The University of Kansas

Information and
Telecommunication
Technology Center

A Technical Report of the
Networking and Distributed Systems (NDS) Laboratory

The Restricted Growing Concept for Object Separation

LeenKiat Soh
Costas Tsatsoulis

ITTC-FY98-TR-11810-02

January 1998

Project Sponsor:

Naval Research Laboratory

The Restricted Growing Concept for Object Separation

ABSTRACT

In this paper we describe a concept that separates touching objects and, based on that concept, we present a two-tiered technique for image processing. First, to establish separation, we employ probabilistic morphology and threshold decomposition. As a result of this process, most objects are reduced in size. Then, to regain size and shape, we grow presently separated but distorted objects back to their original forms under the guidelines of a set of structural rules. These rules, derived by reversing the functionality of a thinning algorithm, forbid an image pixel from becoming an object pixel if such growth destroys the established separation in the vicinity. This strategy is termed as the Restricted Growing Concept which differs from other object separation methodologies in terms of its capability of allowing separated objects to regain and preserve theirs sizes and shapes while maintaining the separation among objects. We also present an evaluation of our technique in different imagery applications and in comparison with other conventional morphological region growing techniques.

1. INTRODUCTION

When two gray level objects touch with shared boundaries, it makes shape analysis and recognition difficult. For example, in industrial vision applications [7], touching objects hinder the recognition of the object's shape and complicate the task of defect inspection. In aerial image and terrain analysis, where the objective is to identify items such as airplanes on an airfield [22], failure to obtain isolated objects creates misunderstanding in the resulting image context. In shape analysis where geometric descriptors are computed for each object, treating a conglomerate of touching objects falsely as one leads to incorrect measurements [11]. Hence, separating objects is important in object-based image analysis.

In object separation, our objectives are (1) to achieve object separation, and (2) to preserve (or approximate as closely as possible) the object's original shape and size. The fact that the two goals cannot be optimized simultaneously further adds to the complexity of this task. This is shown in Fig. 1.

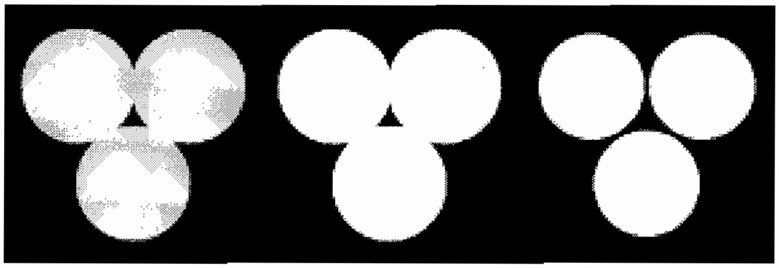

Fig. 1 The conflict between separation and preservation in object separation. The raw image consists of three circles with decreasing intensities at the boundaries. The image in the middle shows the segmentation result of using a low threshold value. This segmentation preserves the size and shape of the objects but fails to separate them. The right image shows the segmentation result of using a high threshold value. It achieves object separation among the three circles at the expense of size and shape preservation.

The figure shows an image with three touching objects, all having pixels with lower intensity values towards the boundary. One simple way of obtaining the objects would be thresholding. Using a low threshold, we would be able to extract the objects with complete shape and size preservation; however, the objects might, as a result, be connected. Using a high threshold, on the other hand, we would be able to separate the objects, but would lose the boundary and some layers of interior object pixels. This tradeoff between separation and preservation of size and shape is inherent in all object separation algorithms. To address this problem, we have designed an object separation technique based on a new concept called the restricted growing concept. This concept divides the task and the accomplishments of the two objectives into two tiers. First, it achieves the separation. Second, it re-establishes the sizes and shapes of the objects lost or distorted during the process that first achieved the separation. It is with this second tier that our concept stands alone among existing object separation strategies. To restore the objects to their original sizes and shapes while preserving the separation, we use a structural algorithm that forbids two growing objects from encroaching. Therefore, we are able to achieve separation with high preservation rate of pixels of the objects. In this paper, we show one alternative design of our concept that utilizes a deviated type of morphology that we call probabilistic morphology to derive separation among objects and employs a reversed-thinning algorithm to restore the objects to their original sizes and shapes. Note that this implementation does not represent the optimal implementation of the restricted growing concept; it does, however, provide us with an applicable technique to demonstrate the usefulness and advantages of the concept. In addition to the results discussed here, the technique has been intalled at the Jet Propulsion Laboratory (JPL) as part of image segmentation package for analyzing sea ice floes of Synthetic Aperture Radar (SAR) imagery [32].

In the next section, we present a background discussion on some mathematical morphology operators and some existing object separation techniques. Section 3 describes the restricted growing concept and its methodology. Section 4 defines the probabilistic morphology and its implementation. Section 5 discusses the restricted growing algorithm. Section 6 shows the results of our technique. Section 7 concludes the paper.

2. BACKGROUND

Mathematical morphology has been widely used, due to its strong theoretic foundations and flexible operators, as an image processing tool in areas such as edge detection, object recognition, image segmentation, image enhancement, noise removal, and non-linear filtering [29][34]. The original work was developed by Matheron [21] and extended by Serra [28]. Since then, [9] has presented a thorough discussion of both binary morphology and gray scale morphology that defines different operators such as dilation, erosion, closing, and opening, and [33] has provided new properties of these operators when utilizing multiple structuring elements. Moreover, the algebraic basis of mathematical morphology [12][25] and its geometric basis [35] have been explored. In what follows, the definitions of binary morphological operators from [8] will be used.

2.1. *Binary Morphology*

Suppose A and B are sets in E^N, with components $a=(a_1,a_2,\cdots,a_N)$ and $b=(b_1,b_2,\cdots,b_N)$, respectively and N is the number of dimensions. The *translation* of A by $x=(x_1,x_2,\cdots,x_N)$, denoted $(A)_x$, is defined as

$$(A)_x=\{c|c=a+x, \text{for } a\in A\}. \tag{1}$$

The *reflection* of B, denoted $\hat{B}$, is defined as

$$\hat{B}=\{x|x=-b, \text{for } b\in B\}. \tag{2}$$

The *complement* of set A is

$$A^c=\{x|x\notin A\}. \tag{3}$$

The *difference of* two sets A and B, denoted $A - B$, is defined as

$$A - B = \{x | x \in A, x \notin B\} = A \cap B^c. \tag{4}$$

Let $\varnothing$ be the empty set. The *dilation* of A by B, denoted $A \oplus B$, is defined as

$$A \oplus B = \{x | (\hat{B})_x \cap A \neq \varnothing\} = \{x \| [(\hat{B})_x \cap A] \subseteq A\} \tag{5}$$

Note that B is sometimes referred to as the kernel or structuring element, while A as the active image or image. The *erosion* of A by B, denoted $A \ominus B$, is defined as

$$A \ominus B = \{x | (B)_x \subseteq A\}. \tag{6}$$

Fig. 2 shows the effects of dilation and erosion operations.

