Ice Floe Identification in Satellite Images Using Mathematical Morphology and Clustering About Principal Curves

Jeffrey D. Banfield; Adrian E. Raftery

Journal of the American Statistical Association, Vol. 87, No. 417 (Mar., 1992), 7-16.

Stable URL:
http://links.jstor.org/sici?sici=0162-1459%28199203%2987%3A417%3C7%3AIFIISI%3E2.0.CO%3B2-W

Ice Floe Identification in Satellite Images Using Mathematical Morphology and Clustering About Principal Curves

JEFFREY D. BANFIELD and ADRIAN E. RAFTERY*

Identification of ice floes and their outlines in satellite images is important for understanding physical processes in the polar regions, for transportation in ice-covered seas, and for the design of offshore structures intended to survive in the presence of ice. At present this is done manually, a long and tedious process that precludes full use of the great volume of relevant images now available. We describe an accurate and almost fully automatic method for identifying ice floes and their outlines. Floe outlines are modeled as closed principal curves, a flexible class of smooth nonparametric curves. We propose a robust method of estimating closed principal curves that reduces both bias and variance. Initial estimates of floe outlines come from the erosion-propagation (EP) algorithm, which combines erosion from mathematical morphology with local propagation of information about floe edges. The edge pixels from the EP algorithm are grouped into floe outlines using a new clustering algorithm. This extends existing clustering methods by allowing groups to be centered about principal curves rather than points or lines. This may open the way to efficient feature extraction using cluster analysis in images more generally. The method is implemented in an object-oriented programming environment, for which it is well suited, and is quite computationally efficient.

KEY WORDS: Erosion; Feature extraction; Nonparametric curves; Object-oriented programming; Remote sensing; Robustness.

Knowledge of the shapes, sizes and spatial distribution of ice floes is important for understanding the physical processes operating on the ice pack in the polar regions. It is also important for practical problems associated with transportation in ice-covered seas and for the design of offshore structures intended to survive in the presence of ice.

Such information can be found in satellite images of the polar regions such as Figure 1, which exist in large and rapidly increasing numbers. Practical use of such images requires identification of the outlines of ice floes above a certain size. To date this has been done manually (Rothrock and Thorndike 1984), a slow and tedious process that often takes a day or more to record the data from a single image and effectively precludes full use of the data. Automating the process is inherently difficult. Problems include the presence of many smaller floes and of melt ponds on the surface of floes, which ensure that floes often do not appear as homogeneous blocks of ice in the image.

In this article we describe an almost fully automatic method for identifying the outlines of ice floes. The outcome of this is shown in Figure 2, and is virtually the same as the result of very careful manual digitization. We model ice floe outlines as closed principal curves (Hastie and Stuetzle 1989—hereafter HS), a flexible family of one-dimensional nonparametric curves in a higher dimensional space. Our method consists of identifying a set of edge pixels and grouping them into clusters about a principal curve. Each cluster corresponds to a floe, and the corresponding principal curve is the estimated floe outline.

* Jeffrey D. Banfield is Assistant Professor, Department of Mathematical Sciences, Montana State University, Bozeman, MT 59717. Adrian E. Raftery is Professor, Department of Statistics, University of Washington, Seattle, WA 98195. Jeffrey D. Banfield's research was supported in part by the Office of Naval Research under Contract N-00014-89-J-1114. Adrian E. Raftery's research was supported in part by the Office of Naval Research under Contract N-00014-88-K-0265. The authors are grateful to David A. Rothrock for providing the data on which this work was based and for many helpful discussions, and to John A. McDonald, Fionn Murtagh, Brian D. Ripley, Paul D. Sampson, Werner Stuetzle, the editor, the associate editor, and two anonymous referees for helpful comments and discussions.

The method involves several new statistical techniques:

1. A way of estimating closed principal curves that reduces both bias and variance and is robust to outliers. Here outliers take the form of melt ponds on the surface of ice floes (Section 1).
2. The erosion-propagation (EP) algorithm provides initial estimates of floe outlines. This combines the existing idea of erosion from mathematical morphology (Matheron 1975; Serra 1982) with that of local propagation of information about floe boundaries (Section 2).
3. A method for clustering about principal curves. It is traditional in clustering algorithms to separate data into groups, each of which is clustered about some central point (Gordon 1981, 1987; Murtagh 1985; Committee on Applied and Theoretical Statistics 1989). Here we generalize this to allow each group to be clustered about a different principal curve. This opens the possibility that cluster analysis may be useful more generally for fast feature extraction in images (Section 3).

The method is implemented in an object-oriented programming environment for which it is well suited and which seems computationally efficient.

1. ESTIMATING CLOSED PRINCIPAL CURVES

In this section we first review the definition and basic properties of principal curves (Section 1.1). We then describe a new algorithm for estimating closed principal curves that reduces both bias and variance and is robust (Section 1.2).

1.1 Principal Curves

A principal curve is a smooth, one-dimensional curve that passes through the middle of an m-dimensional data set. It is nonparametric, and its shape is suggested by the data; it

Journal of the American Statistical Association
March 1992, Vol. 87, No. 417, Applications and Case Studies

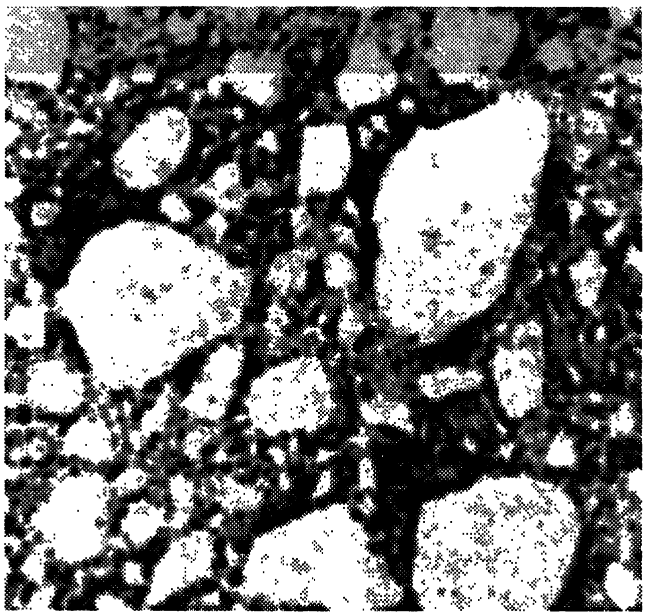


Figure 1. A Polar LANDSAT Image Showing Ice Floes. This is a 200 × 200 pixel image, where each pixel is 80m square; it thus represents a 15 × 15 km area.

thus provides a nonlinear summary of the data. The idea was introduced and developed by Hastie (1984), Hastie and Stuetzle (1985), and HS.

