EasyChair Preprint

№ 4453

Towards Optimal Ship Navigation Using Image Processing

Bekir Sahin, Zia Uddin and Ahmet Soylu

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

October 21, 2020

Towards Optimal Ship Navigation Using Image Processing

Bekir Sahin[1], Zia Uddin[2], and Ahmet Soylu[1]

[1] Norwegian University of Science and Technology, Gjøvik, Norway
{bekir.sahin, ahmet.soylu}@ntnu.no
[2] SINTEF AS, Oslo, Norway
zia.uddin@sintef.no

Abstract. Shipping transportation developed over years with the technological advancements. Modern ship navigation is conducted with the help of Automatic Radar Plotting Aid (ARPA) and Electronic Chart Display and Information System (ECDIS). Location map, marine traffic, geographical conditions, and obstacles in a region can be monitored by these technologies. The obstacles may vary from icebergs and ice blocks to islands, debris, rocks, or other vessels in a given vicinity. In this study, we propose an approach for route optimization using two-dimensional radar images and image segmentation in an environment with obstacles. The navigation algorithm takes image segmentation results as an input and finds the optimal route (i.e. safest and shortest). One of the advantages of this study is that the obstacles are not solely polygonal, but they may be in any shape, size, and color. The proposed approach has some practical and computational limitations; however, the future unmanned vessels could benefit from the improved applications of this route optimization approach in terms of energy consumption, time, and workforce.

Keywords: Navigation, Imaging, Route Planning, Path Optimization, Safety.

1 Introduction

Maritime transportation is of significance in terms of global merchandise, since %90 of cargoes is carried by commercial ships. According to United Nations Conference on Trade and Development (UNCTAD, 2016), the number of total ships navigating through the world is 90917 with 1.8 billion deadweight tons. Maritime transportation will maintain its crucial position in the future due to new shipping routes (i.e. Arctic region), active global economy (i.e. One Belt One Road), technological advancements, such as unmanned vessels, and new energy sources. The global demand pushes the shipping industry to be cheaper, involve less risks, and be eco-friendly. In this study, the ship navigation problem is investigated with the aim of finding an optimal route (shortest and safest simultaneously) in a given maritime environment with obstacles using image processing. The idea is to see whether an optimal path could be found from the radar images collected from Automatic Radar Plotting Aid (ARPA), an electronic navigation equipment, after determining the obstacles such as other vessels, islands,

and icebergs. Since we are using static radar snapshots, the obstacles in this study are static, but in any shape or size rather than polygonal. The shortest path algorithm is not deterministic; therefore, we do not directly include distance, size etc. Moreover, information regarding the weather and atmospheric conditions, such as wind, current, waves, and rain are not considered. One of the novelties of this study is that the safety distance to the obstacles can be predefined. We set it proportional to the size of obstacles to be more realistic.

In the literature, route optimization problem is studied by considerable number of scholars. In a study, the optimal path on a given graph is computed by considering the ship turn angles and safety zone [1]. However, the special coordinates of graph are predefined, which makes the approach semi-automatic and input dependent. In this study, we overcome this shortcoming by utilizing object segmentation regardless of spatial and temporal information. In other words, we do not need to have coordinates of obstacles as priori data. There are applications in various other domains, such as medical, robotics, security, etc. using image processing [2-10]. The segmentation of objects on medical images has been a central task for some time, which requires abstracting semantically significant information from the background [11]. In robotics, real-time tool/device navigation is performed by image-guided platforms, where an object can be traced through a set of sequential frames by state-of-the-art image processing algorithms. Security applications, such as anomaly detection also extensively studied in the previous decades, which are dedicated to detecting suspicious activity by analyzing the optical flow between the sequential image frames. To the best of our knowledge, this is the first study for optimizing ship navigation combining image processing and computational geometry. We processed simulated radar images as input and obtained the safest/shortest path using Dijsktra algorithm.

The rest of the paper is organized as follows. Section 2 provides the methodology in a stepwise manner. Section 3 demonstrates the application on an empirical study. Finally, Section 4 concludes the paper by discussing the limitations of this study and gives the future research perspective.

2 Methodology

The steps of the methodology are provided in a stepwise manner in the followings.

2.1 Radar Images

Due to the complex structure of radar images, we adopted a formal image segmentation method. This complexity stems from ice melting, which is represented as gradually changing intensity patches on an image. This requires us to employ robust image segmentation methods that tackle intensity inhomogeneity.

