

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

---

CSE Technical reports

Computer Science and Engineering, Department  
of

---

2002

## Two Algorithms for Shaped-Based Porosity Measures

Leen-Kiat Soh

University of Nebraska, Isoh2@unl.edu

Follow this and additional works at: <https://digitalcommons.unl.edu/csetechreports>



Part of the [Computer Sciences Commons](#)

---

Soh, Leen-Kiat, "Two Algorithms for Shaped-Based Porosity Measures" (2002). *CSE Technical reports*. 99.  
<https://digitalcommons.unl.edu/csetechreports/99>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Technical reports by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

## Two Algorithms for Shaped-Based Porosity Measures

Leen-Kiat Soh

Computer Science and Engineering Department

University of Nebraska

Lincoln, NE 68588-0115

Tel: (402) 472-6738 Fax: (402) 472-7787 E-mail: lksoh@cse.unl.edu

**Abstract**—In this paper, we present two algorithms for shape-based porosity measures. Specifically, we describe two innovative attributes called perimeter porosity and area porosity as feature descriptors. Perimeter porosity refers to how “porous” or “weak” an image object is, while area porosity determines the degree of “branchiness” of an object. We show the applicability of the two measures through a SAR sea ice image classification example.

### I INTRODUCTION

In this paper, we propose two algorithms for shaped-based porosity measures to describe a 2-dimensional image object. Specifically, we define perimeter porosity and area porosity. Perimeter porosity is a measurement of the porosity or “structural weakness” (e.g., holes and 1-pixel thick protrusions) of an object. Area porosity is a measurement of the branchiness of an object. Branchiness is seen as the number of extended protrusions of an object. In the following, we will present the algorithms in Section 2 and then discuss the application of these two algorithms in an intelligent sea ice classification approach in Section 3. Then we conclude.

### II METHODOLOGY

Since we are dealing with shaped-based features, we need to identify and separate individual objects in an image. This is achieved via segmentation. In the following discussion, we assume that the image has been segmented to a set of objects and a background. Background pixels are labeled 0 while the pixels of the same object shares a unique label.

#### A. *Perimeter Porosity*

To compute for the perimeter porosity of an object, we first construct the chain code [1] of the object’s boundary. In this algorithm, we use a 4-neighbor chain code. It contains an initial point  $\langle x_{ini}, y_{ini} \rangle$ , an end point  $\langle x_{end}, y_{end} \rangle$ , and a set of 4-directions leading from the current point to the next on the boundary. Given an image object with its pixels labeled with a unique value  $o$ , the following algorithm computes the perimeter porosity:

#### ***Algorithm Perimeter\_Porosity***

##### 1. ***Algorithm Outer\_Perimeter***

- a. Find the topmost, leftmost pixel with the label  $o$ . This is the initial point of the chain code.
- b. Construct the chain code by moving to the leftmost 4-neighbor that shares the same label until the next neighbor has reaches the initial point of the chain code.
- c. The number of directions or moves of the chain code is the *outer\_perimeter*.

##### 2. ***Algorithm Perimeter***

- a. Scan from left to right, top to bottom: if a pixel with the label  $o$  has a non- $o$  8-neighbor, then add 1 to the *perimeter* count.

3. Compute:

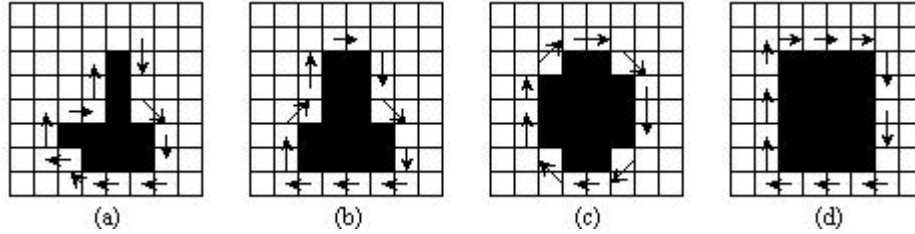
$$perimeter\_porosity = \frac{\max(outer\_perimeter, perimeter)}{\min(outer\_perimeter, perimeter)}$$

\*Note that if we use, instead, an 8-neighbor chain code, then we look for non- $o$  4-neighbors.

