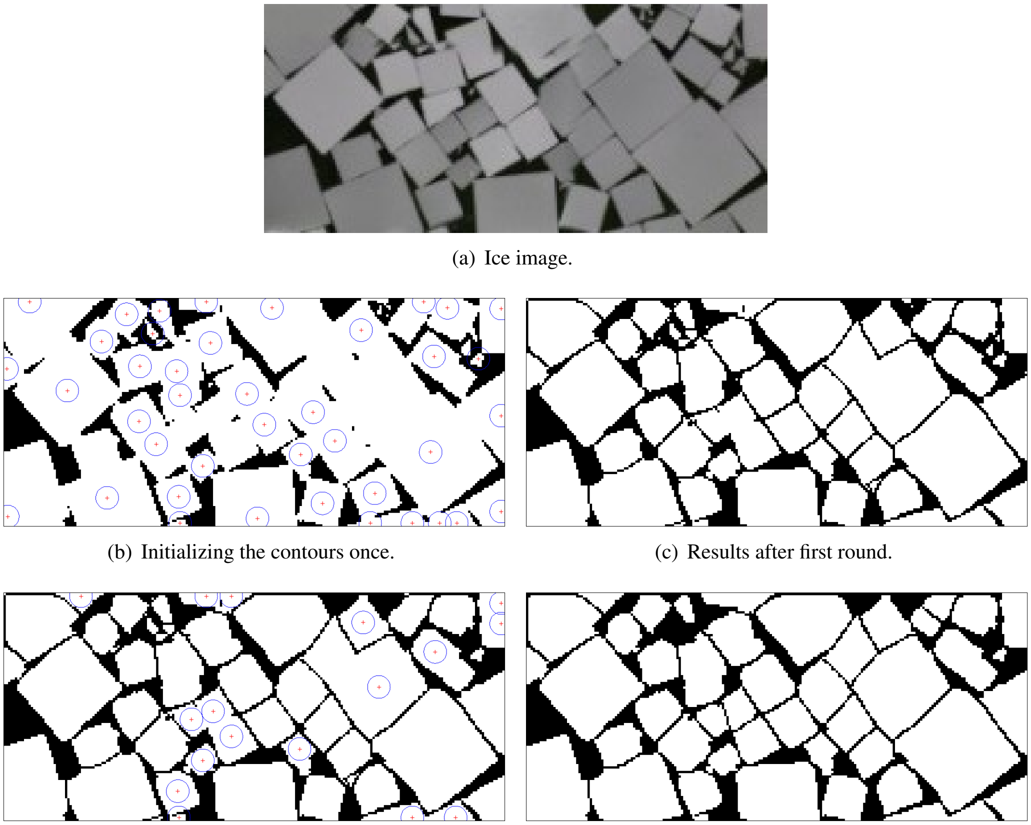
(b) Distance transform.

(c) Local maxima.

(d) Initial contours.

(e) Segmentation result.

Figure 4: Contour initialization and segmentation result.



(a) Ice image.

(b) Initializing the contours once.

(c) Results after first round.

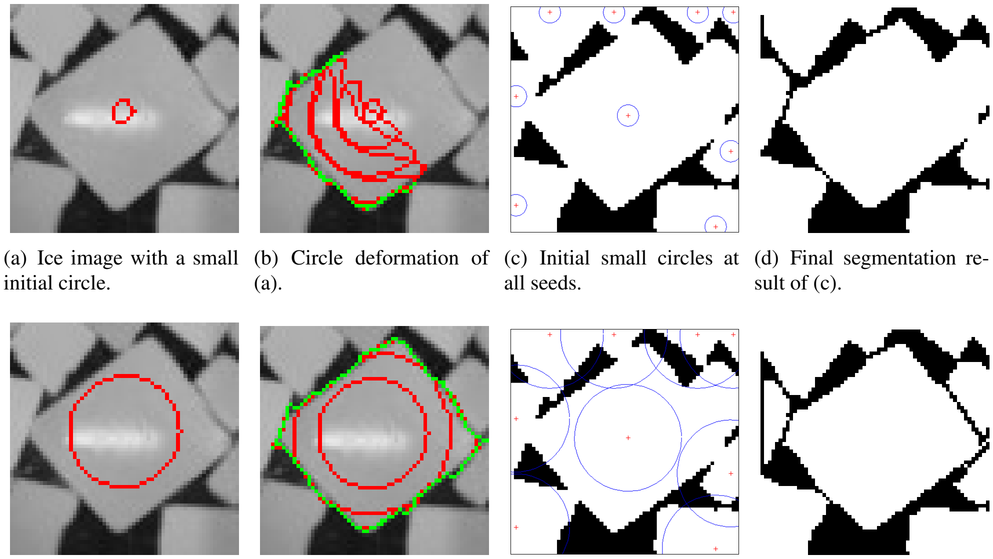
(d) Initializing the contours twice.

(e) Final result.

Figure 5: Crowded ice floes segmentation.

### 3.3. *Initial circle radius*

Because the model ice floes were modeled as square shapes, that is, highly symmetric shapes, the initial contours with circular shapes located at the floe center could deform uniformly toward to the floe boundary, unaware of the floe's orientation. As mentioned in Section 3.1, the initial contours should be located inside the floes to increase the algorithm's effectiveness. Hence, the initial circle radius should not be larger than half of the square floe length. However, if the radius is too small, the initial circle will be lightly "far away" from the floe boundary, and it will need a longer deformation time to find the correct boundary (it may even likely converge to an incorrect result [27]). Take Figure 6 as an example. This example contains some light reflection in the middle of the ice floe, and the pixels that belong to the reflection are lighter than the other pixels of the floe. This phenomenon will affect the boundary detection when the initial circle is too small. As shown in Figure 6(a) and 6(c), when the initial circle is too small, the algorithm uses many steps and does not find the complete of floe boundary (Figure 6(b) and 6(d)). If we enlarge the initial circle (Figure 6(e) and 6(g)), the initial circle allows for a faster determination of the entire floe boundary (Figure 6(f) and 6(h)).