Active Contour Models (ACM) have been one of the most popular methods in image segmentation [12]. There are two classes of active contour models: region-based and edge-based. Edge-based models use image gradients to stop the contour evolution when the contour reaches to object boundary. Classical edge-based ACM have an edge-stopping and balloon force terms to adjust the motion of initial contour. The edge-based

stopping term is used for stopping the contour on the object boundary, whereas balloon force term is used for shrinking or expanding the active contour to provide flexibility for placing the initial curve far from the target object on image. Notably, the choice of balloon force is a challenging task such that if it is too large, evolving contour may leak through the weak boundaries. If the balloon force is not large enough, evolving contour may not snap bay-like parts of the irregularly shaped images [12]. Region-based active contours are preferable in image segmentation applications with weak object boundaries. The reason is that region-based models do not use image gradients. Additionally, the location of initial contour does not affect the segmentation results significantly. One of the most common region-based ACM is Chan-Vese method which has been successful for segmenting images with two regions that have distinct pixel intensity mean. Vese and Chan [13, 14] adopt a multi-phase level set formulation to discriminate multiple regions on an image. Due to the assumption that an image consists of statistically homogeneous segments, these models are called piecewise constant (PC) models [15]. However, the regions on images do not appear statistically homogeneous, and therefore PC models do not output desired segments on those types of images. To overcome these limitations, a piecewise smooth model is proposed [14]. A method to segment images with intensity inhomogeneity is also developed; however, both methods are computationally expensive [16].

Considering the structure of radar images on which melting ice is represented in a gradually decreasing intensity, we adopt the method proposed by Li et al. [12] so that our segmentation core can handle inhomogeneity. The approach is to utilize a kernel function to embed a local binary fitting energy into a region-based contour model [12]. Here, the energy functional is minimized via variational level sets without re-initialization [17].

Mumford Shah Functional. Mumford-Shah is one of the earliest segmentation models using variational calculus to minimize the energy in a formal way. Mumford Shah model aims to find an optimal contour C that separates the sub-regions from each other, and a u function that is smooth approximation of the image I. The energy functional proposed [18] is given as follows:

$$F^{MS}(u,C) = \int_{\Omega} (I-u)^2 \, dxdy + \nu \int_{\Omega \setminus C} \left|\nabla u\right|^2 \, dxdy + \nu \left|C\right| \tag{1}$$

where $|C|$ represents the length of contour C.

Nevertheless, non-convexity of the functional and unknown set C and u make energy minimizing difficult in Equation (1). To address these limitations, some alternative works have been proposed, i.e. Chan-Vese piecewise constant (PC), and Chan-Vese piecewise smooth (PS) methods.

Piecewise Constant Model. Chan-Vese modified Mumford-Shah functional based on an assumption that the image u consists of two sub-regions, which are piecewise constant. In other words, pixel intensities of these regions are represented by two constants.

The energy functional that Chan-Vese aims to minimize for an image *I(x)* is given as follows:

$$E^{CV}(C,c_1,c_2) = \lambda_1 \int_{C_{in}} \left|I(x)-c_1\right|^2 dx + \lambda_2 \int_{C_{out}} \left|I(x)-c_2\right|^2 dx + v\left|C\right| \qquad (2)$$

where C_{out} and C_{in} represent the region outside and inside, c_1 and c_2 are two constants to approximate the pixel intensity inside and outside of the contour *C*. First two terms in Equation (2) are called global binary fitting energy. This energy can be minimized by the steepest descent method representing the contour *C* by zero level set. Notably, in Equation (2), c_1 and c_2 are related to global image properties, interior and exterior of the contour respectively. However, such image properties do not suffice if the image displays intensity and inhomogeneity inside and outside the contour.

Piecewise smooth model and its difficulties. The PS model is proposed to address the limitations of PC model regarding image inhomogeneity [14, 16]. Rather than approximating an image *I* by a piecewise constant function, the PS model utilizes two smooth functions *u+(x)* and *u-(x)* in the sub-regions of interior and exterior of the contour *C*, which can be expressed mathematically as in the following: $\Omega^+ = \{x \in \Omega : \phi(x) < 0\}$ and $\Omega^- = \{x \in \Omega : \phi(x) < 0\}$, respectively. Function *u+(x)* and *u-(x)* are defined on mutually exclusive sub-regions respectively. However, the level set function is defined on the full domain. Additionally, smoothing the terms of *u+(x)* and *u-(x),* which are represented as $\mu\int\left|\nabla u^+\right|^2 dx$ and $\mu\int\left|\nabla u^-\right|^2 dx$ complicates the minimization of the energy functional in such a way that these terms enforce the solution of two partial differential equations (PDEs) with the two unknown *u+* and *u-* in each iteration throughout level set solution. Moreover, practical domain re-initialization of the level set function to a signed distance function is required. Obviously, the need for extension of *u+* and *u-* and the re-initialization of PDEs for *u+* and *u-* increase the computational cost of PS model.