### End Algorithm

Note that the algorithm is based on two different ways of computing perimeters. The value of *perimeter* is the actual boundary pixels of the object. The value of *outer\_perimeter* is the number of moves one needs to take in order to traverse the entire boundary of the object—that may actually require moving to boundary pixels that have been traversed before.

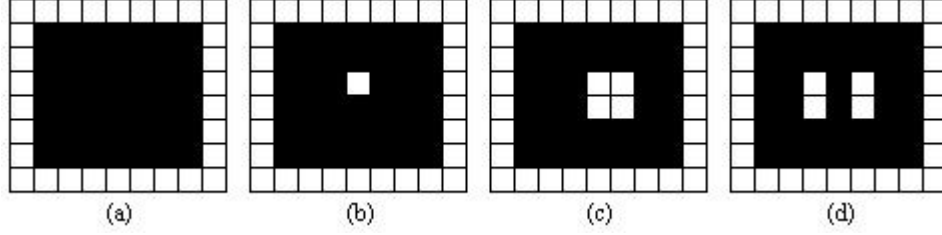
Figure 1 shows four examples related to structural weakness of an object: 1-pixel-thick protrusions. In the first example, the *outer\_perimeter* value is 11, following the 4-neighbor chain code, while the *perimeter* is only 10. Thus, the *perimeter\_porosity* is 14/10 or 1.4. In the second example, the *outer\_perimeter* is 14, while the *perimeter* is 14. Thus, the *perimeter\_porosity* is 14/14 or 1.0. In the third example, we can see that the *outer\_perimeter* is 14, the *perimeter* is 14, and thus the *perimeter\_porosity* is once again 1.0. Finally, for the last example, the *perimeter* and the *outer\_perimeter* are the same at 14, and the *perimeter\_porosity* is 1.0.



**Figure 1** (a) *outer\_perimeter* is larger than *perimeter*, (b) *perimeter* and *outer\_perimeter* are the same, (c) *perimeter* and *outer\_perimeter* are the same, and (d) *perimeter* and *outer\_perimeter* are the same.

On the other hand, if we had used an 8-neighbor chain code and looked for non- $o$  4-neighbors in *Algorithm Perimeter*, we would have: (a) *outer\_perimeter* = 11, *perimeter* = 9, *perimeter\_porosity* = 11/9 = 1.22, (b) *outer\_perimeter* = 12, *perimeter* = 12, *perimeter\_porosity* = 1.0, (c) *outer\_perimeter* = 10, *perimeter* = 10, *perimeter\_porosity* = 1.0, and (d) *outer\_perimeter* = 14, *perimeter* = 14, *perimeter\_porosity* = 1.0. So, in either design, Figure 1(a) would have a higher perimeter porosity value than all the others.

Figure 2 shows the examples related to the porosity of an object: holes within the object. In general, the size of a hole and how the holes are distributed in an object affect the perimeter porosity value. For example, a 1-pixel hole in Figure 2(b) yields a porosity of 1.36 while a 4-pixel hole in Figure 2(c) yields 1.55. Also, even though the total size of holes in Figure 2(c) and Figure 2(d) are the same—that is, 4 pixels—the perimeter porosity values are different. Figure 2(d) results in 1.73.



**Figure 2** (a)  $perimeter\_porosity = 22/22 = 1.0$ , (b)  $perimeter\_porosity = 30/22 = 1.36$ , (c)  $perimeter\_porosity = 34/22 = 1.55$ , and (d)  $perimeter\_porosity = 38/22 = 1.73$ .

Note that if we had used just the number of object pixels and the number of “hole” pixels as a porosity measure, Figure 2(c) and Figure 2(d) would have yielded the same porosity values. Thus, our design for perimeter porosity has a bias: an object with many small holes is in general more porous than an object with a single large hole. This corresponds intuitively to what we expect of a porosity measure.

### B. Area Porosity

The underlying principle for this measure is based on the best-fitting bounding rectangle of an object. The idea is that a *structurally compact* object should fill out its bounding rectangle nicely. One that does not is considered to have a high area porosity measure.