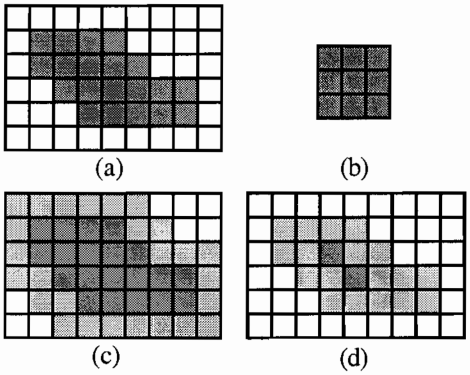

Fig. 2. (a) active image A; (b) structuring element B; (c) A+B: the dark and light shaded areas, (d) A-B: the dark shaded areas.

The *opening* of set A by B, denoted $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B. \tag{7}$$

The *closing* of A by B, denoted $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B. \tag{8}$$

Therefore, opening is applying erosion and dilation sequentially; whereas closing is dilation and erosion sequentially.

2.2. *Discussion*

Opening usually eliminates thin protrusions and breaks narrow bridges between objects. On the other hand, closing usually fills small holes, strengthens linear structures, and eliminates gaps in the contour of an object. However, once an object has been closed or opened, further applications of closing or opening using the same structuring element will not modify the object—connections between objects that were not broken will not be broken at a later iteration. Usually, iterative erosion alone is used.

[2] used a method called erosion-propagation (EP) algorithm coupled with clustering about principle curves [10] to identify objects in satellite images. The EP algorithm morphologically erodes a pixel from object to non-object if any of its 8-neighboring pixels is non-object. As the edge of an object is gradually eroded during the iterative process, the locations of the edge pixels are propagated toward the interior of the object. This information is later used in the clustering about principle curves to determine whether merging of objects is necessary to remedy the effects of excessive erosion. This object separation technique suffers from several disadvantages: (1) the number of iterations required to achieve separation has to be determined manually for each image, (2) objects smaller than $(2i + 1) \times (2i + 1)$ pixels, where i is the number of iterations, will be eliminated, and (3) objects do not preserve their original size.

In another approach [5], a tagging algorithm was used to separate objects with weak connections. At each growing iteration, every object pixel that is untagged and 8-neighbor to the existing grown object pixels is a candidate. A candidate is tagged as belonging to a feature if (1) it has at least two 8-neighbor pixels that are either the candidates of previous iteration or pixels of the feature, and (2) the two pixels must be 4-neighbor to each other. The process grows each object individually; unlike the EP algorithm which applies the same number of iterations to all objects non-discriminatingly, this tagging algorithm is thus able to retain small objects. However, the authors' implementation handled only regions connected by corners or by one-node bridges, rendering it incapable of achieving separation when stronger connections occur.

[29] proposed an interesting approach to separate objects. With the assumption that touching objects create sharp turns or corners in the boundaries where the objects meet, the authors designed a set of constraints to decide whether two corners should be connected. The technique first locates a chain of corners on the boundary and then applies the constraints (which were

based on geometric heuristics and semantic properties of the boundaries), and separation among objects is finally created by linking corners of different boundaries. This approach is able to achieve good separation and complete preservation. However, since it relies on the definition of a corner, it is noise-sensitive and limited to domains in which the aforementioned assumption holds.

In this paper, we present a technique that is able to break strong connections, preserve original shape and size, retain small objects, and handle noisy image.

3. THE RESTRICTED GROWING CONCEPT

To achieve separation and preservation, we have designed an innovative concept called the Restricted Growing Concept (RGC). The main idea of this concept is to decompose the object separation problem into two modules, each dedicated to achieve one of the objectives as outlined in the sections above. Subsequently, as we integrate the two modules together, we achieve both objectives concurrently. The first objective is separation. This is done by shrinking objects such that each object is separated from its touching neighbors. The second objective is preservation of size and shape. This is accomplished by growing the shrunk objects to restore the size and shape. However, the two objectives are contradictory; and to ensure that the separation that has been established after the shrinking is not disturbed, our growing process is *restricted*. This has given rise to our RGC.

Here we outline definitions related to the concept. A *core object* is a version of the original object such that its linkages to neighboring objects are disconnected, satisfying our first objective of object separation. Such an object is reduced in size, but it usually captures the general shape of its original version. An image with core objects is thus a *core image*. A *skin object* is a version of the original object such that the original size of the object is preserved. Skin objects are usually interconnected and could encompass one or more core objects. An image with skin objects, which will serve as constraints on the growing of their cnclosed core objects, is a *skin image*. Note that a less constrained version of the definition is to allow the skin object to be a close approximation of its original. Finally, a *restricted growing algorithm* grows from a core object within the boundary of its corresponding skin object while preserving the object's separation from its neighbors. This definition implies that the growing process stops either when the boundary of the object has been reached or when further growing will damage the object's separation from its neighbors. Thus, conventional region growing [38] or morphological dilation schemes are not restricted growing algorithms. This concludes our definitions for the RGC. The concept itself is intuitive and general; it also covers both requirements of object separation. In the following sections, we present a solution to obtaining the core objects and an algorithm for the restricted growing process.

Fig. 3 shows the execution of the RGC. The original image in this example consists of touching sea ice floes (objects) observed in satellite imagery of the Arctic. Our task is to identify each individual object. First, we generate the skin image, which is a binary segmentation of the original image that identifies the object and background pixels. As noted from Fig. 3, objects are now observed as a network of interconnected entities, making individual object analysis impossible. Second, we generate the core image. Similar to the skin image, the core image is a binary version of the original image. However, the objects are now separated from their touching neighbors, isolated through a shrinking proces.