Local Binary Fitting (LBF). Li et al. developed the LBF model incorporating the local image information in segmentation problem [12]. LBF can handle intensity inhomogeneities and is computationally more efficient than PS model. The main idea is to introduce a kernel function to set a LBF energy functional as in the following:

$$\begin{aligned} &E^{LBF}(\phi, f_1, f_2) = \int_\Omega E_x^{LBF} dx = \\ &\lambda_1 \int \left[\int K_\sigma(x-y) \left|I(y)-f_1(x)\right|^2 H(\phi(y)) dy \right] dx \\ &+\lambda_2 \int \left[\int K_\sigma(x-y) \left|I(y)-f_2(x)\right|^2 (1-H(\phi(y)) dy \right] dx \end{aligned} \qquad (3)$$

where fixed parameters of λ_1 and λ_2 are positive constants, $I:\Omega \to R$ is a given vector valued image, K_σ is a Gaussian kernel of which standard deviation is represented by σ, and f_1 and f_2 are the smoothing functions that approximate the image intensity in the vicinity of every *x* point, which is the center point of integral in Equation (3). In this kernel, localization property results in larger values at the points *y* near point *x*, whereas leads to smaller values as *y* departs from *x*. Therefore, the image intensities at the points where *y* is close to the point *x* weighs in the values of f_1 and f_2 which are to minimize the Equation (3).

Functional E^{LBF} can be minimized with respect to level set ϕ, and the gradient descent is expressed as follows:

$$\frac{\partial \phi}{\partial t} = -\delta_\varepsilon(\phi)(\lambda_1 e_1 - \lambda_2 e_2) \tag{4}$$

In order to secure stability, a distance regularization term is added to Equation (4). Final form of functional is as follows:

$$\frac{\partial \phi}{\partial t} = \mu(\nabla^2\phi - div(\frac{\phi}{|\nabla\phi|})) + v\delta_\varepsilon(\phi)div(\frac{\nabla\phi}{|\nabla\phi|}) - \delta_\varepsilon(\phi)(\lambda_1 e_1 - \lambda_2 e_2) \tag{5}$$

Interested readers are referred to Li et al. [12] for further details about the solution for minimization of Equation (3). Parameters λ_1 and λ_2 control the weight of integrals over interior and exterior regions of the contour C. e_1 and e_2 are given in the Equation (6) as follows:

$$\begin{aligned} e_1(x) &= \int_\Omega K_\sigma(y-x)\,|\,I(x) - f_1(y)\,|^2\,dy \\ e_2(x) &= \int_\Omega K_\sigma(y-x)\,|\,I(x) - f_2(y)\,|^2\,dy \end{aligned} \tag{6}$$

Notably, the standard deviation σ has a significant role in scaling the region from initial neighborhood to the entire image and must be chosen optimally to avoid undesirable outputs and high computational cost.

Local Fitted Image. Zhang et al. improved the idea of Li et al. in terms of decreasing the computational cost by eliminating the convolution terms from being computed in each iteration [17]. Their local fitted image (LFI) is formulated as follows:

$$I^{LFI} = m_1 H_\varepsilon(\phi) + m_2(1 - H_\varepsilon(\phi)) \tag{7}$$

where constant of m_1 and m_2 are expressed as follows:

$$\begin{aligned} m_1 &= mean(I \in (x \in (\phi(x) < 0) \bigcap W_k(x))) \\ m_2 &= mean(I \in (x \in (\phi(x) > 0) \bigcap W_k(x))) \end{aligned} \tag{8}$$

where $w_k(x)$ is a rectangular window function which is to be convolved with a standard deviation σ and size of *4k+1* by *4k+1*, where *k* can be selected as the greatest integer less than σ. They proposed to minimize the difference between the original image and the fitted image by solving an energy functional in the frame LFI. Formal representation of this energy is as follows:

$$E^{LIF}(\phi) = \frac{1}{2}\int_{\Omega}(I_x - I^{LFI}(x))^2 dx \tag{9}$$

Energy in Equation (9) can be minimized by the steepest descent method for which the flow is expressed as follows:

$$\frac{\partial \phi}{\partial t} = (I - I^{LFI})(m_1 - m_2)\delta(\phi)) \tag{10}$$

where δ_ε the regularized Dirac function. Hence, the total level set formulation is defined as in the following:

$$\frac{\partial \phi}{\partial t} = \mu(\nabla^2\phi - div(\frac{\phi}{|\nabla\phi|})) + v\delta_\varepsilon(\phi)div(\frac{\nabla\phi}{|\nabla\phi|}) + (I - I^{LFI})(m_1 - m_2)\delta_\varepsilon(\phi)) \tag{11}$$

Notably, Equation (11) avoids the need for re-initialization of level set which decreases the computational cost as well. Due to massive numbers of obstacles on radar images, we reduced the input size of path planning task by zooming predefined location on a radar image.