We compute for the bounding rectangle indirectly by computing for the size of its area, instead of the actual corner points of the rectangle. We first find the orientation of the object, to gauge how it aligns in the image, based on the object’s first- and second-order moments [2]:

#### Algorithm Orientation

- a. Compute:

$$orientation = \frac{1}{2} \tan^{-1} \left[ \frac{2\bar{\mu}_{1,1}}{\bar{\mu}_{2,0} - \bar{\mu}_{0,2}} \right]$$

where  $\bar{\mu}_{0,2} = \sum_{all\ y} (y - \mu_y)^2$ ,  $\bar{\mu}_{2,0} = \sum_{all\ x} (x - \mu_x)^2$ ,  $\bar{\mu}_{1,1} = \sum_{all\ x\ all\ y} (x - \mu_x)(y - \mu_y)$ ,  $x$  is the column value of a pixel located at  $\langle x, y \rangle$ ,  $y$  is the row value of a pixel located at  $\langle x, y \rangle$ ,  $\mu_x$  is the mean value of all values of  $x$ , and  $\mu_y$  is the mean value of all values of  $y$ .

#### End Algorithm

Given the orientation value, we can then identify the two principal axes as the maximum length and width of the object [2]:

#### Algorithm Maximum Length and Width

- a. For each pixel  $\langle x, y \rangle$ , compute its

$$\alpha = x \cos(orientation) + y \sin(orientation)$$

$$\beta = -x \sin(orientation) + y \cos(orientation)$$

- b. Identify the maximum  $\alpha$  as  $\alpha_{max}$ ; identify the maximum  $\beta$  as  $\beta_{max}$ ; identify the minimum  $\alpha$  as  $\alpha_{min}$ ; identify the minimum  $\beta$  as  $\beta_{min}$ .
- c. Compute the lengths of the principal axes:

$$d_\alpha = \alpha_{max} - \alpha_{min}, \text{ and}$$

$$d_{\beta} = \beta_{\max} - \beta_{\min}.$$

d. Compute

$$\max\_length = \max(d_{\alpha}, d_{\beta}), \text{ and}$$

$$\max\_width = \min(d_{\alpha}, d_{\beta}).$$

**End Algorithm**

Finally, the algorithm for computing for area porosity is:

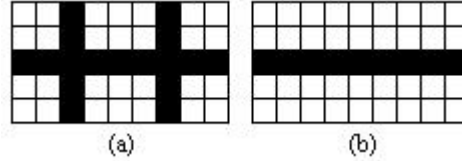
**Algorithm Area\_Porosity**

1. *Algorithm Orientation*
2. *Algorithm Maximum Length and Width.* This yields  $\max\_length$  and  $\max\_width$ .
3. Compute the size of the object by counting all the  $o$ -labeled pixels. This is  $area$ .
4. Compute:

$$area\_porosity = \frac{\max\_width \cdot \max\_length}{area}$$

**End Algorithm**

Figure 3 shows an example related to area porosity. In Figure 3(a), the size of the bounding rectangle is  $9 \times 5 = 45$ . The actual area of the object is 17 pixels. Thus, the area porosity of the object is  $45/17 = 2.65$ . In Figure 3(b), the size of the bounding rectangle is the same as the area of the object: 9. Thus, the area porosity of the object is 1.0.



**Figure 3** (a)  $area\_porosity = 45/17 = 2.65$ , and (b)  $area\_porosity = 9/9 = 1.0$ .

### III DISCUSSION

We have applied the above two measures to a sea ice classification system [3]. This expert system extracts sea ice objects from images and describes each object with a set of measurements. It then converts the measurements into discrete, symbolic facts through simple and composite thresholding. During the classification phase, the system matches these facts against a set of rules. Rules that match will be fired with assertions—each assertion is a classification of the object with a confidence value. The system combines all the assertions into a final, consistent classification for each object.

Here is how we apply the porosity measures to our classification task. First we define a complex measurement based on the two proposed measures:

$$irregularity = area\_porosity \cdot perimeter\_porosity.$$

That is, an object is highly irregular if it has a high area porosity measure and a high perimeter porosity measure.