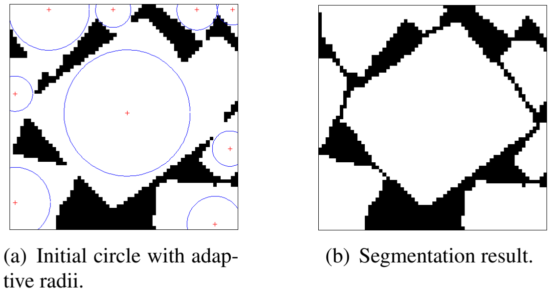


(a) Ice image with a small initial circle. (b) Circle deformation of (a). (c) Initial small circles at all seeds. (d) Final segmentation result of (c).

(e) Ice image with a large initial circle. (f) Circle deformation of (e). (g) Initial large circles at all seeds. (h) Final segmentation result of (g).

Figure 6: Initial circles with different radii and their curve evolutions.

To approach the actual floe boundary, the initial circle radius should be adapted to the floe size. To accomplish this, we have chosen to use the distance transform again. The pixel value at the seed in the distance map (e.g., the red numeral shown in Figure 3(b)) is chosen as the initial circle radius. This choice will ensure that the initial circle is inside the floe and will iteratively approach the floe boundary using adaptive initial circles and the corresponding segmentation result is shown in Figure 7.



(a) Initial circle with adaptive radii.

(b) Segmentation result.

Figure 7: Initial circles with adaptive radii and the segmentation result.

## 4. Algorithm overview

According to Section 3, we propose the following algorithm to segment the model ice floe image. First, derive the GVF from the gray scale input image. Then, separate the ice floes from water by using the thresholding method, and label the floes. Each labeled floe should then be checked to determine whether it satisfies all of the criteria of Section 3.2. Next, apply the distance transform to the floes that do not satisfy any of the criteria and find their seeds and radii. Finally, based on the seeds and radii, initialize the circles and run the snake algorithm. The pseudocode of the proposed algorithm, as well as a maximum iteration time, is given in Algorithm 1.

## 5. Experimental results and discussions

A model ice test was carried out at the ice model basin of the HSVA in the summer of 2011 [7]. A managed ice condition was obtained by cutting a manufactured level ice layer into square pieces with specific dimensions and distributing them over a specific testing area. The cutting procedure was as follows: First, several strips of ice were cut in the longitudinal direction of the ice tank. Next, these strips were cut off such that the length was equal to the width of the strip, resulting in square ice floes. After preparing the ice field and before the test run started, a top-view camera was positioned over the total ice covered basin to produce an overall image of the complete ice field.

To test the validity of the proposed algorithm, it has been tested on both sub images and an overall ice tank image

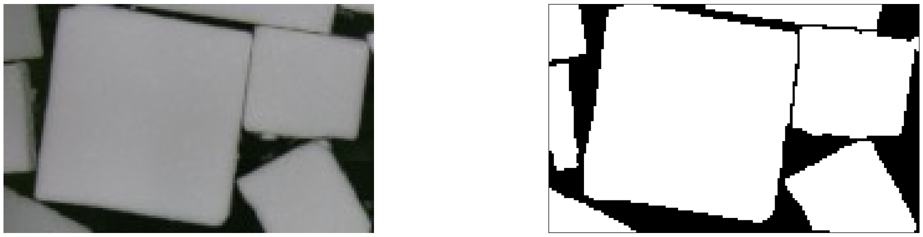
### *5.1. Sub-image tests*

The sub-images were manually extracted from the overall ice tank image, and the algorithm was tested on both the simple and complex cases. Figure 8(a) presents the simplest case, that is, only two or three ice floes connected with clear boundaries. The proposed algorithm found all of the seeds in the first iteration and rapidly located the boundaries. In the case illustrated in Figure 9(a), many ice floes are crowded in the image. After several iterations of seed locating, the algorithm produced satisfactory results in the high density part of the image, while failing to find a few weak boundaries. In addition, the algorithm also failed to detect a number of ice floe boundaries connected to the border of the image as shown in Figure 10. It is because of the incompleteness of the ice floes that some boundary information was weaker. The junctions between a number of ice floes and ice rubble were also too weak to be detected. Because the area of the ice

**Algorithm 1** Model ice floe segmentation.

```
Input: Model ice image
Start algorithm:
 1: $T \leftarrow$ max iteration time
 2: $N_0 \leftarrow 0$
 3: $GVF \leftarrow$ GVF derived from gray-scale of input image
 4: $BW \leftarrow$ binary image
 5: $FLOE \leftarrow$ labeled ice floes in $BW$
 6: $N_1 \leftarrow$ total number of ice floes in $BW$
 7: if $N_0 \neq N_1$ && $T \neq 0$ then
 8:     for each labeled $floe \in FLOE$ do
 9:         $f \leftarrow$ floes that do not satisfy any of the criteria
10:     end for
11:     $k \leftarrow$ number of $f$
12:     if $k \neq 0$ then
13:         $S \leftarrow$ Seeds of $f$ found by local maxima of distance transform
14:         for each seed $s \in S$ do
15:             $r \leftarrow$ local maxima values at $s$
16:             $ic \leftarrow$ initial contours locate at $s$ with its radius $r$
17:             $B \leftarrow$ boundary detected by performing the snake algorithm on $ic$
18:             $BW \leftarrow BW$ with $B$ superimposed
19:         end for
20:     end if
21:     $N_0 \leftarrow N_1$
22:     $T \leftarrow T - 1$
23:     go back to 4
24: end if
25: return $BW$
Output: Segmented image
```