2.2 Simulated images

Border detection. Border detection consists of two steps. First, we separate the black pixels from foreground and then delineate the contour that circumscribe the objects. Obtaining spatial information of the border contour is a significant step to create obstacles for path planning phase.

Thresholding. Thresholding is one of the basic approaches of image segmentation which generates binary images from gray scale images [19]. Basic image thresholding

is adopted to isolate object pixels from the background of original image. The philosophy of the image thresholding is to partition the image based on the intensity value of the pixels with respect to pre-determined threshold (*T*) such that if a pixel's intensity is less than *T*, thresholding substitutes the pixel with a black pixel.

Boundary tracing. We use a Matlab function, *bwboundary* that implements Moore-Neighbour tracing algorithm [20]. This algorithm is initiated at a black pixel toward any direction. Each time it hits a black pixel, it backtracks the white pixel that it previously departed from. This case repeats, if it meets a black pixel. The algorithm terminates when it revisits the starting pixel.

Border expansion. Border expansion consists of two steps. First one is dedicated to find outermost points on the object border. This can be achieved by finding the convex hull points of the border. We stretch the convex points in the desired amount. Significantly, we must note that border inflation is proportional to object size. In the last step, we need to have a continuous curve to circumscribe the objects so that safe path computations will be accurate. To this end, we fit a contour using *imresize* routine provided by Matlab. Notably, obstacle borders are expanded proportionally to their areas. We computed areas of obstacles, which are 2D contours, using *polyarea* routine provided by Matlab.

Determining outermost points of objects' borders. Border expansion is completed by computing the convex hull which is the smallest convex set that contains the points [21]. The algorithm is as follows [22]:

```
Algorithm QUICKHULL
QuickHull(a,b,S)
if isEmpty(S) then return()
else
  c ← indexOf(maxDistance from ab)
  A ← rightPoints(a,c)
  B ← rightPoints(c,b)
  return QuickHull(a,c,A)+(c)+QuickHull(c,b,B)
```

Interpolation for encapsulating new object borders. Bilinear interpolation is one of the basic resampling techniques and used for producing a reasonably realistic image. Image scaling is performed by moving the pixels in a certain direction based on the given scale parameter. However, some pixels may not be assigned to appropriate values during scale-up process if the scale factor is a non-integral. To handle this case, appropriate intensity or color values are assigned to these gaps so that output image does not have holes (non-valued pixels).

Bilinear interpolation takes average of the adjacent 2x2 window of known pixel values, which gives the interpolated value. The weight of each 4-pixel values is based on the Euclidean distance of pixels to the each of the known points. The algorithm development is given below.

Let us assume, we are after finding the value of a function *f* at the *(x, y)* point. Consider the value of *f* at the four points, $W_{11}=(x_1,y_1)$, $W_{12}=(x_1,y_2)$, $W_{21}=(x_2,y_1)$, $W_{22}=(x_2,y_2)$,

$$f(x_1)=\frac{x_2-x}{x_2-x_1}f(W_{11})+\frac{x-x_1}{x_2-x_1}f(W_{21}) \tag{12}$$

$$f(x,y_2)=\frac{x_2-x}{x_2-x_1}f(W_{12})+\frac{x-x_1}{x_2-x_1}f(W_{22}) \tag{13}$$

$$f(x,y)=\frac{y_2-y}{y_2-y_1}f(x,y_1)+\frac{y-y_1}{y_2-y_1}f(x,y_2) \tag{14}$$

$$\approx\frac{y_2-y}{y_2-y_1}\left(\frac{x_2-x}{x_2-x_1}f(W_{11})+\frac{x-x_1}{x_2-x_1}f(W_{21})\right)+\frac{y-y_1}{y_2-y_1}\left(\frac{x_2-x}{x_2-x_1}f(W_{12})+\frac{x-x_1}{x_2-x_1}f(W_{22})\right) \tag{15}$$

$$=\frac{1}{(x_2-x_1)(y_2-y_1)}\begin{bmatrix}x_2-x & x-x_1\end{bmatrix}\begin{matrix}f(W_{11}) & f(W_{12})\\ f(W_{21}) & f(W_{22})\end{matrix}\begin{bmatrix}y_2-y\\ y-y_2\end{bmatrix} \tag{16}$$

Shortest path algorithm. In this study, Dijkstra algorithm is employed for shortest path between two predefined points. The algorithm is given as follows [23-24]:

```
Algorithm DIJKSTRA (G,w,s)
d[s] = 0
for each v∈V-{s}
  do d[v] = ∞
S = ∅
Q = V
while Q ≠ ∅
  do u = extractMin(Q)
  S = S ∪ {u}
  for each v ∈ adj{u}
    do if d[v] > d[u] + w(u, v)
    then d[v] = d[u] + w(u, v)
```