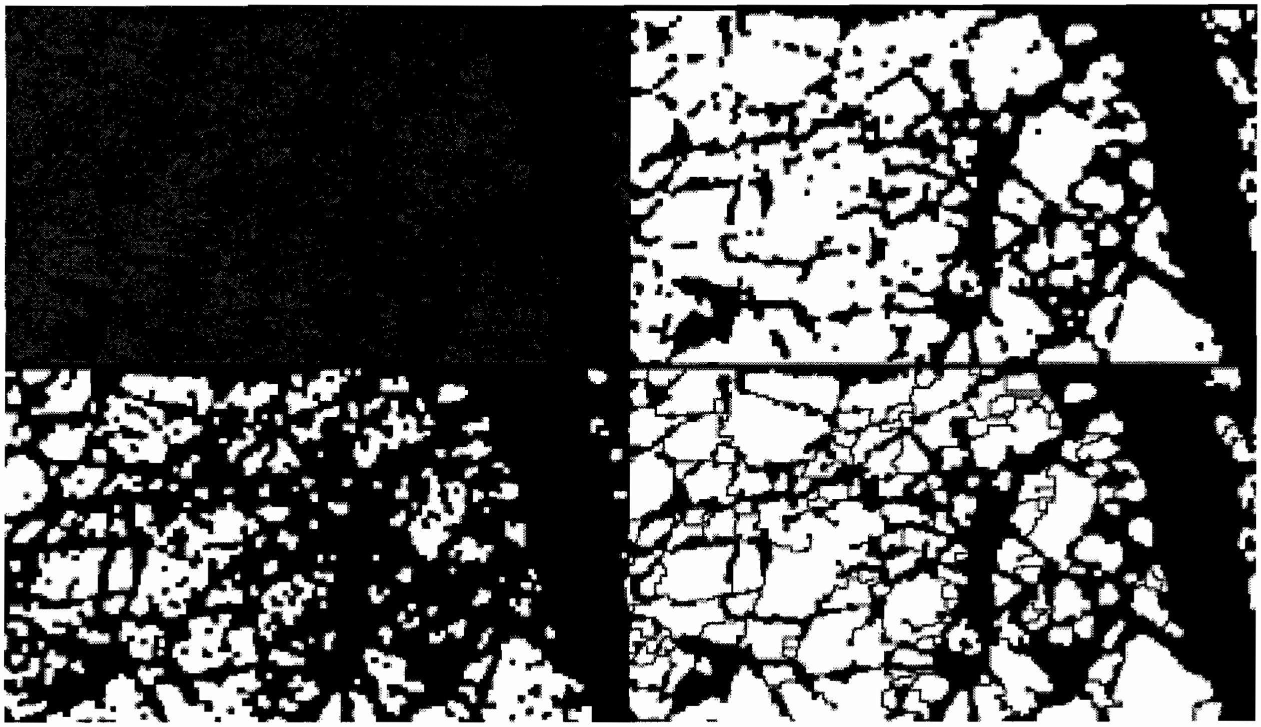

Fig. 3 The execution of the Restricted Growing Concept. The upper left image shows an aircraft STAR-2 SAR sea ice imagery (obtained from Dr. J. Comiso of NASA Goddard SFC). The upper right image is the skin image. The lower left image is the core image. The lower right image is the result of restricted growing. Note that skin image retains the shape and size of the objects and core image establishes the separation among the objects. The RGC process combines the two to achieve object separation.

Comparing the skin and core images, one can observe that the skin objects always enclose one of more core objects in the corresponding area. During the restricted growing process, a core object will not be grown outside of the enclosure. In addition, a pixel will not be grown from non-object to object class when such growth will destroy the existing separation status of the area. As can be observed from the Fig. 3, for example, objects at the bottom right corner have been disconnected from their neighbors while shapes and sizes have been preserved.

4. PROBABILISTIC MORPHOLOGY

To obtain the core image, we use the idea of erosion. However, the morphological erosion operator erodes an object non-discriminatingly and requires manual inspection as a stopping condition. In our implementation, we have designed an operator that is able to erode *actively* the shared boundary of two objects while remaining *passive* to other pixels. In addition, we assume a ramp edge that, if the objects are brighter (or darker) than the background, pixels at the proximity of the boundaries of the objects are darker (or brighter) than those at the interior, which is usually observed in most images.

4.1. *Probabilistic Morphology*

Suppose A and B are sets in E^N, with components $a=(a_1,a_2,\cdots,a_N)$ and $b=(b_1,b_2,\cdots,b_N)$, respectively. The *binary probabilistic dilation* of A by B, denoted as $A\oplus^\alpha B$, where α is a probability threshold, is defined as

$$A\oplus^\alpha B=\{x|x'\in C_0,P((B)_x\cap A)\geq\alpha\}, \qquad (9)$$

where C_0 contains pixels of non-object class and x' is the pixel class before the operation. The binary probabilistic *erosion* of A by B, denoted $A\ominus{}^\alpha B$, is defined as

$$A\ominus{}^\alpha B=\{x|x'\in C_1,P((B)_x\cap A)\geq\alpha\}, \qquad (10)$$

where C_1 contains pixels of object class and x' is the pixel class before the operation. The two equations are essentially the same, except that dilation is applied only to non-object pixels and erosion to object pixels. Note that P denotes the probability. Thus, a pixel will be dilated (changed from non-object to object) if its neighbors match the structuring element to a certain degree of probability (higher than the threshold α). Similarly, a pixel will be eroded (changed from object to non-object) if the probability of its neighbors matching the structuring element is less than the threshold α. In traditional, non-probabilistic binary morphology, dilation is carried out as long as there is one object pixel, and erosion is performed whenever there is one non-object pixel in the neighborhood. Suppose we define

$$P((B)_x \cap A) = \frac{\|(B)_x \cap A\|}{\|B\|}, \qquad (11)$$

where $\|f\|$ denotes the number of object class pixels covered by the area of f. Then

$$A \oplus B = A \oplus^{1/\|B\|} B, \text{ and} \qquad (12)$$

$$A \ominus B = A \ominus^{1.0} B. \qquad (13)$$

We argue that the conventional morpohological dilation is too inclusive and erosion is too exclusive. Using a probabilistic version, we instead examine the neighborhood of a pixel to measure the validity or confidence of that pixel getting dilated or eroded. In the case of erosion, the probabilistic version thus is able to erode actively boundary pixels while remaining passive to pixels with substantial number of object neighbors.

4.2. *Implementation*

The implementation of probabilistic morphology lies mainly on the computation of the probability function. In our object separation technique, we use binary probabilistic erosion to create the core and skin images; thus, we only provide the solution for the erosion operator in this paper. To implement $P((B)_x \cap A)$, we have used threshold decomposition [30] and superposition [20]. Suppose an image A is represented by $g(m,n)$, which denotes the intensity of the pixel located at (m,n), and that $g(m,n)$ takes values between $l = 0,1,2,\cdots,L-1$, where L is the number of intensity levels in the image. By thresholding g at all possible intensity levels, we obtain the threshold binary images [20]:

$$g_l(m,n) = \begin{cases} 1, & g(m,n) \geq l \\ 0, & g(m,n) < l \end{cases}. \qquad (14)$$

The threshold binary images can be combined to reconstruct $g(m,n)$:

$$g(m,n)=\sum_{l=0}^{L-1} g_l(m,n)=\max\{l|g_l(m,n)=1\}. \quad (15)$$

Suppose $l' \in \{T+(k-1)s, T+ks, T+(k+1)s, \cdots, T+Ks\}$, $l' \in l$, $k=1,2,\ldots,K$, where K is the number of elements in the set, s is the interval between each pair of successive element values, and T is the starting threshold value. Then,

$$g'(m,n;T,s,K)=\sum_{k=1}^{K} g_{T+ks}(m,n)=\max\{k|g_{T+ks}(m,n)=1\}. \quad (16)$$

Note that $g(m,n)=g'(m,n;0,1,L-1)$. Given the above formulations, we now define our design of the probability function. Our assumption of interactions of objects is that intensities at the boundaries where objects meet are lower than those in the interior, and the objects are brighter than the background (and the opposite is also accommodated by our formulation). As we increase the value of the threshold, we discard more and more boundary pixels. On the other hand, if a pixel is surrounded by high intensity neighbors, it is very likely to be an object pixel. Hence, we use a successive thresholding scheme to slice the image intensities into several planes and collect object-class pixels across the planes. The probability is defined as the ratio of the number of these object-class pixels to the accumulative size of the structuring element, that is the number of object-class pixels covered by the structuring element. We define

$$P\left((B)_x \cap A\right)=\frac{\sum_{k=1;(m,n)\in B_x}^{K} \left\| g_{T+ks}(m,n) \cap (B)_x \right\|}{\sum_{k=0}^{K-1} \|B\|}. \quad (17)$$