A one-dimensional curve in m-space is an m-vector consisting of m functions of a single variable λ, called coordinate functions. The variable λ parameterizes the curve and provides an ordering along it; λ will often be arc-length along the curve. Let $\mathbf{X} \in \mathbf{R}^m$ be a continuous random vector. Then $\mathbf{f}(\lambda)$ is a *principal curve* of $\mathbf{X}$ if

$$E[\mathbf{X} \mid \mathbf{f}^{-1}(\mathbf{X}) = \lambda] = \mathbf{f}(\lambda),$$

where

$$\mathbf{f}^{-1}(\mathbf{x}) = \sup_{\lambda}\{\lambda : \|\mathbf{x} - \mathbf{f}(\lambda)\| = \inf_{\mu} \|\mathbf{x} - \mathbf{f}(\mu)\|\}.$$

The quantity $\mathbf{f}^{-1}(\mathbf{x})$ is the value of λ for which $\mathbf{f}(\lambda)$ is closest to $\mathbf{x}$. Given the distribution of $\mathbf{X}$, HS proposed the following iterative algorithm for finding $\mathbf{f}$:

$$\mathbf{f}_{i+1}(\lambda) = E[\mathbf{X} \mid \mathbf{f}_i^{-1}(\mathbf{X}) = \lambda], \tag{1.1}$$

where $\mathbf{f}_i$ is the ith iterate.

When the distribution of $\mathbf{X}$ is unknown, this extends to estimation of $\mathbf{f}$ from a data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ by estimating $E[\mathbf{X} \mid \mathbf{f}_i^{-1}(\mathbf{X}) = \lambda]$. HS did this by means of scatterplot smoothing, using neighborhoods of each point defined by their projections onto the current estimate of the principal curve, rather than by their position in $\mathbf{R}^m$.

Let $\hat{\mathbf{f}}_i$ be the ith iterate and let $\tilde{\lambda}_j^i = \hat{\mathbf{f}}_i^{-1}(\mathbf{x}_j)$ $(j = 1, \ldots, n)$. Let $\lambda_{(j)}^i$ be the jth order statistic of the set $\{\tilde{\lambda}_1^i, \ldots, \tilde{\lambda}_n^i\}$, and let $x_{(j)}^i$ be the data point that projects onto $\lambda_{(j)}^i$. Then let $N_{(j)}^i$ be the set of data points $x_{(l)}^i$ such that $\lambda_{(l)}^i$ is in a neighborhood of $\lambda_{(j)}^i$; the size of the neighborhood is controlled by the *span*, equal to the fraction of all the data points that are in the neighborhood. The next iterate is then

$$\mathbf{f}_{i+1}(\lambda) = \hat{E}[\mathbf{X} \mid N_{(j)}^i]. \tag{1.2}$$

This is calculated using a coordinatewise scatterplot smoother, and various possibilities are discussed by HS.

There is no formal proof that the algorithm converges, but HS report that they have had no convergence problems with more than 40 real and simulated examples. Figure 3 shows the result of applying the HS estimation procedure to a simulated data set.

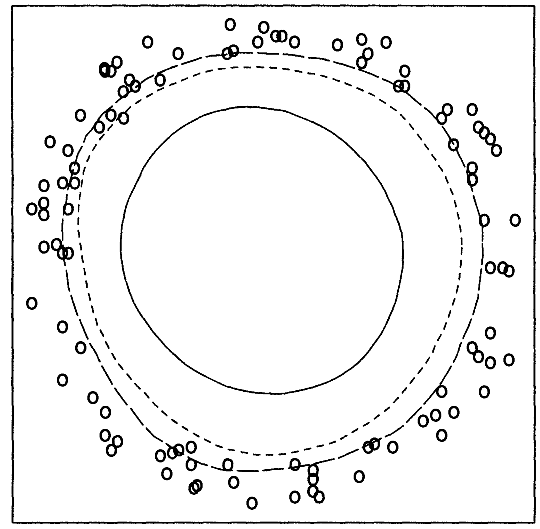


Figure 3. Estimated Principal Curves for Simulated Data. The data were obtained by generating points uniformly on the circumference of a circle and perturbing them randomly along the normal to the circle according to a Gaussian distribution. The principal curves were estimated using the algorithm (1.2) of HS with spans of .2 (outer dashed line), .3 (inner dashed line), and .5 (solid line).

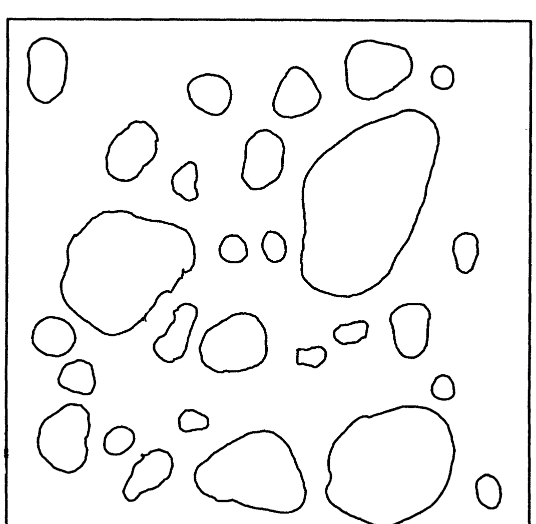


Figure 2. The Ice Floe Outlines, Larger Than a Fixed Minimum Size, Found by Our Procedure for the Data in Figure 1.

1.2 An Algorithm for Estimating Closed Principal Curves that Reduces Both Bias and Variance and Is Robust

Principal curve estimation is an iterative procedure, each iteration consisting of two steps. In the first step the data are ordered according to their projection on $\hat{\mathbf{f}}_i$, the current estimate of the principal curve. The ordering defines neighborhoods, $N_{(j)}^i$, that are used in the coordinatewise smoothing of the second step to provide the next estimate, $\hat{\mathbf{f}}_{i+1}$, of the principal curve as given in Equation (1.2).

Scatterplot smoothers generally produce curves that are biased toward the center of curvature. It is clear from Figure 3 that the estimated principal curve suffers from this problem, and that the bias increases with the span. In most statistical problems there is a trade-off between bias on one hand and smoothness and variance on the other. For closed curves, the center of curvature is interior to the curve (except for small local regions in nonconvex curves), which emphasizes the bias. We propose a modification of the HS algorithm that, for a fixed level of smoothness, reduces the bias. It should be noted that the HS algorithm was not designed for estimating closed curves.

Equation (1.1) can be rewritten as

$$\mathbf{f}_{i+1}(\lambda) = \mathbf{f}_i(\lambda) + \boldsymbol{\delta}^i(\mathbf{X}, \lambda), \tag{1.3}$$

where

$$\boldsymbol{\delta}^i(\mathbf{X}, \lambda) = E[\mathbf{X} - \mathbf{f}_i(\lambda) \mid \mathbf{f}_i^{-1}(\mathbf{X}) = \lambda].$$

We can think of $\boldsymbol{\delta}^i(\mathbf{X}, \lambda)$ as a measure of the bias at $\mathbf{f}_i(\lambda)$. We define $\mathbf{p}^i_{(j)} = \mathbf{x}^i_{(j)} - \hat{\mathbf{f}}_i(\lambda^i_{(j)})$ to be the projection residual of $\mathbf{x}^i_{(j)}$ projected onto $\hat{\mathbf{f}}_i$. The bias measure $\boldsymbol{\delta}^i(\mathbf{X}, \lambda)$ is the expected value of the projection residuals of the $\mathbf{x}$'s that project onto $\mathbf{f}_i$ at $\mathbf{f}_i(\lambda)$. This suggests that, when the distribution is unknown and is estimated from the data using an iterative algorithm such as (1.2), the projection residuals of the data in $N^i_{(j)}$, rather than the data themselves, should be used to calculate $\hat{\mathbf{f}}_{i+1}(\lambda)$. If we define $\bar{\mathbf{p}}^i_{(j)}$ as the coordinatewise average of the projection residuals of the data in $N^i_{(j)}$, we can use Equation (1.3) to estimate $\hat{\mathbf{f}}_{i+1}(\lambda)$ according to

$$\hat{\mathbf{f}}_{i+1}(\lambda^i_{(j)}) = \hat{\mathbf{f}}_i(\lambda^i_{(j)}) + \bar{\mathbf{p}}^i_{(j)}.$$

This may be regarded as a two-dimensional extension of the twicing procedure of Tukey (1977) for the one-dimensional case, which smooths the residuals from a nonparametric regression curve and then adds the smoothed residuals to the estimated curve. Here, however, we add the smoothed *projected* residuals to the curve.