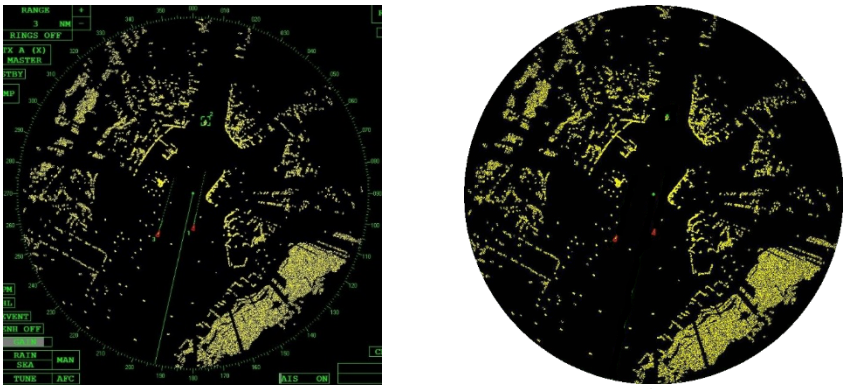
Smoothing process. Having characteristic maneuvers, the path we found is not realistic in terms of ship navigation. After obtaining the shortest and safest path, smoothing is conducted by using the conventional smooth routine in MATLAB.

3 Empirical Study

In this section, an empirical evaluation of the proposed approach is presented based on the radar images.

3.1 Radar Images

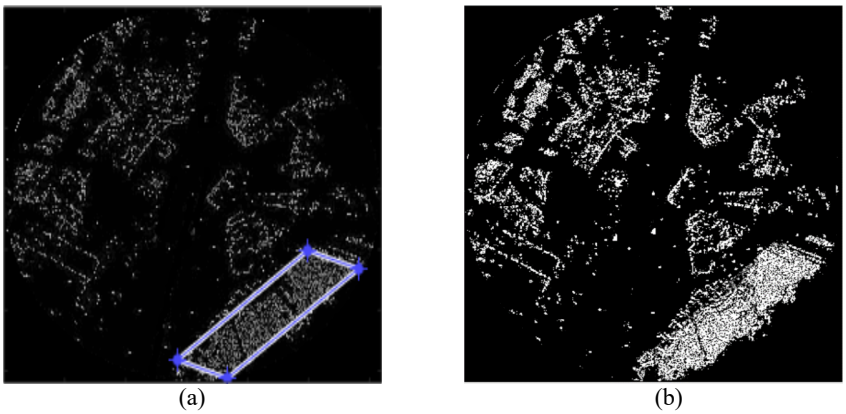
The input of the proposed system is a snapshot of a radar image where the ship is at the center as a dot. Generally, radar images (see Fig. 1 (a)) have annotations, and mark-ups on it which must be cleaned. We impainted the images manually; therefore, the input will be the image shown in Fig. 1 (b).



(a) (b)

Fig. 1. (a) Radar Screen (Courtesy of Buffalo Computer Graphics) and (b) manually impainted real radar image.

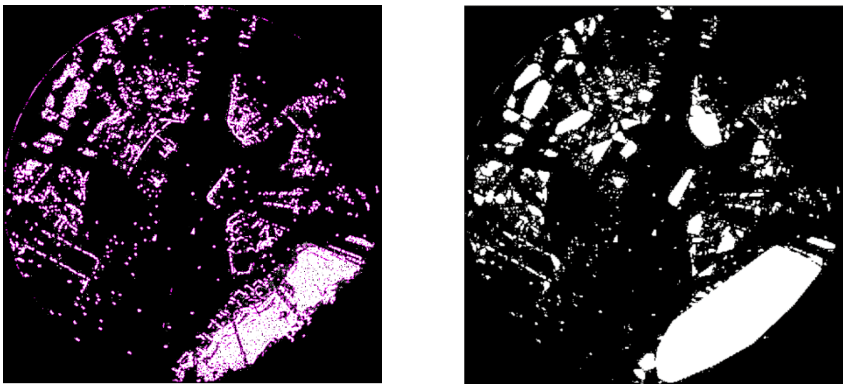
To perform the segmentation, an initial contour must be placed. This initial contour evolves the around the boundary of foreground objects as the result of image segmentation process as shown in Fig. 2 (a). Initial contour localization is not automatically done due to the 'region' related sensitivity of segmentation algorithm. In Fig. 2 (a), a region that has similar intensity with the desired output is selected as the initial contour. In Fig. 2 (b), output mask of the segmentation is shown.



(a) (b)

Fig. 2. (a) Initial contour is manually placed, and (b) segmented radar image.