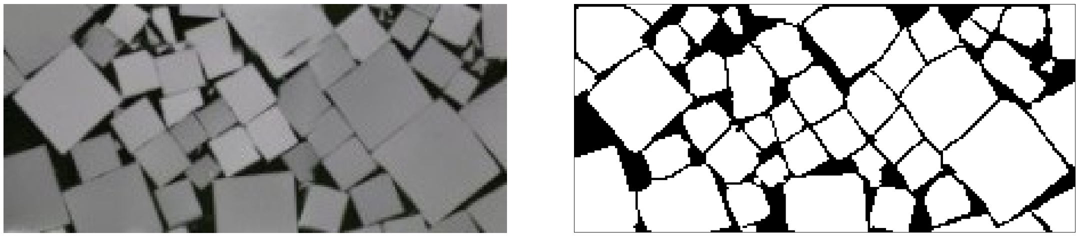
rubble is too small relative to the area of its connected ice floe, the snake was rapidly attracted to the rubble boundaries when it deformed at the junction. The most complex case is shown in Figure 11(a), where the crowded ice floes were aligned in the image. The segmentation result indicates that the proposed algorithm still performs well with the exception of a number of border and rubble effects.



(a) Ice image 1.

(b) Segmentation 1.

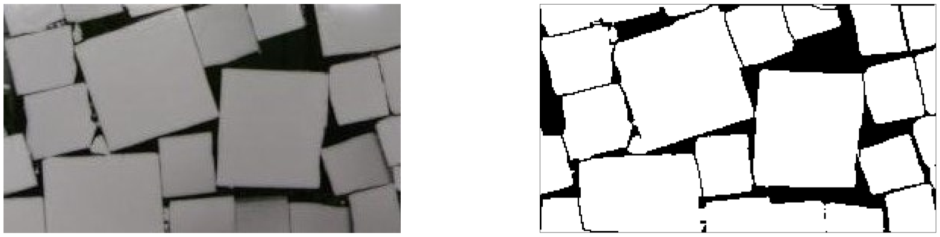
Figure 8: Segmentation result 1.



(a) Ice image 2.

(b) Segmentation 2.

Figure 9: Segmentation result 2.



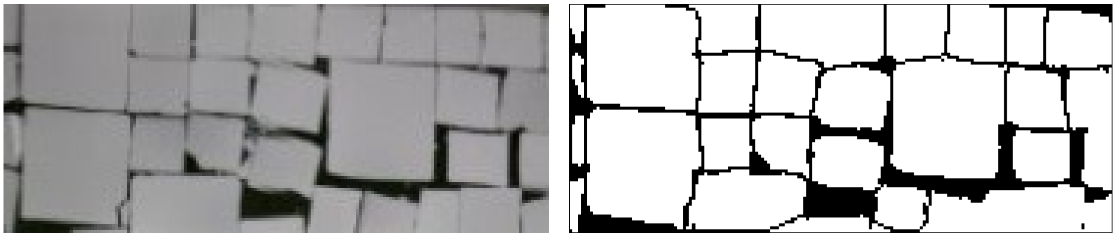
(a) Ice image 3.

(b) Segmentation 3.

Figure 10: Segmentation result 3.

### *5.2. Overall ice tank image*

The overall ice tank image, obtained by a top view camera, provides a complete overview of the ice floe distribution in the ice tank (see Figure 12). The lightness of the image is rather uneven, and the nonuniform



(a) Ice image 4.

(b) Segmentation 4.

Figure 11: Segmentation result 4.

illumination will affect the segmentation result. An illumination correction, such as homomorphic filtering [32], is adopted to preprocess the image. After performing the proposed algorithm, the identified ice floes may contain black spots inside the floes because of noise. Those spots are filled in post-processing. The segmentation result is shown in Figure 13(a). The floes are labeled using different colors based on their areas according to

$$\mathbf{Color}(p) = \begin{cases} 0 & \text{if } p \notin FLOE \\ (1 - \exp(-area(i)/1000)) * 10000 & \text{if } p \in floe(i), \end{cases} \tag{12}$$

where $FLOE = \{floe(1), floe(2), floe(3), \cdots\}$ is a set of identified floes, $floe(i) \in FLOE$, and $area(i)$ is the area of $floe(i)$. Smaller floes are blue, and larger floes are red. The floe positions, found by averaging the positions of the pixels of each floe, are denoted using black dots. The corresponding size distribution histogram is presented in Figure 13(b).