As with the HS algorithm, there is no formal proof that our modification of it converges. Nonetheless, in the course of the present work we have used it to estimate over 50 different closed principal curves, and in each case our algorithm has converged in a satisfactory way.

Figure 4 shows the results of applying this algorithm to the same data and with the same spans as used for the HS algorithm in Figure 3. Our algorithm does not suffer from the bias problem inherent in that of HS. Moreover, for closed curves, the performance of our method is relatively insensitive to the precise choice of span; this is not true for the HS method. The principal curve estimated by our method with a span of .5 is smoother than that estimated with a span of .2, but the difference between them is barely noticeable. Figure 5 shows that our algorithm produces a much smoother curve when the span is small enough for that of HS to yield a relatively unbiased curve.

Outliers arise in the form of shallow, but sometimes large, melt ponds on the surface of the ice floe. An example of this is shown in Figure 6, which shows the edge pixels of

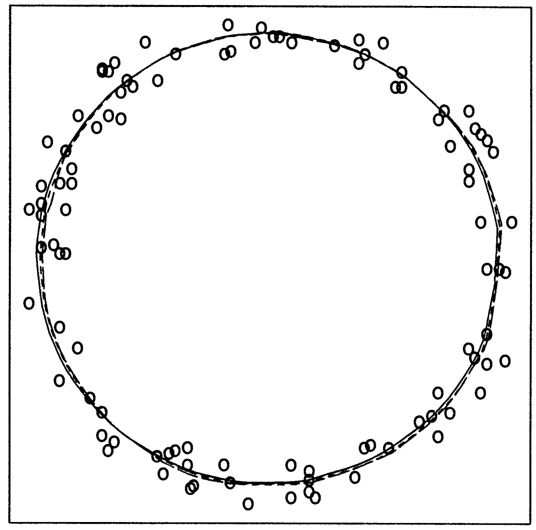


Figure 4. Estimated Principal Curves Using the Unbiased Algorithm Proposed in This Article. This shows the principal curves resulting from our unbiased estimation algorithm on the same data and at the same spans (.2, .3, and .5) used for the HS algorithm in Figure 3. The three curves almost totally overlay each other.

one floe in Figure 1 identified by the EP algorithm. The points near the left side interior to the floe are from melt ponds. Since they do not belong to the edge of the floe, we need to ensure that they do not affect the estimate of the principal curve.

To eliminate the effect of outliers, we use a slight modification of an approach suggested by HS. In the scatterplot smoothers we use a weighted average, where the weight of a point depends on its distance from the current estimate of the principal curve. We calculate the standard deviation of the lengths of the projection residuals and set the weight for a point to zero if it is more than three standard deviations from the current estimate of the principal curve, and

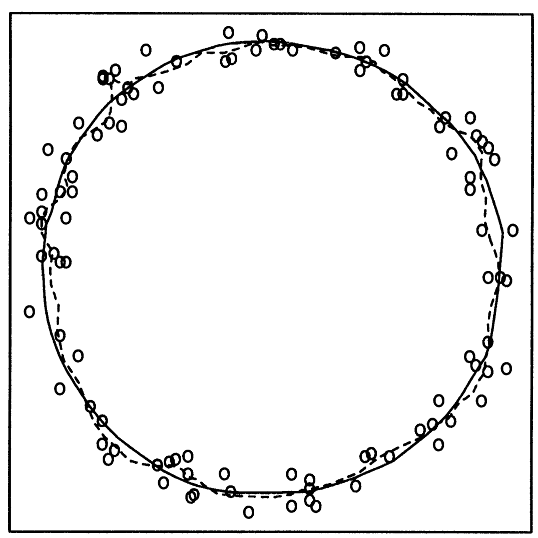


Figure 5. The Principal Curve From the HS Algorithm (dashed line) at a Span Small Enough to Eliminate Most of the Bias, Compared With the Estimate Proposed Here (solid line) With a Span of .2. The smaller the span, the less the bias and the rougher the estimated curve. Notice how rough the HS curve is, compared with our estimate.

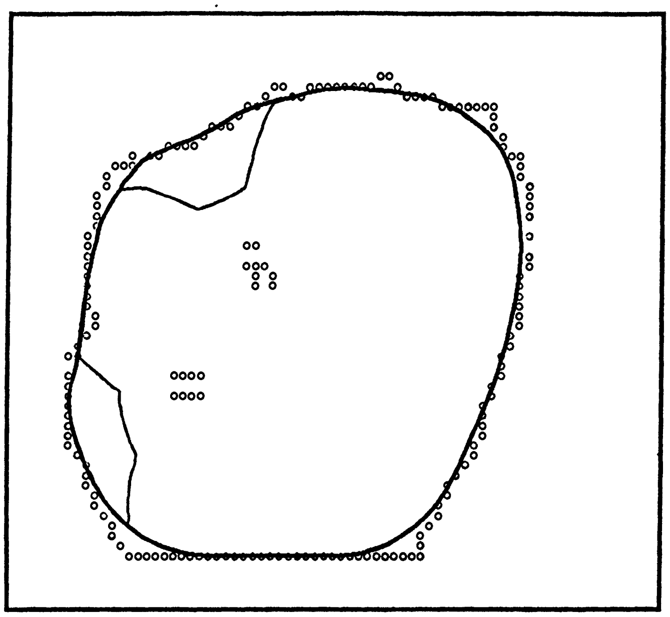


Figure 6. The Small Circles Are the Edge Pixels for One of the Floes in Figure 1, as Identified by the EP Algorithm. The points interior to the floe are from melt ponds. The lines show a principal curve estimated using the robust procedure described in the text (thick line), compared with the estimate from the nonrobust procedure (thin line). The robust estimate is unaffected by the melt ponds, while the nonrobust estimate is pulled toward them.

to unity otherwise. Figure 6 shows the result of this robust procedure as well as that of the nonrobust procedure, which uses the mean of all the data in each neighborhood. The robust procedure has clearly achieved its goal. It could be improved further by using a robust measure of scale, such as 1.25 times the mean absolute deviation, in place of the standard deviation. For our data, however, such a refinement makes no difference to our final result.

2. THE EROSION–PROPAGATION (EP) ALGORITHM

To select the potential edge pixels and provide an initial grouping of them into floe outlines, we use the EP algorithm. This operates on binary images. Images of ice floes, however, such as Figure 1, are usually greyscale. The marginal distribution of pixel intensities is highly bimodal, and so we work with the simpler binary image obtained by thresholding the original image; see Figure 7. The final result is relatively insensitive to the precise choice of threshold.