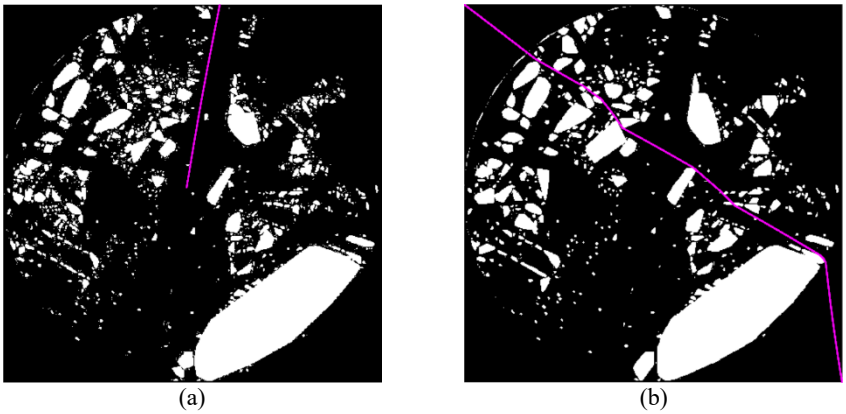
Note that, shortest path algorithm takes a binary matrix as an input, where ones represent the obstacles and zeros represent the background. Hence, the image shown in Fig. 2 (b), which is stored as a binary matrix in computer, can be passed to shortest path algorithm.



(a) (b)

Fig. 3. (a) All obstacles are segmented, and (b) merged radar image.

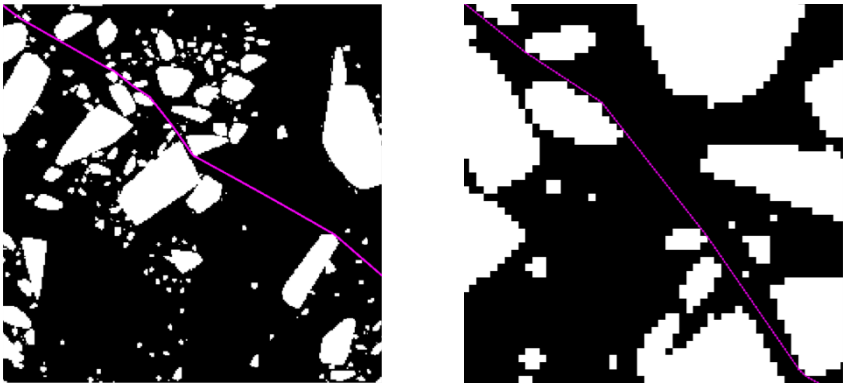
However, the vast number of segmented objects (see Fig. 2 (b) and Fig. 3(a)) severely increase the computational time to compute the shortest path yet may lead ship to penetrate through the group of objects. To avoid this case, we propose to merge the obstacles which are in high proximity to each other (see Fig. 3 (b)). Merging process is done using a MATLAB routine called *bwconvhull*. Here, the merged obstacles are shown, and the image is the input of shortest path algorithm.



(a) (b)

Fig. 4. (a) The shortest path where the ship's position is at the center of the image, and (b) the shortest path starting from top left to bottom right.

Fig. 4 (a) represents the shortest path from the center of the image to the desired destination. In order to show the applicability and generality, we use whole image as a region with obstacles as shown in Fig. 4 (b). It is assumed that the ship starts to navigate intended region from the top left point through the bottom right corner. The shortest and safest path is successfully found as in Fig. 4 (b).



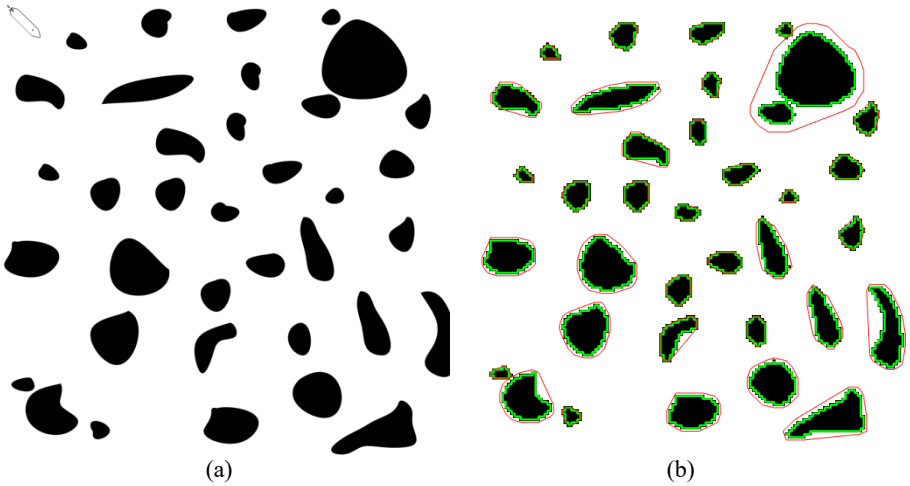
(c) (d)

Fig. 5. (a) Screenshots of the zoomed region, and (b) closer look to the zoomed region.