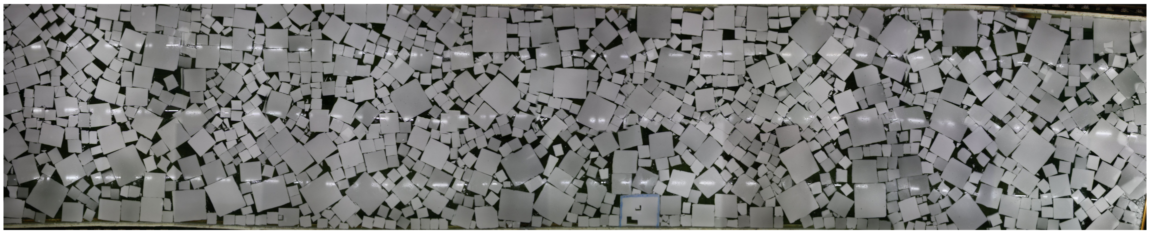
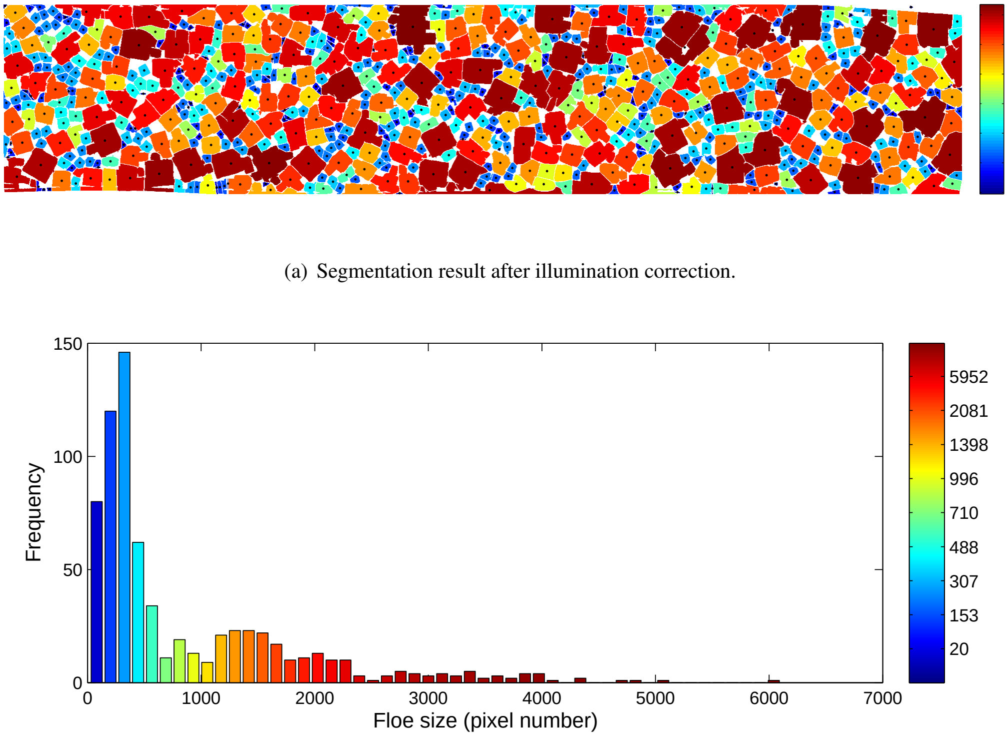


Figure 12: Overall ice tank image

However, the segmentation result shown in Figure 13(a) is not ideal. A number of over- and under-segmentations still exist because a uniform parameter for the GVF field cannot represent the overall ice tank image. For instance, if we switch the GVF capture ranges of Figure 10 and Figure 11, then error segmentations will occur (see Figure 14). Usually, weak boundaries tend to be more difficult to detect when increasing the GVF capture range, which results in under-segmentation. In contrast, if the capture range is decreased, the noise is enhanced, which leads to over-segmentation. In an image containing overlapping ice floes as shown in Figure 15(a), the overlapping part is brighter than the rest of the floe. When segmenting the image using a low GVF capture range, boundaries are extracted, but the two overlapped floes are identified as three individual floes (Figure 15(b)). When increasing the GVF capture range, the weaker boundary is lost, and the two overlapped floes are divided. However, one of the floes loses half of its area. Thus, processing the local sub-images of the overall ice tank image locally is recommended.

The segmentation of the floes situated at the boundary of the analyzed image is more prone to error. It is not



(a) Segmentation result after illumination correction.

(b) Ice floe size distribution histogram of (a).

Figure 13: Ice floe identification after illumination correction and floe size distribution.

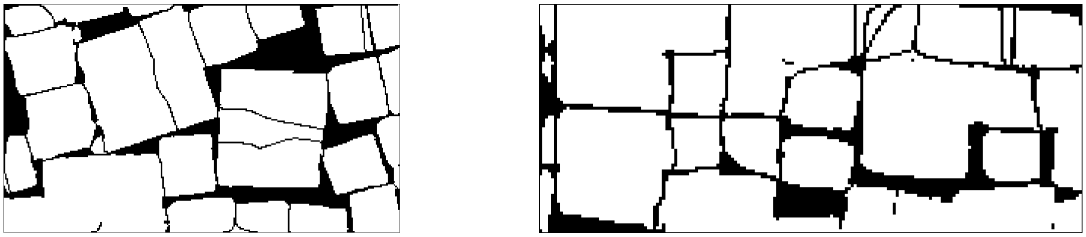
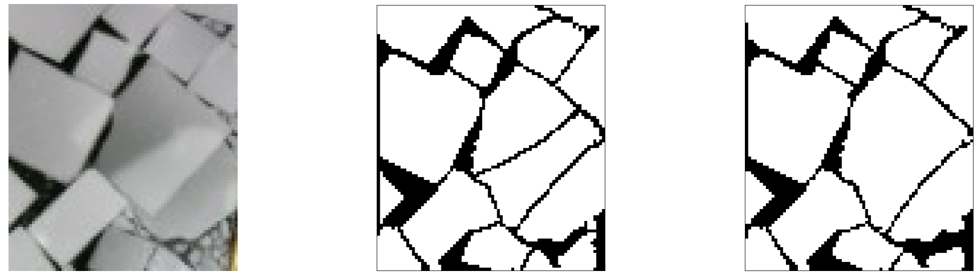


Figure 14: Error segmentation by using improper parameters.



(a) Overlapping floes image.

(b) Segmentation result via low GVF capture range (100 iterations for GVF).

(c) Segmentation result via high GVF capture range (150 iterations for GVF).

Figure 15: Overlapping floes image segmentation.

Table 1: The parameter values of the GVF field for each sub-image.