The erosion part of the EP algorithm, which identifies the potential edge elements, is a standard application of ideas in mathematical morphology (Serra 1982). The propagation part of the EP algorithm keeps track of the floe to which an edge pixel belongs by locally propagating the information about edge elements into the interior of the floe as it is eroded. This is facilitated by the object-oriented programming environment.

The algorithm is iterative and operates on a binary image consisting of figures (ice floes) on a contrasting background (water). At the first iteration, if a pixel is ice and a specified subset of its neighbors is water, the pixel is "melted" and becomes water, so that the figure to which it belongs is eroded. In our implementation, a pixel is melted if any of the eight neighboring pixels is water. At the second iteration, the same operation is performed on the image resulting from the first iteration, and so on. This can be formally described in terms of structuring elements using the terminology of mathematical morphology (Serra 1982; Banfield 1988). Any edge locating operator can be used to provide the initial set of potential edge pixels. We use the pixels melted at the first iteration of the EP algorithm.

Some results are shown in Figure 7. We can control the minimum size of the floes by waiting until a specified number of iterations, $i_{\min}$, have passed before recording a floe. The smallest floe that can be recorded is then a square of side $(2i_{\min} + 1)$ pixels. Smaller floes melt and are not recorded.

The idea of the propagation part of the EP algorithm is that the locations of the edge pixels are propagated toward the interior of the figure as it is eroded. At the end of the process, a single interior point of the figure will "know" the locations of all the edge pixels to which it corresponds. The location information is passed to only a few pixels that are taken as far from the eroded pixel as possible, subject to them not belonging to a different floe. This ensures that the amount of location information to be processed does not become unmanageable. It also prevents loss of information due to irregularity of the floe, melt ponds, or pixel misclassification at the thresholding stage.

The key to the propagation part of the EP algorithm is that it is never necessary for any pixel to know the direction of the interior of the floe. If a pixel is eroded at the ith iteration, then it was the center of a $(2i + 1)$ by $(2i + 1)$ square of ice pixels before the start of the erosion process. Thus its location may be passed anywhere within the square of side $(2i + 1)$ pixels surrounding it without any risk of the information being passed to another floe. We have found that it is enough to pass the location information in two opposite directions within the square. All of the eroded pixels are processed in exactly the same manner, and it is this uniform processing that allows the algorithm to be implemented on parallel processing machines.

In Figure 8 we show the results of the EP algorithm applied to the data in Figure 1, with a minimum floe size of 15×15 pixels (i.e., 1.2 km. square). The results are reasonably good: Of the 35 floes identified by the EP algorithm, 23 are "right" in the sense of being close to floes identified by careful manual digitization.

However, the EP algorithm tends to subdivide floes. This can occur when the floes are nonconvex or when they have melt ponds or noise in the interior. Figure 9 shows an example of the nonconvex case. As the floe shown in Figure 9 was eroded, the narrow middle section was pinched in and the floe was divided into two partial floes. Figure 10 is an example of a floe with melt ponds that cause the EP algorithm to produce three partial floes.

3. CLUSTERING ABOUT CLOSED PRINCIPAL CURVES

The EP algorithm tends to subdivide floes. We have, therefore, developed a method for determining which of the floes identified by the EP algorithm should be merged, based

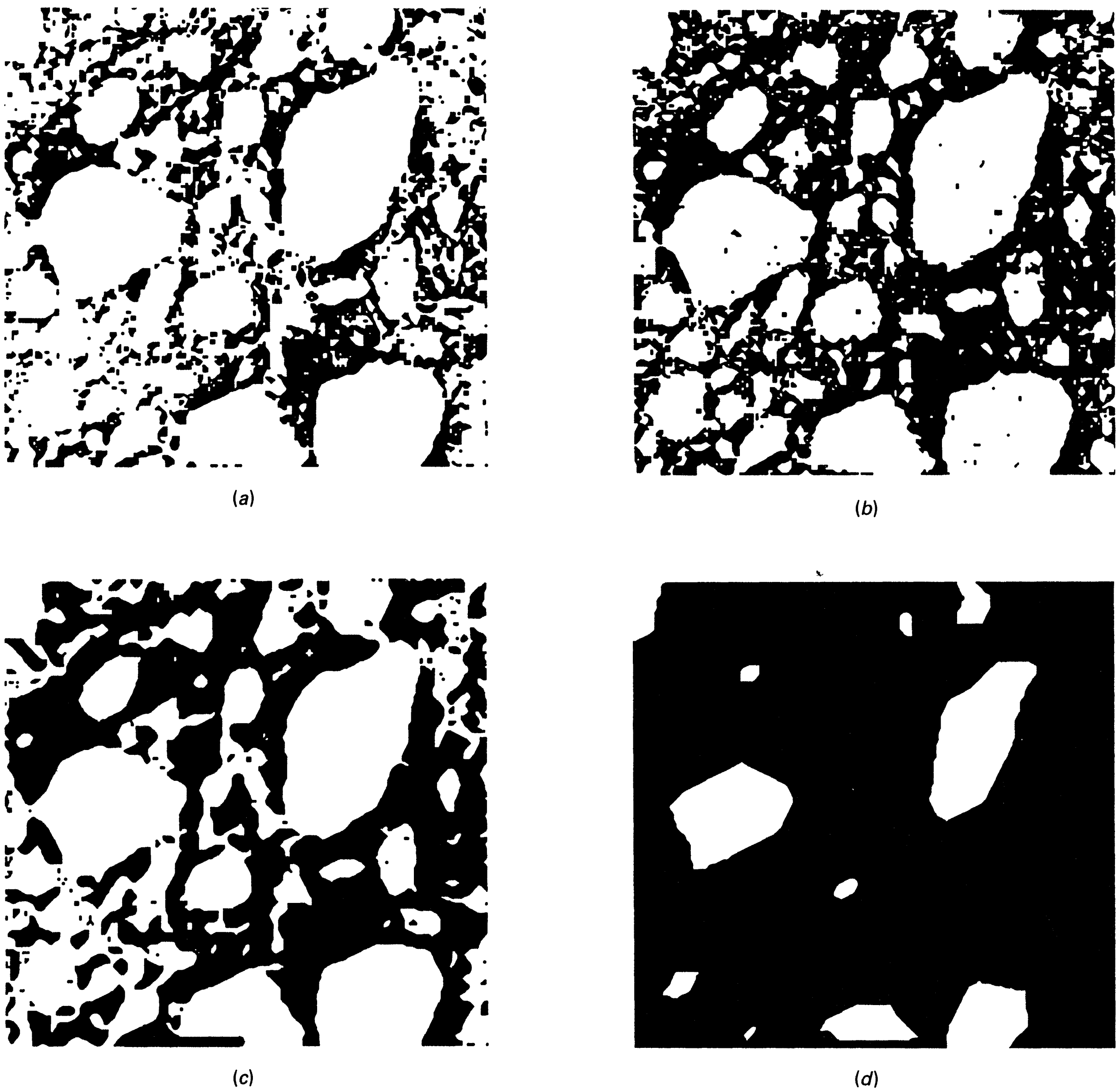


Figure 7. Binary Version of Figure 1 for Two Threshold Levels. The results are similar. However, (a) has a lower threshold level than (b) and, therefore, has more clutter in the water but less noise interior to the floes. The results of applying the EP algorithm to (a) is shown in (c) after three iterations and in (d) after 12 iterations.

on an algorithm for clustering about closed principal curves. Since we want to find out whether to merge tentatively identified floes, this is hierarchical and agglomerative.