The structuring element B can be a 4-neighborhood, 8-neighborhood, 2-unit city-block neighborhood, etc. If we use a 8-neighborhood structuring element (a window consists of 3 × 3 pixels of object-class), then the intersection of such a window with a variable will be the variable itself. Thus, we can simplify Eq. (17) to

$$P\left((B)_x \cap A\right)=\frac{\sum_{k=1;(m,n)\in B_x}^{K} \left\| g_{T+ks}(m,n) \right\|}{\sum_{k=0}^{K-1} \|B\|}=\frac{\sum_{(m,n)\in B_x} g'(m,n;T,s,K)}{K \times \|B\|}. \quad (18)$$

Our implementation of the core image of A by B is:

$$Core_B(A) = A \ominus^{0.90} B = \left\{ x \left| \frac{\sum_{(m,n)\in B_x} g'(m,n;t,2,5)}{5 \times \|B\|} \geq 0.90 \right. \right\}, \quad (19)$$

where t is a threshold value generated by a segmentation phase, $s = 2$, and $K = 5$.

$$Skin_B(A) = A \ominus^{0.75} B = \left\{ x \left| \frac{\sum_{(m,n)\in B_x} g'(m,n;t,2,3)}{3 \times \|B\|} \geq 0.75 \right. \right\}. \quad (20)$$

The values of K, s, and α were determined experimentally. Throughout our testing of the technique, these values were never changed, proving that they were general for different types of imagery. On the other hand, these parameters could be changed to adapt to other type of imagery, providing three degrees of freedom for each image.

The motivations behind using successive threshold slices are threefold. As we increase the value of the threshold at a pixel's neighbor, we impose a tougher environment for that pixel to have object-class neighbors. This increasing level of constraint serves as a refining filter to weed out boundary pixels—therefore, the design is able to retain small objects. Also, our segmentation techniques (global thresholding or local thresholding) cannot locate precisely the desirable threshold at every pixel. This problem may lead to misidentification of object and non-object pixels. With successive threshold values, we, in effect, segment the image using a range of thresholds; and hence relieve the burden of the segmentation process to obtain the exact threshold value. Third, images are noisy and especially so at the object boundaries where luminations are not consistent. By decomposing the image into different slices, we break down the noise effects by aggregating layers of neighborhood values. This makes the design less susceptible to noise.

5. THE RESTRICTED GROWING ALGORITHM

The objective of the restricted growing algorithm is to grow core objects within the boundary of their corresponding skin objects while preserving existing separation among the core objects. In our definition, a growing algorithm (1) expands the core object to its original shape and size, and

(2) introduces no additional connectivity to its neighbors. Thinning or skeletonization, on the contrary, is a process that (1) reduces an object to its skeleton, and (2) ensures connectivity of skeletal branches within the object. That these two processes are closely related and opposite to each other prompted us to design the restricted growing algorithm by modifying a thinning algorithm. Formally, a thinning algorithm reduces a binary object into a one-pixel thick skeleton, which should (1) preserve the object's topology, (2) be central to the object, or (3) enable the original object to be recoverable from the skeleton [19]. The definition of a skeleton was first introduced by Blum [4] as the medial axis, obtained by setting up grassfires simultaneously at all fronts. However, this method led to alterations of both skeletal legs [13] and connectivity of the object [26]. Numerous designs of thinning algorithms have since appeared in literature [15][16], aimed at improving the rate of convergence, accuracy, connectivity, and preservation of skeletal legs.

5.1. *A Thinning Algorithm*

Our restricted growing algorithm is based on the thinning algorithm proposed by Zhang and Suen [37]. The thinning algorithm is described as follows. Suppose the object pixels are assigned 1 and non-object pixels 0, and a boundary pixel is a pixel of an object that has at least one non-object 8-neighbor. This agorithm consists of iterations of two basic steps applied to the boundary pixels of an object. The 8-neighborhood numbering is as shown in Fig. 4

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Fig. 4 The numbering sequence of the 8-neighborhood of a pixel P1.

During the first step, a boundary pixel p_1 is flagged if the following conditions are satisfied:

(1) $2 \leq N(p_1) \leq 6$,
(2) $S(p_1) = 1$,
(3) $p_2 \cdot p_4 \cdot p_6 = 0$, and
(4) $p_4 \cdot p_6 \cdot p_8 = 0$,

where $N(p_1)$ is the number of nonzero neighbors of the boundary pixel p_1, and $S(p_1)$ is the number of 0-1 transitions in the ordered sequence of the neighbors of p_1. During the second step, a boundary pixel p_1 is flagged if

(1) $2 \leq N(p_1) \leq 6$,
(2) $S(p_1) = 1$,
(3) $p_2 \cdot p_4 \cdot p_8 = 0$, and
(4) $p_2 \cdot p_6 \cdot p_8 = 0$.

One iteration of the algorithm includes applying the first step to flag boundary pixels for deletion, deleting the flagged pixels, applying the second step to flag remaining boundary pixels for deletion, and deleting the flagged pixels; and it iterates until no further pixels are deleted.

The first condition is violated when p_1 has only one or seven 8-neighbors valued as object pixels. A boundary pixel having only one such neighbor implies that it is situated at the end of a skeletal leg, and therefore should not be deleted. On the other hand, if it has seven such neighbors, the deletion of the pixel would penetrate into the object and thus is not allowed. The second condition is violated when the neighborhood contains a one pixel thick line ($S(p_1) > 1$). A deletion of such a boundary pixel is forbidden since it would introduce a break to the connectivity of the object. To satisfy the last two conditions, as well as the first two conditions, of the first step, a pixel must be an east or south boundary pixel or a northwest corner pixel in the boundary. Correspondingly, during the second step, a pixel must be a north or west boundary pixel or a southeast corner pixel in the boundary. Any point matching any of these four patterns is not part of the skeleton and should be removed.

5.2. *The Restricted Growing Algorithm*

Our design of the restricted growing algorithm was based closely on the thinning algorithm described above. Note that a pixel is deleted if it satisfies all conditions mentioned in the above subsection during the thinning process. Here, a pixel is grown or rejected from growing based on the decisions as shown in Fig. 5.

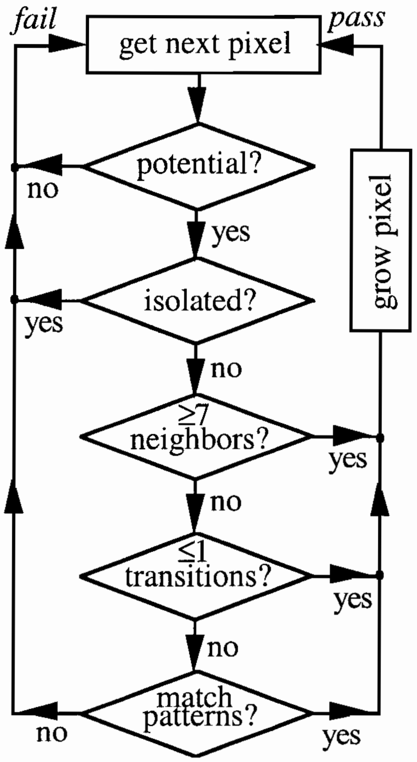

Fig. 5 The tests performed sequentially by the restricted growing algorithm.