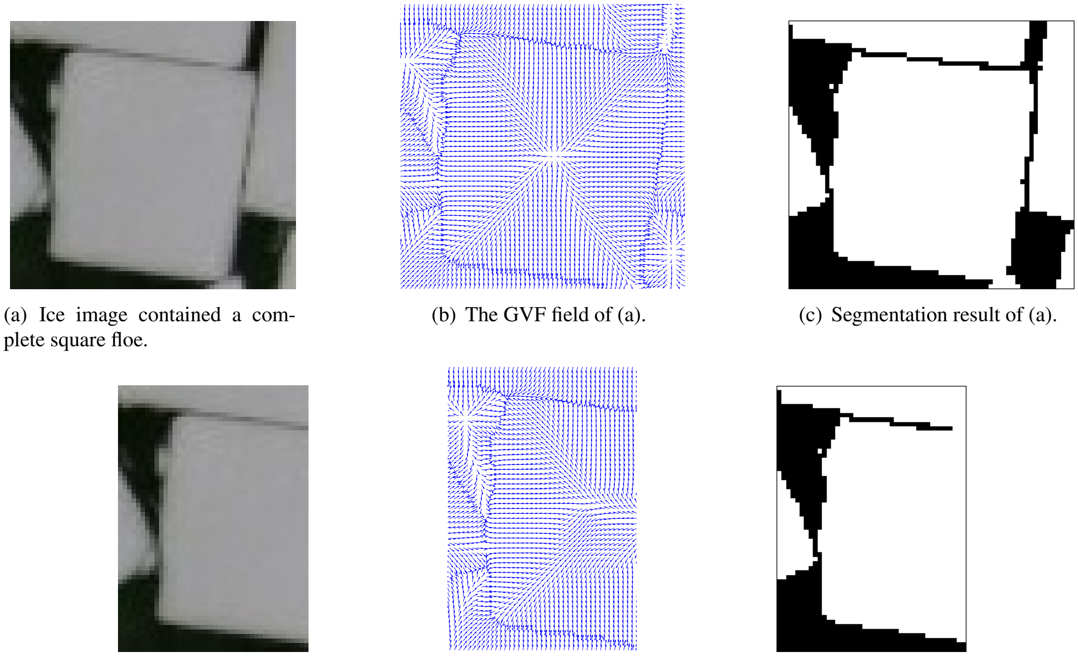
| Sub-image No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| The GVF iteration number | 150 | 65 | 65 | 65 | 160 | 60 | 110 | 130 | 90 | 130 | 170 | 160 | 100 | 90 | 90 | 100 | 90 | 80 | 90 | 80 |

only because these floes, being "cut" by the image border, do not fulfill the criteria formulated in Section 3.2, but also because the incompleteness of the floe influences the GVF field. As shown in Figure 16, the GVF field of a complete floe radiates from the floe center to its boundaries. The proposed algorithm then successfully finds all the boundary pixels. Contrary to this, if some of the floe boundary is lost due to an image border, the GVF field of this incomplete floe will radiate from the image border to the floe boundaries, with the result that the segmentation by the snake algorithm fails.

To derive a more precise result, the overall image is first divided into several smaller, overlapping (to avoid image border effects) sub-images, and we perform the proposed algorithm locally on each sub-image. The parameter values of the GVF field for each sub-image are listed in Table 1. The final segmentation result and its floe size distribution histogram are shown in Figures 17(a) and 17(b), respectively.

By comparing the color distribution in Figures 13(a) and 17(a) and their histograms, one can see that there are more blue ice floes and fewer red floes in Figure 17(a) than in Figure 13(a). This result indicates that a greater number of connected floes have been separated and that under-segmentation has been reduced in Figure 17(a). The minor of under- and over-segmentation that still exists in Figure 17(a) is due to "overly weak" boundaries and a lingering border effect. This issue can be improved by adjusting the number, position, and size of the sub-images. A larger number of smaller sub-images produces better results at the expense of a larger stitching operation.

Light reflection also affects the segmentation results. Several lights were installed at the bottom of the ice tank to supply a sufficient brightness for the ice observation. However, these lights are detrimental to the ice image analysis because they can be identified as ice floes due to their brightness, and the light reflection on the ice floe may induce erroneous segmentation. As shown in Figure 18, the initial circle meets a strong light reflection when deforming, and some boundaries around the reflection become too weak to be detected. Hence, a part of the circle deformed toward the reflection rather than toward the true floe boundary. Fortunately, the light reflection did not significantly affect our result, but we still suggest that the lights be disabled or that a polarizer be placed in the front of camera before taking the picture.



(a) Ice image contained a complete square floe.

(b) The GVF field of (a).

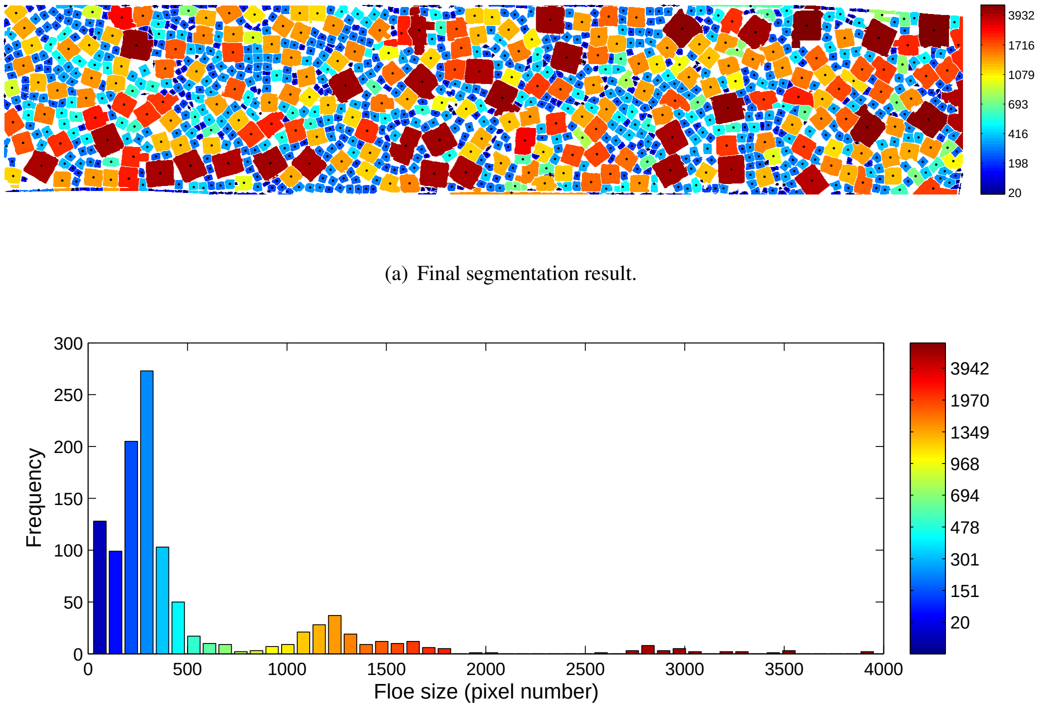
(c) Segmentation result of (a).