The objective of cluster analysis is to group a set of observations into "interesting" subsets. In practice this has usually meant grouping observations that are close to one another. Ward (1963) proposed a hierarchical agglomerative algorithm for dividing data into g groups such that the sum of the within-group sums of squares is minimized. The algorithm starts by assigning each observation to a separate group. At each agglomeration two groups are merged, chosen so as to minimize the increase in the sum of within-group sums of squares. This clustering criterion is optimal if the data are generated by a finite mixture of spherical normal distributions. This corresponds to clusters that tend to be of the same size and spherical.

In cluster analysis, one of the major problems is deciding when to stop clustering. This is not a problem in our approach. The EP algorithm can subdivide floes, but potential partial floes can be identified by the fact that they will share edge elements. Complete floes that are touching can also share edge elements, so this fact alone does not provide an adequate criterion for determining which floes should be merged. We develop a clustering criterion based on a weighted decomposition of variance, V^*, that can distinguish between partial and complete floes. In general there are very few objects within each set of partial floes, so it is possible to evaluate all possible mergers and choose the result that minimizes $\Sigma\ V^*$. Since we are finding the partition of each set of partial floes that minimizes $\Sigma\ V^*$, the question of when to stop clustering does not arise.

Murtagh and Raftery (1984) proposed decomposing the

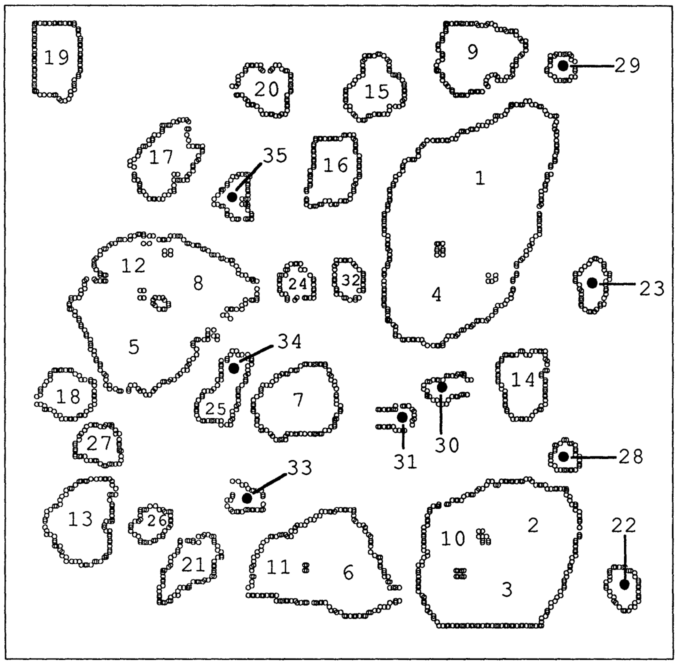


Figure 8. Result of the EP Algorithm Applied to the Data in Figure 1. Floes are not recorded unless they have survived at least 7 iterations. This corresponds to a minimum floe size of 15 × 15 pixels, or 1.2 km square. The open circles are the edge elements identified by the EP algorithm. The numbers (or solid dots) interior to each floe are the centers found by the EP algorithm. Note that centers 1 and 4 are on the same floe, which was subdivided because of the melt ponds. Other floes were also subdivided.

within-group sum of squares into parts and using a weighted sum of the parts as the clustering criterion, with weights chosen so as to emphasize aspects of interest in the application. For two-dimensional data, they suggested decomposing the within-group sum of squares into parts parallel and perpendicular to the first principal component of the group and downweighting the parallel part. This criterion

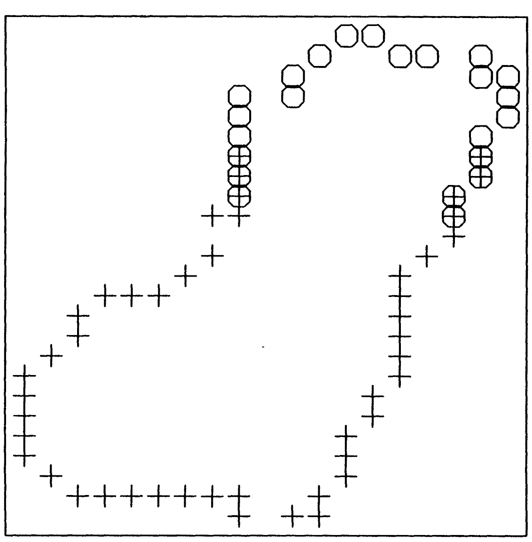


Figure 9. This Figure Shows How, When a Nonconvex Floe is Eroded, the Narrow Region Can be "Pinched Off," Resulting in Two Partial Floes (indicated by a + and a ○).

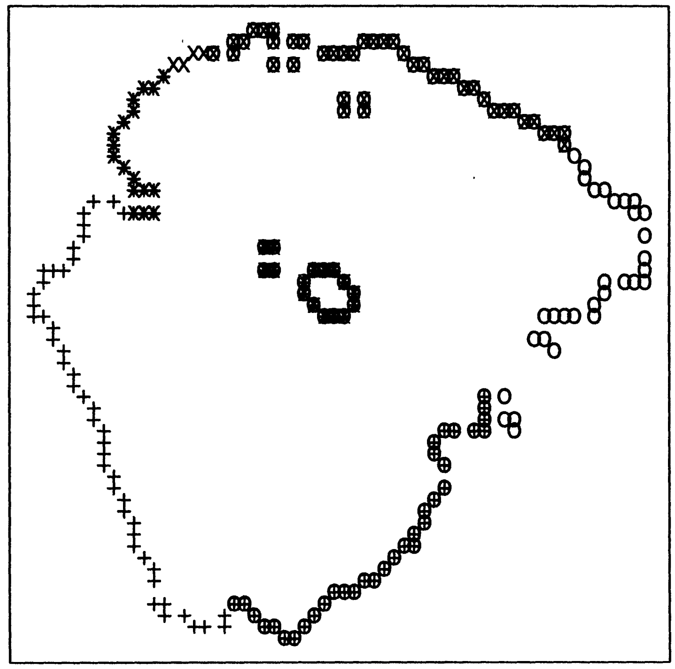


Figure 10. Noise in the Interior of a Floe Can Erode Outwards and Cause the Floe to be Subdivided. In this case, the melt ponds in the center caused the floe to be subdivided into three partial floes (indicated by an ×, +, and a ○).

was generalized by Banfield and Raftery (1989), who also showed that it is optimal when the data are generated by a mixture of normal distributions with covariance matrices whose eigenvalues are constant across clusters. This corresponds to clusters that tend to be elliptical with the same size and shape but different orientations.