Test 1: Potential Growing Condition If a pixel in the core image is a non-object pixel and its corresponding pixel in the skin image is an object pixel, then the pixel is qualified for further test.

This test selects only non-object core pixels for potential growth. This is how we guarantee that we only grow core objects within the boundary of skin objects. In addition, pixels that are non-object in both core and skin images are deemed as true non-object pixels without any potential of becoming object pixels.

Test 2 Isolation Condition If the pixel in the core image does not have an object pixel in its core image as an 8-neighbor, then the pixel is disqualified.

This condition avoids erroneous separation within an object. For example, small dark specks in an object would be eroded to non-object pixels during the generation of the core image. If allowed to grow, these non-object holes could become individual object regions within the object that encloses them. With this test, a small hole within an object will instead be consumed by the encroaching object during the growing process.

Test 3: Connectivity Condition 1 If the pixel in the core image has seven or eight object pixels in its core image 8-neighborhood, then the pixel is grown.

If the 8-neighborhood of an pixel has seven or more object pixels, that means all object pixels in that neighborhood are connected. Hence, the growth of the pixel from non-object to object does not damage the existing (or non-existing) separation. Note that Tests 2 and 3 correspond collectively to the first condition of the thinning algorithm.

Test 4: Connectivity Condition 2 If the pixel in the core image has less than two 1-0 transition in its core image 8-neighborhood, then the pixel is grown.

If less than two 1-0 transitions are found, that means all object pixels in the area are connected and further growth does not alter the separation. This condition is analogous to the second condition of the thinning algorithm.

Test 5: Connectivity Condition 3 If the pixel in the core image has its core image 8-neighborhood match one of the four corner patterns, then the pixel is grown.

The *corner patterns* are shown in Fig. 6. Note that any of these patterns could have two or more 1-0 transitions yet have all its object pixels connected. Thus, a pixel growth will not alter the separation. Note that this condition is a combination of the last two conditions of both steps of the thinning algorithm.

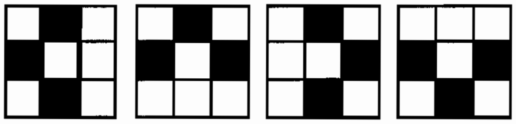

Fig. 6 Four patterns examined in Test 5 of the restricted growing algorithm. Dark pixels denote object pixels; unshaded pixels denote "don't care" pixels.

The restricted growing algorithm performs these five tests sequentially. At each iteration, the algorithm moves rasterly until the whole core image is scanned. Any growing is immediately executed after each pixel has been inspected. To delay the growth effect on the current pixel from propagating to the next pixel, the algorithm jumps to a distant pixel to continue the raster scan. The jump distance is proportional to the size of the image. The algorithm iterates until the growth rate converges to or drops below a prespecified limit such as 0 or 0.01% of the initial number of grown pixels. This sequential implementation of the algorithm is fast.

6. RESULTS

We have conducted several experiments to evaluate the performance of our restricted growing algorithm. First, we have implemented three techniques: (A) morphological erosion with manually determined number of iterations, (B) morphological erosion, with manually determined number of iterations, integrated with the restricted growing algorithm, and (C) probabilistic morphological erosion (fully automated, one iteration) with the restricted growing algorithm. The comparison between the results of techniques A and B showed the usefulness of the restricted growing algorithm; whereas the comparison between B and C the advantage of using probabilistic morphology. Fig. 7 shows the experiment on touching circles. As can be observed from the figure, technique A was able to separate the circles while failing to preserve the size and shape of the circles. Technique B fared better; it was able to restore most object pixels but was still unable to eliminate the distortion on the shape of the circles. This was caused by the excessive erosion during the first phase. However, our probabilistic morphology, as used in technique C, was able to selectively erode pixels at the boundary. This caused separation among objects without erroneously transforming the objects and allowing good restoration during the growing process. Table 1 shows the information and results of the experiment. Numerically, technique C was able to retrieve 99.05% of object pixels while separating objects with 62 shared boundary pixels.

Image	Original Object (pixels)	Shared Boundary (pixels)	Result A (pixels,%)	Result B (pixels,%)	Result C (pixels,%)
C40	15080	0	15075(99.97)	15078(99.99)	15073(99.95)
C41	15787	1	14811(93.82)	15766(99.87)	15753(99.79)
C42	16543	44	8739(52.83)	16120(97.44)	16454(99.46)
C43	17215	62	7242(42.07)	16595(96.40)	17052(99.05)

Table 1 The information and results of the experiment on touching circles. Each image consists of three circles. C40 for example has circles of radius of 40 pixels each, and so on. Original object means the total number of object pixels in the image. Shared boundary means the total number of shared boundaries of the touching circles. Result A was generated by morphology erosion only; Result B by morphology erosion with restricted growing; Result C by probabilistic morphology erosion with restricted growing.