(d) Ice image contained an incomplete square floe.

(e) The GVF field of (d).

(f) Segmentation result of (d).

Figure 16: Image border effects.



(a) Final segmentation result.

(b) Ice floe size distribution histogram of (a).

Figure 17: Final ice floe identification result and floe size distribution.

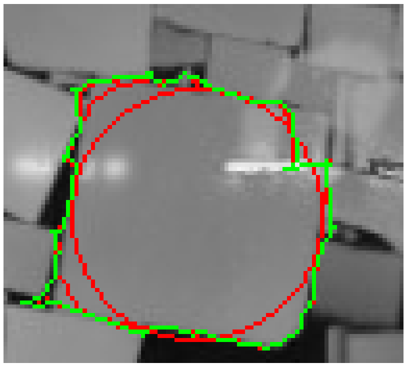
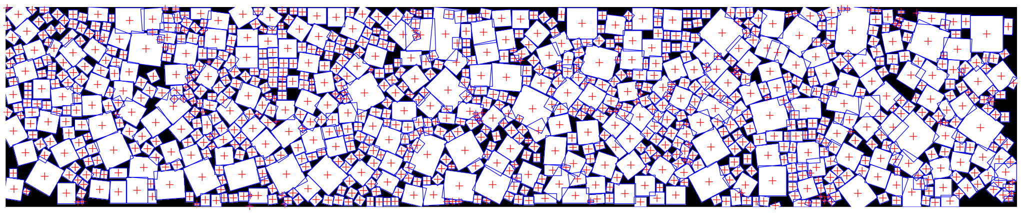


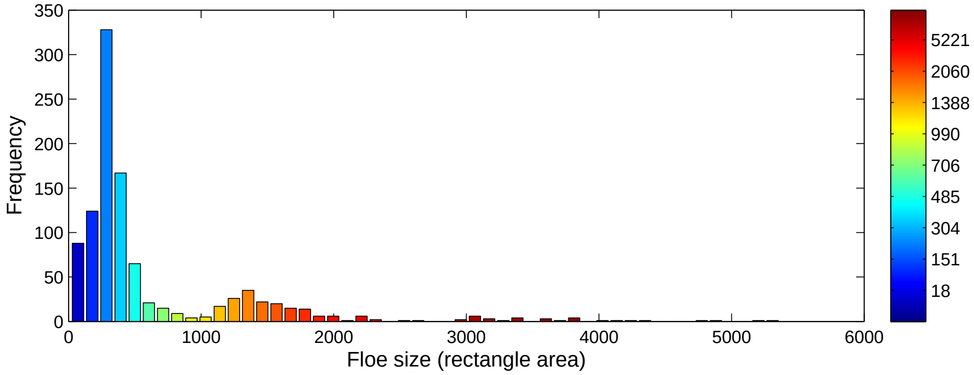
Figure 18: Light reflection impact.

### *5.3. Ice floe model*

In the numerical simulation of the ice-structure interaction, all of the ice floes are modeled as rectangular floes, and the positions of the vertices are important to an ice-structure analysis. Therefore, we proposed to perform ice floe rectangularization. Ice floe rectangularization was achieved by assigning the minimum area bounding rectangle to each ice floe. Due to under- and over-segmentation, the rectangles with a length-to-width ratio less than the given threshold were removed. The final rectangularization result is shown in Figure 19(a). If the floes are not segmented well, the rectangularized floes will be smaller or larger than the actual segmented floes, and some floes will overlap. To indicate overlaps, a "flag" was added to each rectangular floe to record the serial number of the floes with which the current floe overlaps. Identifying the overlapping between ice floes is important when using the resulting image analysis and identified ice floes as a starting condition for the validation of a numerical simulation with a 3D capacity against a real test and data set.
The colorized histogram of the rectangular floe size distribution is presented in Figure 19(b). Under-segmentation could induce a large area difference as well as overlapping, which explains why the largest floe in Figure 19(a) is much larger than the largest identified floe in Figure 17(a), as seen by comparing Figure 19(b) and Figure 17(b).



(a) Rectangularization of Figure 17(a).



(b) Ice floe size distribution histogram of (a).

Figure 19: Ice floe rectangularization and floe size distribution.