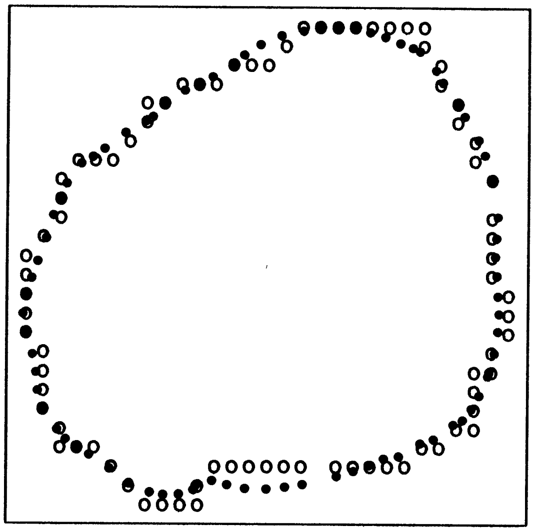
We now apply the idea of decomposing and reweighting the within-group sum of squares to the present problem. The edge pixels for an ice floe that has not been subdivided should be (a) tightly clustered about the floe outline, as estimated by the principal curve, and (b) regularly spaced along the outline, so that the variance of the distances between neighboring edge pixels should be small; see Figure 11.

We decompose the variance, V, for a group of edge pixels into parts corresponding to (a) and (b), and a residual part, as follows. Let the locations of the edge pixels be $\mathbf{x}_j$ ($j = 1, \ldots, n$), ordered so that $\hat{\mathbf{f}}(\lambda_j) = \mathbf{x}_j$, and $\lambda_1 \leq \cdots \leq \lambda_n$, where $\hat{\mathbf{f}}$ is the principal curve of the group, estimated by the method of Section 1.2. Let $\boldsymbol{\varepsilon}_j = \hat{\mathbf{f}}(\lambda_j) - \hat{\mathbf{f}}(\lambda_{j+1})$ (subscripting is modulo n) be the vector defined by two adjacent projections onto the estimated principal curve.

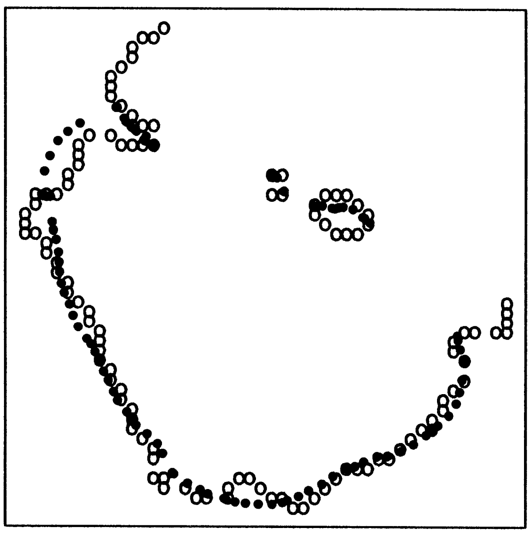
We may now write the within-group variance as

$$V = \sum_{j=1}^{n} \|\mathbf{x}_j - \bar{\mathbf{x}}\|^2 = V_{\text{about}} + V_{\text{along}} + V_{\text{residual}},$$

where $V_{\text{about}} = \sum_{j=1}^{n} \|\mathbf{x}_j - \hat{\mathbf{f}}(\lambda_j)\|^2$ and $V_{\text{along}} = (1/2) \sum_{j=1}^{n} \|\boldsymbol{\varepsilon}_j - \bar{\boldsymbol{\varepsilon}}\|^2$. The component V_{about} is a measure of lack of tightness of the distribution of the edge pixels *about* the floe outline and thus of the extent to which requirement (a) is not satisfied. The component V_{along} is a measure of the lack of regularity



(a)



(b)

Figure 11. A Complete Floe (a) and a Partial Floe (b), Showing the Edge Pixels Identified by the EP Algorithm (open circles) and Their Projections Onto the Estimated Principal Curve (solid circles). The distance between adjacent projections in (a) has a smaller variance than the distance between adjacent projections for the partial floe shown in (b).

of the distribution of the pixels *along* the floe outline and thus of the extent to which requirement (b) is not satisfied. It was shown by Banfield (1988) that

$$V_{\text{residual}} = \frac{1}{2}\sum_{j=1}^{n} \|\boldsymbol{\varepsilon}_j\|^2 + \sum_{j=1}^{n} \mathbf{r}_j \cdot \mathbf{r}_{j+1},$$

where $\mathbf{r}_j = \hat{\mathbf{f}}(\lambda_j) - (1/n)\sum_{j=1}^{n} \hat{\mathbf{f}}(\lambda_j)$ and $\cdot$ denotes the dot product. It follows that V_{residual} is a measure of the overall size of the floe. The general reweighted within-group variance is thus

$$\alpha V_{\text{about}} + \beta V_{\text{along}} + \gamma V_{\text{residual}}.$$

The component V_{residual} contains no information of interest to us here, so we set $\gamma = 0$. Moreover, without loss of generality, we set $\beta = 1$. Our clustering criterion is therefore

$$V^* = \alpha V_{\text{about}} + V_{\text{along}}.$$

To determine whether a set of groups should be merged, we first calculate V^* for each of the individual groups, and then we calculate V^* for the union of the groups. If the union has a smaller value of V^* than any of the individual groups, we merge them, and otherwise we do not. To make the number of candidate mergers manageable, we note that, if a floe has been subdivided, the partial floes will have common edge pixels, but not conversely. Therefore, only mergers of groups with common edge pixels are considered.

To determine α, we note that, by arguments similar to those of Banfield and Raftery (1989), V^* will be an optimal criterion, conditional on the estimated principal curves, if the edge pixels are normally distributed about the floe outlines and if $E[V_{\text{along}}] = \alpha E[V_{\text{about}}]$. We therefore estimate α as the average of $V_{\text{along}}/V_{\text{about}}$ for the floes that we know were not subdivided, namely, those that have no shared edge elements. Using a span of .3 to estimate the principal curves, this yielded $\hat{\alpha} = .39$. Reasonable choices for the smoothing parameter can be determined by the time of year (early summer floes are rough and jagged; late summer floes have lost their rough edges) and location (open ocean, marginal ice zone, or within the ice pack). The procedure is relatively insensitive to the precise choice of the smoothing parameter, as one would expect from Figure 4. Since $\hat{\alpha}$ is estimated from the complete floes, it can adapt the procedure to whatever smoothing parameter is used.

Table 1. Floe Merger Results

	Criterion values	
Floes	Individual	Merged
1	11.36	
4	5.71	.64
5	4.07	
8	10.86	2.36
12	8.29	
2	1.32	
3	1.36	.86
10	6.95	
6	4.37	
11	3.71	.60
25	1.33	
34	1.90	.33

NOTE: This table shows the values of the merger criterion V* for the subdivided floes from Figure 8. The value of the criterion for the individual partial floes is shown under the Individual column. The value under the Merged column is the criterion value when the floes in the left column are merged. The fact that the value under the Merged column is smaller than any of the values under the Individual column indicates that the floes should be merged.

Table 1 shows the results of merging the partial floes in Figure 8, and Table 2 shows the results of trying to merge floes that should not be merged. In each case our method gives a result that is both correct and clearcut. Figure 12 shows the final results of the procedure, together with the identified edge pixels.