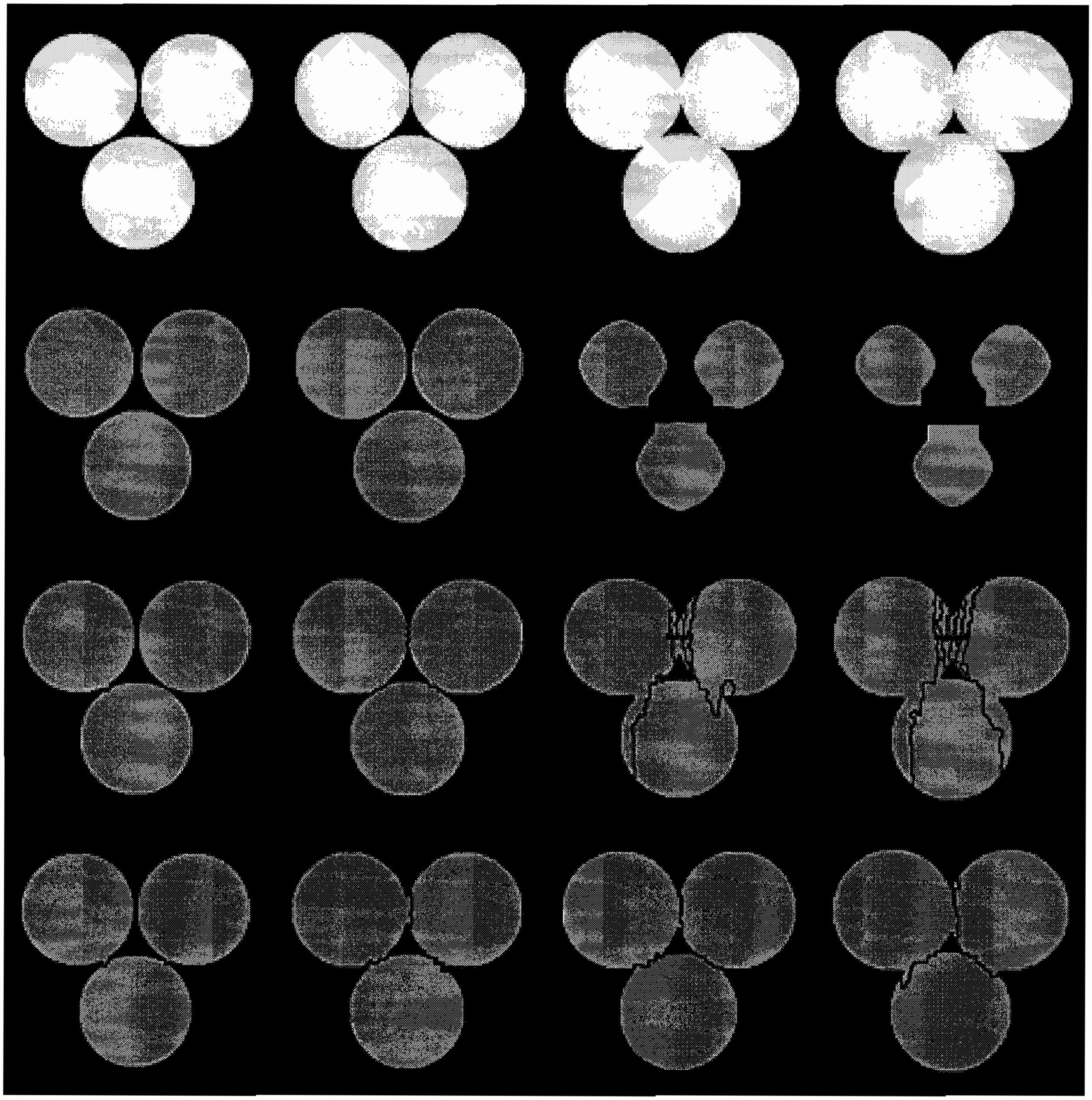

Fig. 7 Experiments on touching circles. Row 1 shows the original circles with increasing shared boundary pixels from left to right. Row 2 shows the result of using morphology erosion alone. Row 3 shows the result of using morphology erosion with restricted growing. Most object pixels were restored; but the shape were not due to excessive erosion during the separation phase. Row 4 shows the result of using probabilistic morphology with restricted growing. The circles were separated and their original size and shape were almost completely preserved.

Fig. 8 shows a similar experiment on touching squares. Table 2 shows the corresponding information and results. As we can see, our technique was able to successfully separate the two touching squares with up to 25 shared boundary pixels. This capability could only be achieved by morphological erosion alone after eleven iterations, and the distortion to the objects thereby was unrecoverable. From these two experiments, we conclude that (1) the restricted growing algorithm helps restore eroded objects to their original shape and size while preserving the established separation, as evident when we compare techniques A and B; (2) probabilistic morphology is better than conventional morphology in terms of selectively eroding shared boundary pixels, as evident when we compare techniques B and C.

Image	Original Object (pixels)	Shared Boundary (pixels)	Result A (pixels,%)	Result B (pixels,%)	Result C (pixels,%)
S90	7200	1	6728(93.44)	7199(99.99)	7199(99.99)
S85	7200	6	6272(87.11)	7187(99.82)	7187(99.82)
S80	7200	11	5408((75.11)	7112(98.78)	7178(99.69)
S75	7200	16	4232(58.78)	7058(98.03)	7172(99.61)
S70	7200	21	3528(49.00)	6999(97.23)	7161(99.46)
S66	7200	25	2888(40.11)	7004(97.28)	7157(99.40)

Table 2 The information and results of the experiment on touching squares. Each image consists of two overlapping squares. S80 for example has squares overlapping with 11 shared boundary pixels. Result A was generated by morphology erosion only; Result B by morphology erosion with restricted growing; Result C by probabilistic morphology erosion with restricted growing. Note that the last technique was able to separate two 60 x 60 squares even though the two squares shared a common boundary of 25 pixels (41.62% of each side).

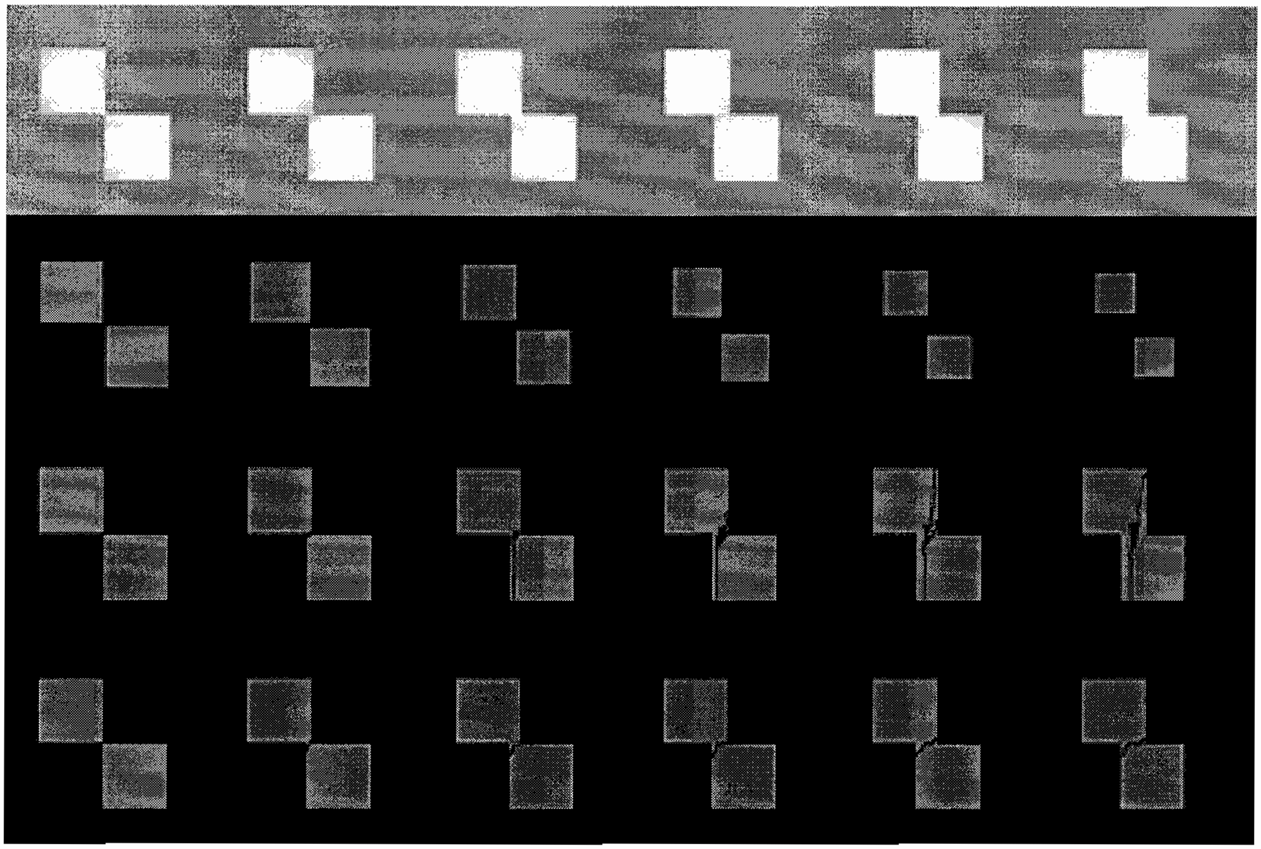

Fig. 8 Experiments on touching squares. Row 1 shows the original squares with increasing shared boundary pixels from left to right. Row 2 shows the result of using morphology erosion alone. Row 3 shows the result of using morphology erosion with restricted growing. Row 4 shows the result of using probabilistic morphology with restricted growing. The squares were separated and their original size and shape were almost completely preserved.