Nevertheless, the shortest path we found passes nearby obstacles as seen in Fig. 5 (a) and Fig. 5 (b), which is still not a desired case. To address this constraint, we propose to establish a 'safe zone' circumscribing each obstacle. This safe zone can be delineated by expanding the obstacle border. We tested that object boundary can expand from the outermost points (convex hull points) of the obstacle borders. We set the expansion amount directly proportional to obstacle area. Note that both for convex hull and polygon area calculation, we used MATLAB routines. After border expansion from convex points, we interpolated the new boundary using *imresize* to recapture the original shape of obstacles. Considering the large size of radar images and extensive computational cost for processing them, we reduced the number of obstacles by zooming a particular area of the radar image.

3.2 Simulated images

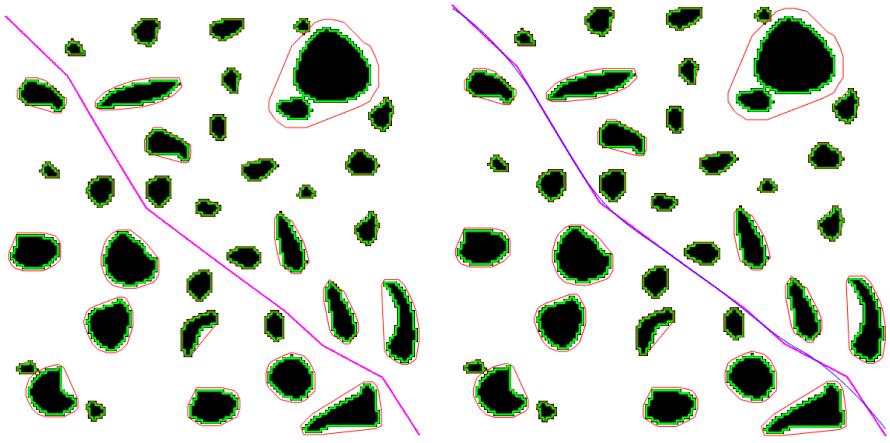
In the empirical study, the ship approaches a two-dimensional region with obstacles. The obstacles might be icebergs, islands, and also other ships. As it is seen in the Fig. 6 (a), the static obstacles are not polygonal and instead they might appear in any shape and size. The particulars of the ship are omitted and only the pivot point of the ship is considered as a navigating object. The ships must not touch the obstacles so as to navigate safely. Similarly, underwater of the icebergs are not observed visually, and it is mostly voluminous than the seen part. Therefore, the distance to the obstacles should be adjusted. In the Fig. 6 (b), the borders of the obstacles are detected. Besides any numerical assignments can be set for the safety distance of obstacles, we consider their sizes for the intended problem.



(a) (b)

Fig. 6. (a) The region that the vessel encounters, and (b) borders of obstacles are detected for safe navigation.

The shortest and safest path is found as given in Fig. 7 (a). To make it more realistic, the optimal route is smoothened (see Fig. 7 (b)).



(a) (b)

Fig. 7. (a) The shortest path that the vessel navigates safely, and (b) the shortest path with smoothing

4 Conclusions

Autonomous on-board management systems, including self-navigation, are highly important as they increase the speed and safety of the shipping operations. In this study,

image processing approaches are implemented for determining the shortest and safest path based on static radar images. After processing images, Dijsktra algorithm is employed for two types of images (i) manually impainted real radar images and (ii) simulated images. Following conclusions are reached: any obstacles (ships, ice blocks, islands, etc.) observed on the radar screen can be processed; the obstacles in any shape, size and color are easily segmented; safe zones are calculated and tested for simulated images successfully; safe zones can also be considered for manually impainted real radar images; and, safe zones are determined based on the obstacles' volume.

Yet the proposed approach currently has certain limitations hindering its practical applicability, such as the high computation time required to process the images and run the algorithm and the facts that the obstacles are static and iceberg volume of underwater and weather and atmospheric conditions are not taken into account. Particularly, the environment, targets, and obstacles of the sea change continuously and the proposed approach needs to take this dynamism into account. Nevertheless, the work presented demonstrates that processing of radar images has potential to aid ship navigation and should be explored as a research direction.