4. DISCUSSION

We have developed an almost fully automatic method for finding the outlines of ice floes in satellite images. It is

Table 2. Floe Nonmerger Results

Floes	Criterion Values: Individual	Criterion Values: Merged
27	.24	
13	.39	10.70
34	1.90	
25	1.33	7.41
7	.20	
30	.59	
31	.93	2.27

NOTE: This table shows the values of the merger criterion V* for nonsubdivided floes from Figure 8. The value of the criterion for the individual partial floes is shown under the Individual column. The value under the Merged column is the criterion value when the floes in the left column are merged. The fact that the value under the Merged column is larger than any of the values under the Individual column indicates that the floes should not be merged.

accurate and computationally efficient. It involves three new statistical techniques: a way of estimating closed principal curves that reduces both bias and variance and is robust, the EP algorithm, and a method for clustering about principal curves.

The only control parameters that need to be specified by the user are the span in the principal curve estimation algorithm and the threshold for the initial transformation to a binary image. The final results are quite insensitive to even fairly large changes in these control parameters. Once they have been set for one image, the same values can safely be used for the available stream of similar images, opening the way to truly automatic processing of the image database.

The approach would seem to be applicable more generally to the detection of nonlinear features in images. It extends cluster analysis to the case where similar pixels tend to be grouped about arbitrary curved features, open or closed, using the idea of decomposing and reweighting the within-group sum of squares proposed by Murtagh and Raftery (1984). This suggests that cluster analysis may be useful for feature extraction in images more generally.

The procedure is implemented in an object-oriented programming environment. One of the advantages of this environment is that each floe resulting from the procedure can be represented as an instance of a "floe object" and can carry with it, in the form of instance variables, specific information about the floe to be used in further analysis. It is relatively fast: A 512 × 512, eight-bit image can be analyzed in approximately one hour on a Symbolics 3600 and should be considerably faster on some of the newer workstations (e.g., about 20 minutes on a Sun-3/80 and three minutes on a Sun SPARCstation 1). The processing time is linear in the number of pixels, but it does depend on the complexity of the image.

In our implementation of the EP algorithm, we erode an

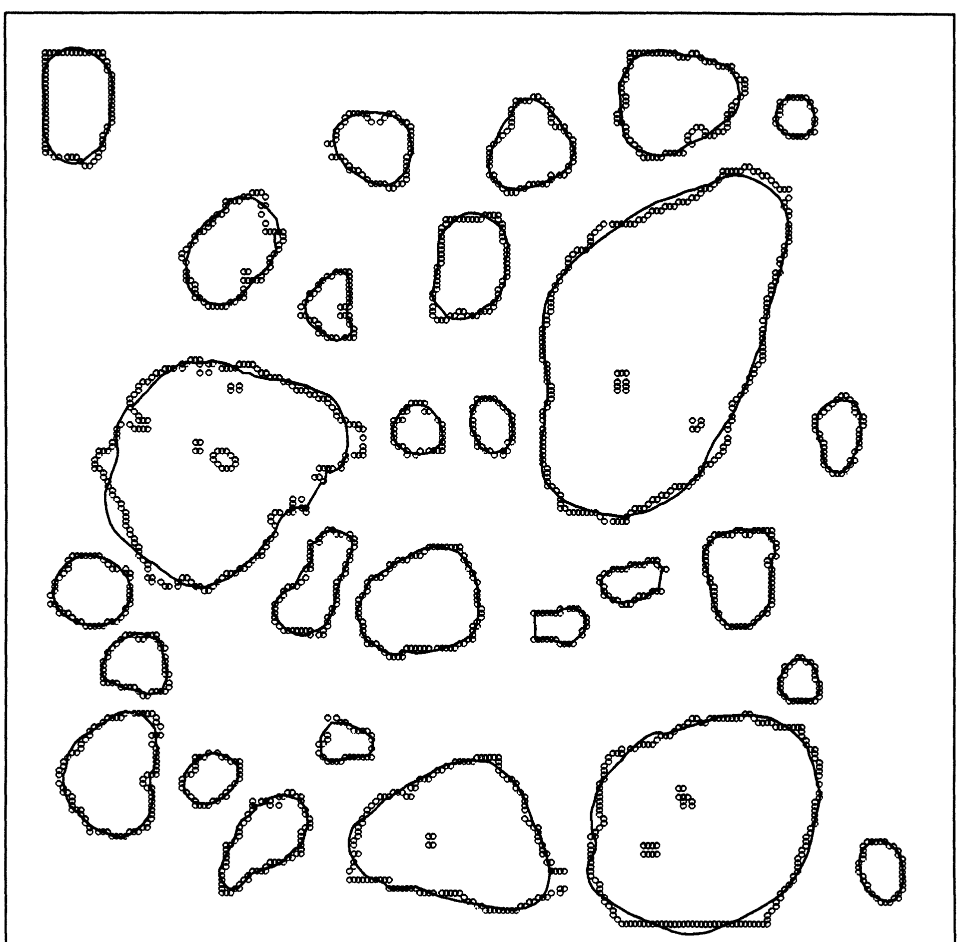


Figure 12. The Circles Are the Floe Edge Elements Found by the EP Algorithm. The lines are the principal curves of the floes, which were estimated after using the clustering method to merge partial floes.

ice pixel if any of its eight neighbors are water. Other rules for eroding a floe may be used, and they can change the rate of erosion, the effect of pixel misclassification, and the shape of the floes that can be found. The rules may also be changed as the erosion procedes. For example, dilating the image, or "refreezing" the ice, at an early iteration could eliminate some pixel misclassifications and other noise in the interior of the floes. The EP algorithm has the potential of being implemented on parallel processing machines.

To date, the development of automated techniques for the analysis of polar satellite images has been limited to ice floe tracking (Ninnis, Emery, and Collins 1986; Fily and Rothrock 1986, 1987; Vesecky, Samadani, Smith, Daida, and Bracewell 1988). The primary tool in these automated tracking methods is cross-correlation, which provides the ability to match regions in two different images, but does not give any information about the morphology of the individual ice floes or the spatial structure of the ice pack. Vesecky et al. (1988) used segments of ice floe boundaries to track ice floe movements, but this does not provide the type of information needed to study ice floe morphology and spatial structure. The need for more information on both morphology and spatial structure was clearly shown by the 1984 Marginal Ice Zone Experiment (Burns et al. 1987; Campbell et al. 1987). Floe outlines estimated using the present methods have been used to track individual floes through a series of remotely sensed images (Banfield 1991).

A referee has suggested that the entire problem could be solved using mathematical morphology in combination with the propagation algorithm introduced here, but without recourse to principal curves or clustering. This would be done by applying a thinning process (Rosenfeld and Kak 1982; Pavlidis 1982) to produce a skeletal set of pixels or medial axis. Such an approach would probably work well when the floes of interest are well separated, however, this is not usually the case. In the situation where the floes are not well separated, methods based on thinning algorithms would tend to merge floes when it is not appropriate. The problem then becomes one of how to break the floes apart, rather than how to merge them. This seems to leave us no further along than with the output of the EP algorithm. While such an approach is certainly worth further investigation, we feel that the work reported here does provide a satisfactory solution to the ice floe identification problem.