Then, we use this composite measure to derive symbolic facts. First, we determine whether an object is an *ice lead*. To be considered as an ice lead, a feature must be elongated and irregular. Thus:

If  $elongation > T_{elongated}$  and  $irregularity > T_{irregular}$  then  $lead = true$   
 else  $lead = false$ .

Second, we determine whether an object is a *blob*. A blob is a very large, shapeless object. This could be due to (1) a large area of old ice covering most of the image, with holes within the area, or (2) an image artifact due to poor image quality that hinders effective extraction of individual objects. This is based on the measurements *area*, *irregularity*, and *eccentricity*. To obtain the fact, we have:

If  $area > T_{size,2}$  and (  $irregularity > T_{irregular}$  or  $eccentricity > T_{eccentric}$  ) then  
 $blob = true$  else  $blob = false$ .

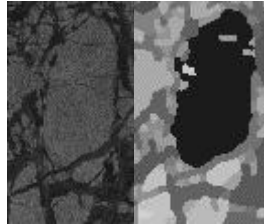
Table 1 documents a subset of the rules that we have used in our rule bases for sea ice classification. The delimiter within each rule is the semicolon. Each rule has an index, an English text description of what the rule does, the set of attribute-value pairs (separated by commas), the classification, and the confidence of that classification. So, for example, rule 157 says that if the image was taken in winter, and the object has a medium SSM/I concentration value and is irregular, then the object is classified as open water with confidence 0.8.

rule=4;If lead and return is dark then open_water;winter true,lead true,return dark;open_water;0.4
rule=5;If it is a dark lead and smooth then open_water;summer true,smooth true,return dark,lead true;open_water;0.6
rule=25;If lead and return is dark and mottled then first_year_ice;winter true,lead true,return dark;first_year_ice;0.6
rule=67;If it is lead and it is bright then it is not old_ice;winter true,lead true,return bright;old_ice;-0.9
rule=76;If its dark, darker than, smooth, and is really large, its probably not MY;winter true,blob true,return dark,brighter false,smooth true;old_ice;-0.8
rule=77;If its grey, darker than, smooth, and really large, its probably not MY either;winter true,blob true,return grey,smooth true,brighter false;old_ice;-0.6
rule=86;if not blob but irregular, thin, and darker than, it could be FY;winter true,blob false,irregular true,thin true,brighter false;first_year_ice;0.7
rule=88;if not blob but irregular, thin, dark, and darker than, it could be FY;winter true,blob false,irregular true,return dark,thin true,brighter false;first_year_ice;0.7
rule=89;if blob, mottled, enclose darker, and rougher then MY;winter true,blob true,mottled true,enclose darker,smoother false;old_ice;0.6
rule=90;if blob, mottled, enclose darker then MY;winter true,blob true,mottled true,enclose darker;old_ice;0.5
rule=92;if blob, mottled, enclose darker, rougher, brighter then MY;winter true,blob true,mottled true,enclose darker,smoother false,brighter true;old_ice;0.8

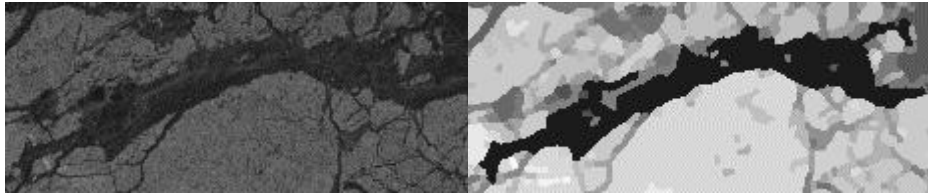
rule=157;if the SSM/I is med and the feature is irregular, its probably ni;winter true,ssmicon medium,irregular true;open\_water;0.8

**Table 1** A subset of rules involving the facts *blob*, *lead*, and *irregular* in our sea ice classification expert system. These three facts utilize the area porosity and perimeter porosity measures.