Future work includes processing image sequences to mimic real time video analysis. Our goal is to plug the computational core we designed here to the on-board ships to conduct real time surveillance. Secondly, we argue that parallelization of the shortest path algorithm will decrease the computational time considering the large size of radar images, i.e. 619x619. Real radar images rather than manually impainted real radar images should also be extracted for processing. Lastly, instead of using convex points of the contours around obstacles for offsetting the border outward, constant velocity level set propagation can be used as a more formal mathematical approach [25]. Embedding the proposed algorithm to the ship's radar systems should be studied. Shortest path should be found in a dynamic manner; therefore, new shortest path algorithms (i.e. deterministic, heuristic approaches) can be deployed.

References

1. Ari, I., Aksakalli, V., Aydogdu, V. and Kum, S. (2013). Optimal ship navigation with safety distance and realistic turn constraints. European Journal of Operational Research, 229(3), 707-717.
2. Austin, G. L., Bellon, A., Riley, M. and Ballantyne, E. (1985). Navigation by computer processing of marine radar images. Journal of Navigation, 38(03), 375-383.
3. Austin, G. L., Bellon, A. and Ballantyne, E. (1987). Sea trials of a navigation system based on computer processing of marine radar images. Journal of Navigation, 40(01), 73-80.
4. Kaya, S., Bayraktar, M., Kockara, S., Mete, M., Halic, T., Field, H. E., and Wong, H. K. (2016). Abrupt skin lesion border cutoff measurement for malignancy detection in dermoscopy images. BMC bioinformatics, 17(13), 367.
5. Bayraktar, M. (2017). Noise Sensitive Trajectory Planning for MR Guided TAVI. Functional imaging and modelling of the heart. 63
6. Bayraktar, M., Sahin, B., Yeniaras, E. and Iqbal, K. (2014). Applying an active contour model for pre-operative planning of transapical aortic valve replacement. In Workshop on Clinical Image-Based Procedures. Springer International Publishing. 151-158.

7. Bayraktar, M., Kaya, S., Yeniaras, E. and Iqbal, K. (2016). Trajectory Smoothing for Guiding Aortic Valve Delivery with Transapical Access. In Workshop on Clinical Image-Based Procedures. Springer International Publishing. 44-51.
8. Bayraktar, M. (2015). Image detection and compression for memory efficient system analysis. In Seventh International Conference on Machine Vision). International Society for Optics and Photonics.
9. Bayraktar, M. (2011). Image guided preoperative planning for aortic valve replacement (Doctoral dissertation, University of Houston.
10. Li, S., Lu, R., Zhang, L. and Peng, Y. (2013). Image processing algorithms for deep-space autonomous optical navigation. Journal of Navigation, 66(04), 605-623.
11. Cremers, D., Rousson, M., and Deriche, R. (2007). A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. International journal of computer vision, 72(2), 195-215.
12. Li, C., Kao, C-Y, Gore, J. C., and Ding, Z. (2007). Implicit Active Contours Driven by Local Binary Fitting Energy. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1-7.
13. Chan, T. and Vese, L. (1999). An active contour model without edges. In International Conference on Scale-Space Theories in Computer Vision. Springer Berlin Heidelberg, 141-151.
14. Vese, L. A. and Chan, T. F. (2002). A multiphase level set framework for image segmentation using the Mumford and Shah model. International journal of computer vision, 50(3), 271-293.
15. Duan, Y., Huang, W., Zhou, J., Chang, H. and Zeng, T. (2014). A Two-Stage Image Segmentation Method Using Euler's Elastica Regularized Mumford-Shah Model. In Pattern Recognition (ICPR), 2014 22nd International Conference on IEEE. 118-123.
16. Tsai A., Yezzi A. and Willsky, A. S. (2001). Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. IEEE transactions on Image Processing, 10(8), 1169-1186.
17. Zhang, K., Zhang, L., Song, H., and Zhang, D. (2013). Reinitialization-free level set evolution via reaction diffusion. IEEE Transactions on Image Processing, 22(1), 258-271.
18. Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. Communications on pure and applied mathematics, 42(5), 577-685.
19. Sezgin M. and Sankur B. (2004). Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging 13(1), 146–165.
20. Gonzalez, R. C., Woods, R. E. and Eddins, S. L. (2010). Digital Image Processing Using MATLAB®, McGraw Hill Education.
21. Barber, C. B., Dobkin, D. P. and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. ACM Transactions on Mathematical Software (TOMS), 22(4), 469-483.
22. o'Rourke, J. (1998). Computational geometry in C. Cambridge University Press.
23. Hart, C., (2013). Graph Theory Topics in Computer Networking. University of Houston-Downtown, Department Computer and Mathematical Sciences, Senior Project.
24. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik. 1, 269–271. doi:10.1007/BF01386390.
25. Kimmel, R. and Bruckstein, A. M. (1993). Shape offsets via level sets. Computer-Aided Design, 25(3), 154-162.