The problem considered here does not fall neatly into one of the problem areas in image understanding that have been intensely studied in recent years, namely, image restoration, classification, segmentation, and feature extraction. It does, however, combine elements from all of them. Image restoration attempts to reconstruct a degraded image. Image classification tries to assign each pixel to one of several predetermined categories; It may be regarded as a special case of restoration. Image segmentation (Rosenfeld and Kak 1982; Chellappa and Sawchuk 1985) seeks to identify areas of contiguous pixels that are, for example, devoted to the same crop. The aim of feature extraction is to find linear or curvilinear features in images.

Our problem shares goals with feature extraction, segmentation, and classification. While restoration is not an explicit goal, we would expect the methods developed here to work well in the presence of degradation. Restoration and classification methods do not, by themselves, address the present problem. Current feature detectors would seem to have difficulty locating features as arbitrary as ice floes. For example, the Hough transform (Hough 1962; Ballard 1981; Davis 1982), an obvious candidate for locating closed curves, requires an initial pattern description which it then tries to find in the image. It would be difficult to provide an initial pattern description that is general enough to accommodate the wide range of commonly found ice floe shapes. Our approach may also be applicable to segmentation problems, especially those concerned with identifying not only regions but also the shapes of their outlines.

A large variety of commercial software is available for image processing. Many of these programs can identify *distinct* figures but provide only a rudimentary capability, based on template matching, for identifying overlapping figures. This limits their application to images in which all the figures have a similar shape. Our method can distinguish between arbitrarily shaped overlapping figures since it is based on an edge pixel model instead of templates. In addition, our method can identify and properly handle "holes," such as melt ponds. None of the commercial packages that we are aware of have this capability.

The Bayesian and stochastic relaxation approach of Geman and Geman (1984) may well be applicable to the present problem, although it has to date been used mainly for restoration. It would require extensive modeling assumptions for the ice floe problem, and experience suggests that it would be computationally expensive. The computational burden might be reduced by using as an approximation the ICM algorithm of Besag (1986), although this does not yet appear to have been applied to problems such as the present one. Ripley (1989) has suggested using mathematical morphology to provide starting values for an approach to such problems based on Bayesian image reconstruction. Our procedure on the other hand requires only the assumption that the ice floe boundaries be closed curves, and it is relatively fast.

[*Received August 1989. Revised May 1991.*]

REFERENCES

Ballard, D. H. (1981), "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, 13, 111–112.

Banfield, J. (1991), "Automated Tracking of Ice Floes: A Stochastic Approach," IEEE Transactions on Geoscience and Remote Sensing, 29, 905–911.

——— "Constrained Cluster Analysis and Image Understanding," unpublished Ph.D. dissertation, University of Washington, Dept. of Statistics.

Banfield, J. D., and Raftery, A. E. (1989), "Model-Based Gaussian and Non-Gaussian Clustering," Technical Report 186, University of Washington, Dept. of Statistics.

Besag, J. (1986), "On the Statistical Analysis of Dirty Pictures" (with discussion), *Journal of the Royal Statistical Society*, Ser B, 48, 259–302.

Burns, B. A., Cavaileri, D. J., Keller, M. R., Campbell, W. J., Grenfell, T. C., Maykut, G. A., and Gloersen, P. (1987), "Multisensor Comparison of Ice Concentration Estimates in the Marginal Ice Zone," *Journal of Geophysical Research*, 92, 6843–6856.

Campbell, W. J., Gloersen, P., Josberger, E. G., Johannessen, P. S., Guest, P. S., Mognard, N., Shuchman, R., Burns, B. A., Lannelongue, N., and Davidson, K. L. (1987), "Variations of Mesoscale and Large Scale Sea Ice Morphology in the 1984 Marginal Ice Zone Experiment as Observed by Microwave Remote Sensing," *Journal of Geophysical Research,* 92, 6805–6824.

Chellappa, R., and Sawchuk, A. (1982), *Digital Image Processing and Analysis* (2 vols.), Silver Spring, MD: IEEE Computer Society Press.

Committee on Applied and Theoretical Statistics (1989), "Discriminant Analysis and Clustering," *Statistical Science,* 4, 34–69.

Davis, L. S. (1982), "Hierarchical Generalized Hough Transforms and Line-Segment Based Generalized Hough Transforms," *Pattern Recognition,* 15, 277–285.

Fily, M., and Rothrock, D. A. (1986), "Extracting Sea Ice Data From Satellite SAR Imagery," *IEEE Transactions on Geoscience and Remote Sensing,* 24, 849–854.

——— (1987), "Sea Ice Tracking by Nested Correlations," *IEEE Transactions on Geoscience and Remote Sensing,* 25, 570–580.

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6, 721–741.

Gordon, A. D. (1981), *Classification,* London: Chapman and Hall.

——— (1987), "A Review of Hierarchical Classification," *Journal of the Royal Statistical Society,* Ser. A, 150, 119–137.

Hastie, T. (1984), "Principal Curves and Surfaces," unpublished Ph.D. dissertation, Stanford University, Dept. of Statistics.

Hastie, T., and Stuetzle, W. X. (1985), "Principal Curves and Surfaces," Technical Report 56, University of Washington, Dept. of Statistics.

——— (1989), "Principal Curves," *Journal of the American Statistical Association,* 84, 502–516.

Hough, P. V. C. (1962), "Method and Mean for Recognizing Complex Patterns," U.S. Patent 3069654.

Matheron, G. (1975), *Random Sets and Integral Geometry,* New York: John Wiley.

Murtagh, F. (1985), *Multidimensional Clustering Algorithms* (COMPSTAT Lectures, 4), Vienna: Physica-Verlag.

Murtagh, F., and Raftery, A. E. (1984), "Fitting Straight Lines to Point Patterns," *Pattern Recognition,* 17, 479–483.

Ninnis, R. M., Emery, W. J., and Collins, M. J. (1986), "Automated Extraction of Pack Ice Motion From Advanced Very High Resolution Radiometer Imagery," *Journal of Geophysical Research,* 91, 10725–10734.

Pavlidis, T. (1982), *Algorithms for Graphics and Image Processing,* Rockville, MD: Computer Science Press.

Ripley, B. D. (1989), "Object Recognition in Images Using Models of Geometric Structures," in *Geobild '89,* eds. A. Hübler, W. Nagel, B. D. Ripley, and G. Werner, Berlin: Akademie-Verlag, pp. 87–92.

Rosenfeld, A., and Kak, A. C. (1982), *Digital Picture Processing* (2 vols.), New York: Academic Press.

Rothrock, D., and Thorndike, A. (1984), "Measuring the Sea Ice Floe Size Distribution," *Journal of Geophysical Research,* 89, 6477–6486.

Serra, J. (1982), *Image Analysis and Mathematical Morphology,* New York: Academic Press.

Tukey, J. W. (1977), *Exploratory Data Analysis,* Reading, MA: Addison-Wesley.

Vesecky, J. F., Samadani, R., Smith, M. P., Daida, J. M., and Bracewell, R. N. (1988), "Observation of Sea-Ice Dynamics Using Synthetic Aperture Radar Images: Automated Analysis," *IEEE Transactions on Geoscience and Remote Sensing,* 26, 38–48.

Ward, J. H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association,* 58, 236–244.