Fig. 9 shows the effects of noise to our techniques. Noise was additive: 5% noise means the amplitude of noise intensity is uniformly distributed within the 5% range of the image intensity. We used as much as 25% noise. The performance of all three techniques deteriorated as the noise increased. However, the probabilistic morphology with restricted growing still yielded the best results in terms of preserving the largest parts of contiguous objects in the image. Fig. 10 shows the effects of median-filtered noise to our techniques. More object pixels were preserved and the size and shape of the circles were greatly improved, compared to those in Fig. 9. For all results, the circles' outlines were roughly preserved, showing that our technique is resistant to noise as much as 15%-25%. This is because our design of the probabilistic morphology is able to look at an aggregate of successive threshold neighborhoods of a pixel (as opposed to a single threshold neighborhood) before deciding on keeping or eliminating an object pixel.

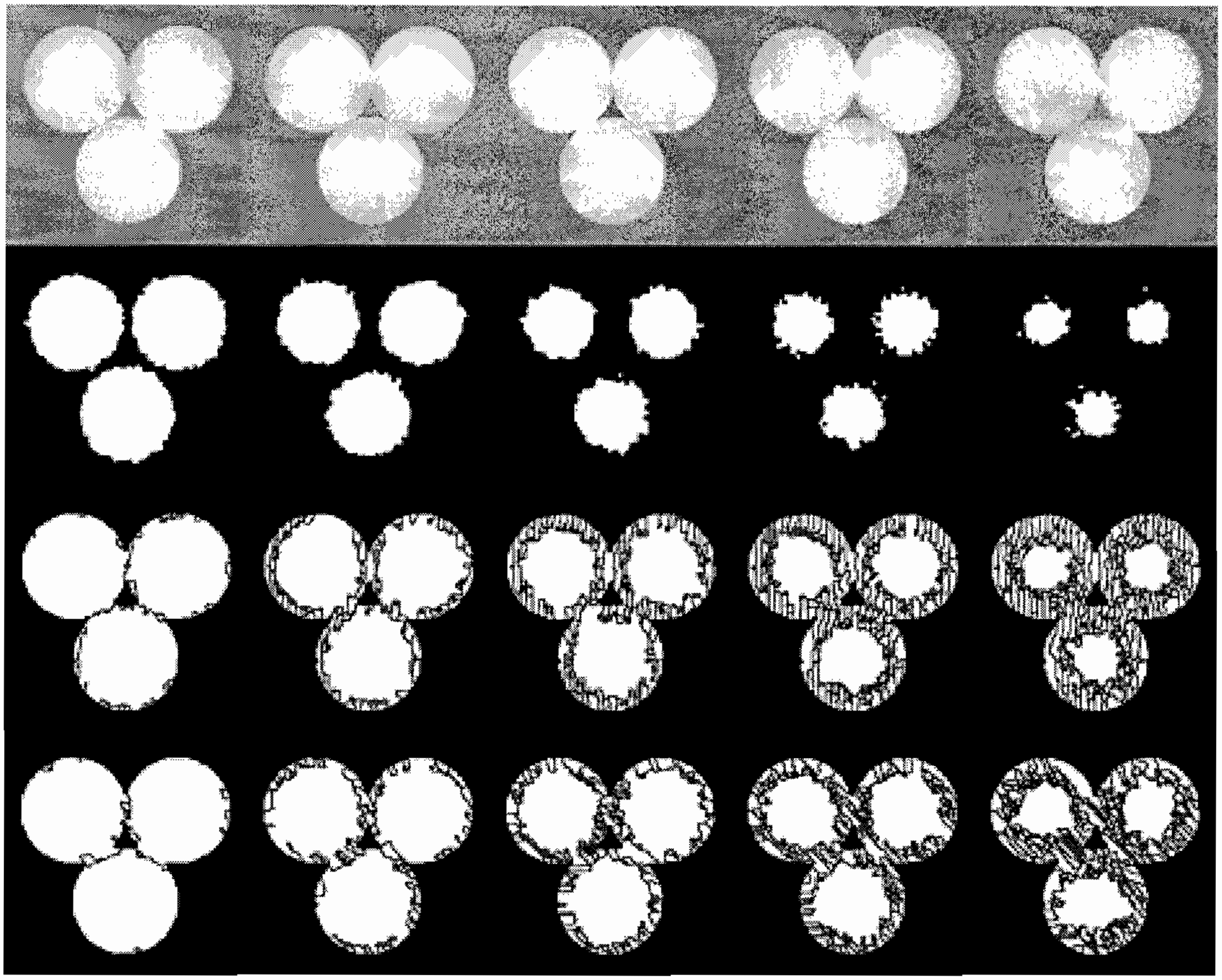

Fig. 9 Experiments on noisy objects. Row 1 shows the original images with increasing noise (5% to 25%) from left to right. Row 2 shows the result of using morphology erosion alone. Row 3 shows the result of using morphology erosion with restricted growing. Row 4 shows the result of using probabilistic morphology with restricted growing. As observed, the last technique yielded the best results. It was able to extract largest contiguous regions from the image.