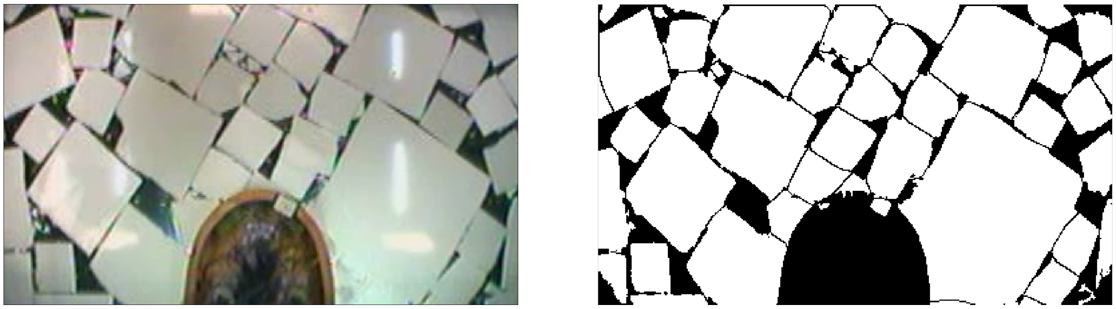
### 5.4. *Ice concentration*

The ice concentration for an image is the total area of ice relative to the image area. The ice concentration can be estimated by counting the number of pixels for each respective area in the image. The target ice concentration in this test was $90\%$. The ice concentration derived from Figure 17(a) is $76.96\%$, while it is $83.17\%$ when calculated using the threshold method [7]. In the proposed algorithm, the ice pixels detected as a boundary were changed to water pixels (e.g., Figure 15), so the calculated ice concentration was reduced in Figure 17(a).

The ice concentration calculated by summing all the areas of the rectangular floes in Figure 19(a) over the image domain is $87.75\%$. This value is slightly higher than the thresholding result because the overlapping parts have been identified and considered. The overlapping parts compensate for the loss of ice concentration, and increase the calculated ice concentration closer to the target value.

## 6. Application: Monitoring maximum floe size

An example of the proposed algorithm is to monitor the maximum floe size entering the protected vessel from a physical ice management operation. In the analyzed runs, the heading of the Arctic Drillship [7] was constant at 180° and the velocity of the towing carriage with the model was increased halfway during the run. From the full scale velocity of 0.25 to 0.50 knots was simulated. The motion captured by the video is retrieved by analyzing at one frame per second.The impediments around the tank were removed first (as shown in Figure 20(a)) [7], the proposed algorithm was applied to segment the connected ice floe (Figure 20(b)), and the maximum floe area for each frame were calculated. Figure 21 presents the maximum floe size entering the protected vessel as a function of time. Based on this result, a warning can be sent to the risk management system if the estimated risk based on the maximum floe size is too large.



(a) Pre-processed frame at 965s.

(b) Segmentation result of (a).

Figure 20: Video processing.

## 7. Conclusion

We have proposed an automatic contour initialization procedure for the GVF Snake algorithm to separate seemingly connected model ice floes into individual ice floes. Based on the characteristics of a model ice image, we obtained the seeds by calculating the distance transform of the binarized image. Based on these

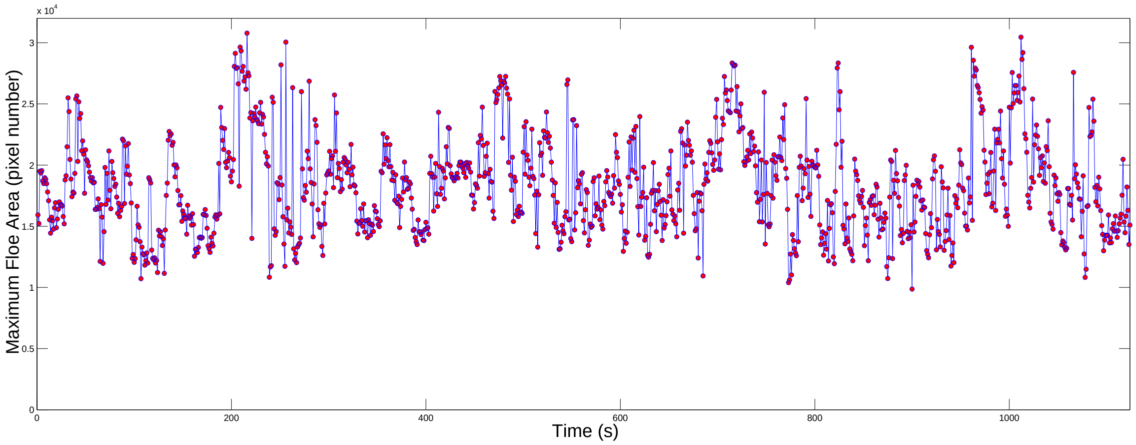


Figure 21: Maximum floe size entering the protected vessel.

seeds, the snake contours were initialized and the GVF Snakes were evolved. The experiments on the model ice images produce acceptable segmentation results and demonstrate that the proposed algorithm is effective for connected model ice floe separation. Furthermore, the validation of numerical simulation models based on the identified broken ice and further application were also presented in this paper.

## 8. Acknowledgments

The authors acknowledge the Research Council of Norway (RCN) for funding the Arctic DP project through Project No. 199567/I40 and the MARTEC ERA-NET project DYPIC through Project No. 196897, which supplied experimental data from the Hamburg Ship Model Basin. The authors would also like to thank the industry partners: Statoil ASA, Kongsberg Maritime, Det Norske Veritas (DNV), and the Hamburg Ship Model Basin for their kind support.

[1] J. Haugen, L. Imsland, S. Løset, R. Skjetne, Ice observer system for ice management operations, in: Proceedings of 21st International on Offshore (Ocean) and Polar Engineers Conference, Maui, Hawaii, USA, 2011.

[2] D. Rothrock, A. Thorndike, Measuring the sea ice floe size distribution, Journal of Geophysical Research: Oceans (1978–2012) 89 (C4) (1984) 6477–6486.

[3] S. Ji, H. Li, A. Wang, Q. Yue, Digital image techniques of sea ice field observation in the bohai sea, in: International Conference on Port and Ocean Engineering under Arctic Conditions, Montral, Canada, 2011.

[4] J. Millan, J. Wang, Ice force modeling for DP control systems, in: Proceedings of the Dynamic Positioning Conference, Houston, Texas, USA, 2011.