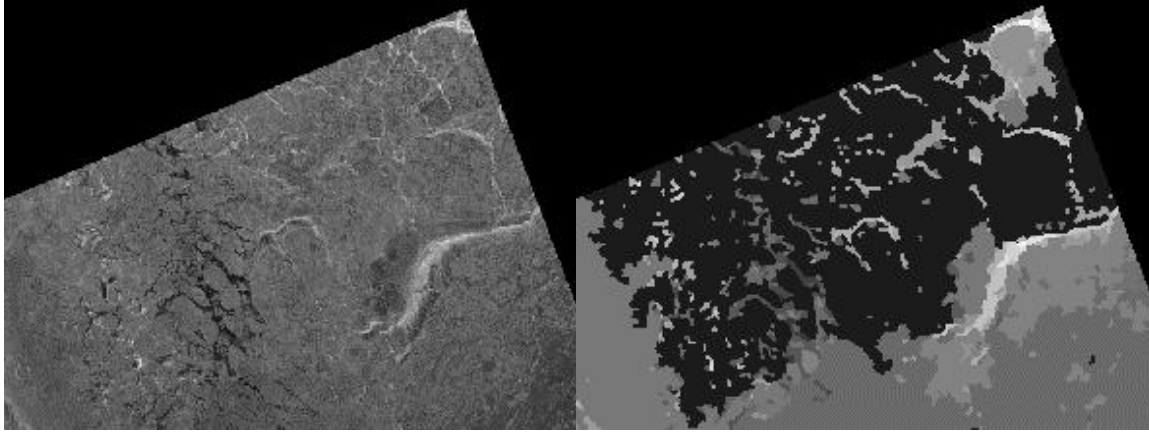
Figure 4 shows an example of an ice floe whose porosity measures are close to 1.0. This floe object was classified as old ice with a confidence level of 0.99. Figure 5 shows an example of an ice lead whose area porosity measure is much greater than 1.0, significantly contributing to its irregularity. This lead object was classified as first year ice with a confidence level of 0.66. Figure 6 shows an example of a blob whose perimeter porosity measure is above 2.0, significantly contributing to its irregularity. This blob was classified as old ice with a confidence level of 0.98. We see that due to the ability of the two shape-based porosity measures to capture the inherent “porosity” characteristics of the objects, they play a significant role here in helping classify these three image objects, given the rules such as those in Table 1.



**Figure 4** (a) A portion of an original image. Copyright ESA. (b) The segmented image with object #1535 colored blue. Measurements: area = 2321, perimeter = 251, outer\_perimeter = 208, orientation = 0.1974, maximum length = 81.21, maximum width = 42.95, area\_porosity = 1.54, perimeter\_porosity = 1.21, irregularity = 1.86.



**Figure 5** (a) A portion of an original image. Copyright ESA. (b) The segmented image with object #2264 colored blue. Measurements: area = 4012, perimeter = 724, outer\_perimeter = 644, orientation = -0.2253, maximum length = 226.20, maximum width = 57.27, area\_porosity = 3.23, perimeter\_porosity = 1.13, irregularity = 3.63.



**Figure 6** (a) A portion of an original image. Copyright ESA. (b) The segmented image with object #22 colored blue. Measurements: area = 72025, perimeter = 7213, outer\_perimeter = 3577, orientation = -0.3722, maximum length = 453.86, maximum width = 255.37, area\_porosity = 1.61, perimeter\_porosity = 2.02, irregularity = 3.25.

#### IV CONCLUSION

We have presented two algorithms for shape-based porosity measures: (1) perimeter porosity that estimates the “structural weakness” such as holes and 1-pixel-thick structures, making use of a chain code and a simple count of boundary pixels, and (2) area porosity that estimates the “branchiness” of an object based on its best-fitting bounding rectangle. We have shown through an example application in SAR sea ice image classification how these two measures can be used to capture “porosity” characteristics of sea ice objects such as leads, floes, blobs, and irregular objects. We consider these two porosity measures applicable to other image processing applications such as sandbar detection in a river, fracture analysis in concrete, and as general density measurements.

#### V REFERENCES

- [1] A. Bovik, *Handbook of Image and Video Processing*, San Diego, CA: Academic Press, 2000.
- [2] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood, CA: Prentice Hall, 1989.
- [3] L.-K. Soh and C. Tsatsoulis, “ARKTOS: A Knowledge Engineering Software Package for Satellite Sea Ice,” in *Proceedings of the 2000 International Geoscience and Remote Sensing Symposium (IGARSS'00)*, Honolulu, Hawaii, pp. 696-698, 2000.