[5] T. Toyota, S. Takatsuji, M. Nakayama, Characteristics of sea ice floe size distribution in the seasonal ice zone, Geophysical research letters 33 (2) (2006) L02616.

[6] T. Toyota, C. Haas, T. Tamura, Size distribution and shape properties of relatively small sea-ice floes in the Antarctic marginal ice zone in late winter, Deep Sea Research Part II: Topical Studies in Oceanography 58 (9) (2011) 1182–1193.

[7] Q. Zhang, S. van der Werff, I. Metrikin, S. Løset, R. Skjetne, Image processing for the analysis of an evolving broken-ice field in model testing, in: Proceedings of 31st International Conference on Ocean, Offshore and Arctic Engineering, OMAE2012-84117, ASME, Rio de Janeiro, Brazil, 2012.

[8] Q. Zhang, R. Skjetne, S. Løset, A. Marchenko, Digital image processing for sea ice observation in support to Arctic DP operation, in: Proceedings of 31st International Conference on Ocean, Offshore and Arctic Engineering, OMAE2012-83860, ASME, Rio de Janeiro, Brazil, 2012.

[9] J. Blunt, V. Garas, D. Matskevitch, J. Hamilton, K. Kumaran, Image analysis techniques for high Arctic, deepwater operation support, in: OTC Arctic Technology Conference, Houston, Texas, USA, 2012.

[10] Q. Zhang, R. Skjetne, B. Su, Automatic image segmentation for boundary detection of apparently connected sea-ice floes, in: Proceedings of the 22nd International Conference on Port and Ocean Engineering under Arctic Conditions, Espoo, Finland, 2013.

[11] J. Banfield, Automated tracking of ice floes: A stochastic approach, Geoscience and Remote Sensing, IEEE Transactions on 29 (6) (1991) 905–911.

[12] J. D. Banfield, A. E. Raftery, Ice floe identification in satellite images using mathematical morphology and clustering about principal curves, Journal of the American Statistical Association 87 (417) (1992) 7–16.

[13] A. Keinonen, I. Robbins, Icebreaker characteristics synthesis, icebreaker performance models, seakeeping, icebreaker escort, Icebreaker Escort Model User's Guide: Report prepared for Transport Development Centre Canada (TP12812E) 3 (1998) 49.

[14] A. Palmer, K. Croasdale, Arctic Offshore Enginnering, World Scientific Publishing Company, 2012.

[15] J. S. Brown, E. H. Martin, A. Keinonen, Ice management numerical modeling and modern data sources, in: The International Conference and Exhibition on Performance of Ships and Structures in Ice (ICETECH 2012), Banff, Alberta, Canada, 2012.

[16] C. Daley, S. Alawneh, D. Peters, B. Quinton, B. Colbourne, GPU modeling of ship operations in pack ice, in: The International Conference and Exhibition on Performance of Ships and Structures in Ice (ICETECH 2012), Banff, Alberta, Canada, 2012.

[17] G. Vachon, M. Sayed, I. Kubat, Methodology for determination of ice management efficiency, in: The International Conference and Exhibition on Performance of Ships and Structures in Ice (ICETECH 2012), Banff, Alberta, Canada, 2012.

[18] M. Sayed, I. Kubat, B. Wright, A. Iyerusalimskiy, A. Phadke, B. Hall, Numerical simulations of ice interaction with a moored structure, in: The International Conference and Exhibition on Performance of Ships and Structures in Ice (ICETECH 2012), Banff, Alberta, Canada, 2012.

[19] M. Sayed, I. K. Kubat, B. Wright, Numerical simulations of ice forces on the kulluk: the role of ice confinement, ice pressure and ice management, in: In Proceedings of OTC Arctic Technology Conference, Houston, Texas, USA, 2012.

[20] A. Gürtner, B. Bjørnsen, T. H. Amdahl, S. R. Sberg, S. H. Teigen, Numerical simulations of managed ice loads on a moored Arctic drillship, in: In Proceedings of OTC Arctic Technology Conference, Houston, Texas, USA, 2012.

[21] I. Metrikin, S. Løset, N. A. Jenssen, S. Kerkeni, Numerical simulation of dynamic positioning in ice, Marine Technology Society Journal 47 (2) (2013) 14–30.

[22] B. Wright, et al., Evaluation of full scale data for moored vessel stationkeeping in pack ice, PERD/CHC Report (1999) 26–200.

[23] M. Steele, J. H. Morison, N. Untersteiner, The partition of air-ice-ocean momentum exchange as a function of ice concentration, floe size, and draft, Journal of Geophysical Research: Oceans (1978–2012) 94 (C9) (1989) 12739–12750.

[24] M. Steele, Sea ice melting and floe geometry in a simple ice-ocean model, Journal of Geophysical Research: Oceans (1978–2012) 97 (C11) (1992) 17729–17738.

[25] T. Toyota, H. Enomoto, Analysis of sea ice floes in the sea of okhotsk using ADEOS/AVNIR images, in: Proceedings of the 16th IAHR International Symposium on Ice, Dunedin, New Zealand, 2002, pp. 211–217.

[26] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (4) (1988) 321–331.

[27] C. Xu, J. L. Prince, Snakes, shapes, and gradient vector flow, IEEE Transactions on Image Processing 7 (3) (1998) 359–369.

[28] C. Xu, J. L. Prince, `http://www.iacl.ece.jhu.edu/static/gvf/` (accessed 2014-09-23).

[29] A. Rosenfeld, J. L. Pfaltz, Sequential operations in digital picture processing, Journal of the ACM (JACM) 13 (4) (1966) 471–494.

[30] A. Rosenfeld, J. L. Pfaltz, Distance functions on digital pictures, Pattern Recognition 1 (1) (1968) 33–61.

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

[32] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 2nd Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.