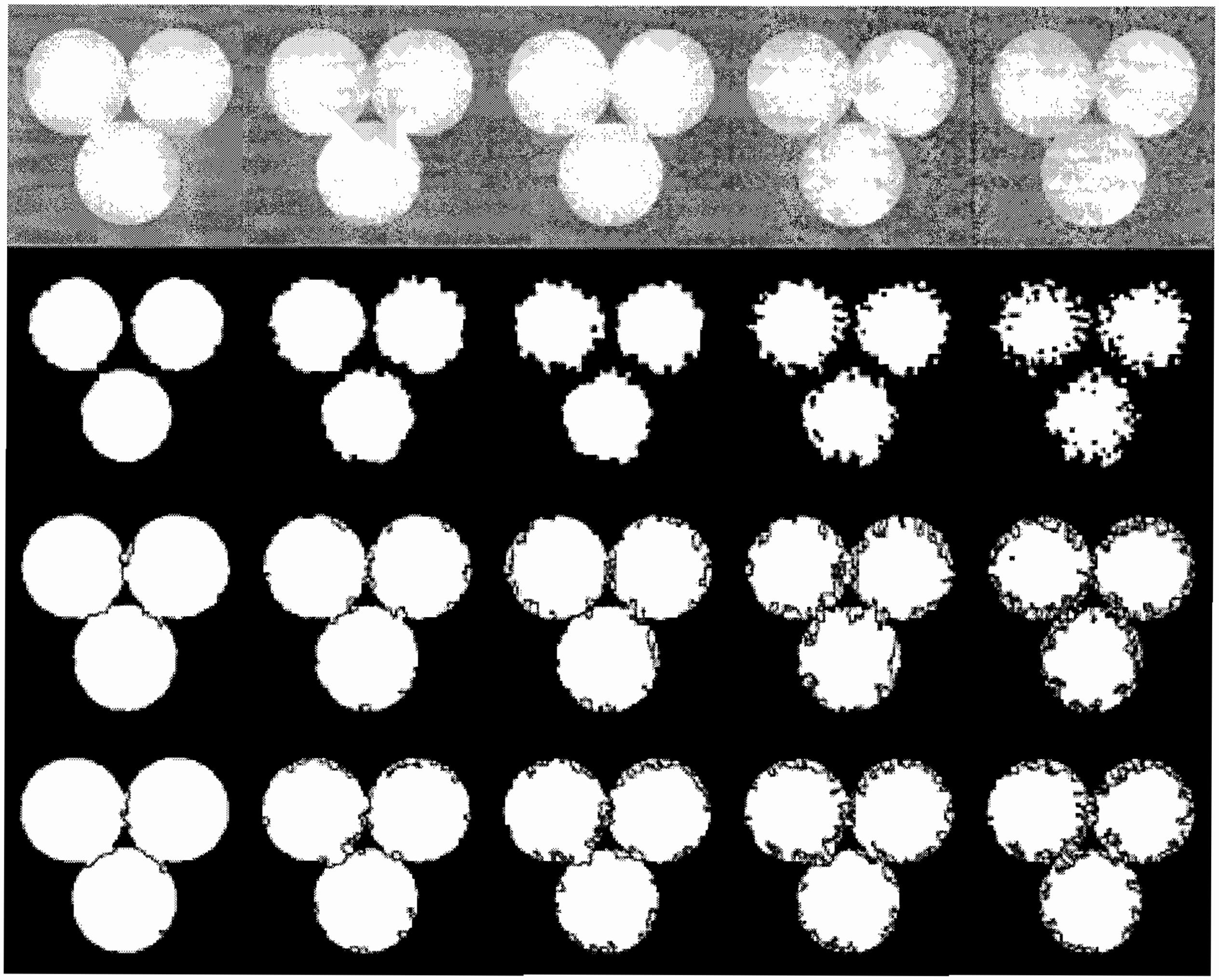

Fig. 10 Experiments on noisy objects with median filtering. Row 1 shows the original images with increasing noise (5% to 25%) from left to right, after 1 iteration of median filtering. Row 2 shows the result of using morphology erosion alone. Row 3 shows the result of using morphology erosion with restricted growing. Row 4 shows the result of using probabilistic morphology with restricted growing. As observed, the last technique was now able to preserve most of the circles in terms of shape and size.

Note that for both morphological erosion techniques, we had to run the experiments several times in order to find the least number of iterations to ensure separation among objects. As for our probabilistic morphology, no additional runs were needed.

Fig. 11 shows the flexibility of our implementation of probabilistic morphology. The original image is a photo of worms taken in a laboratory. The objective was to segment the image and identify the two separated worms. The first column displays the results of using a threshold value of 130. Note that the objects are now represented as dark pixels. Both the thresholded result and the result after applying technique C achieved separation. As we increased the threshold value, more and more object pixels were introduced to the resulting images; and thus the two worms grew closer to each other. The second column displays the results of using a threshold value of 155. As observed, our technique was still flexible enough to detect the shared boundary (14-pixel wide) between the two objects and created a successful separation. This flexibility could be attributed to the use of a range of threshold values in our implementation of probabilistic morphology.

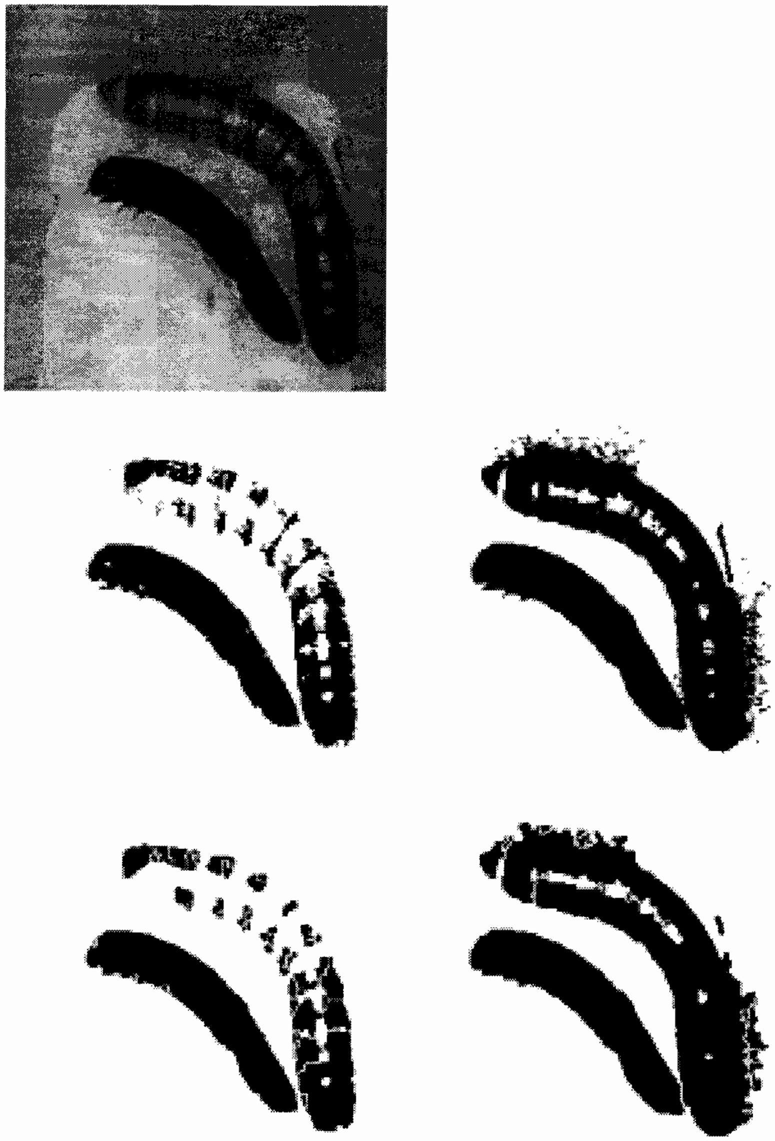

Fig. 11 Experiments on flexibility of threshold values. The raw image shows a laboratory photo of worms. Column 1 shows the results by thresholding the image at 130. Column 2 at 155. As observed, using a low threshold, the worms were clearly separated. Using a high threshold, the worms were connected (14 pixels wide at the shared boundary). The bottom row shows the results of restricted growing. Our technique was able to separate the worms in both cases.

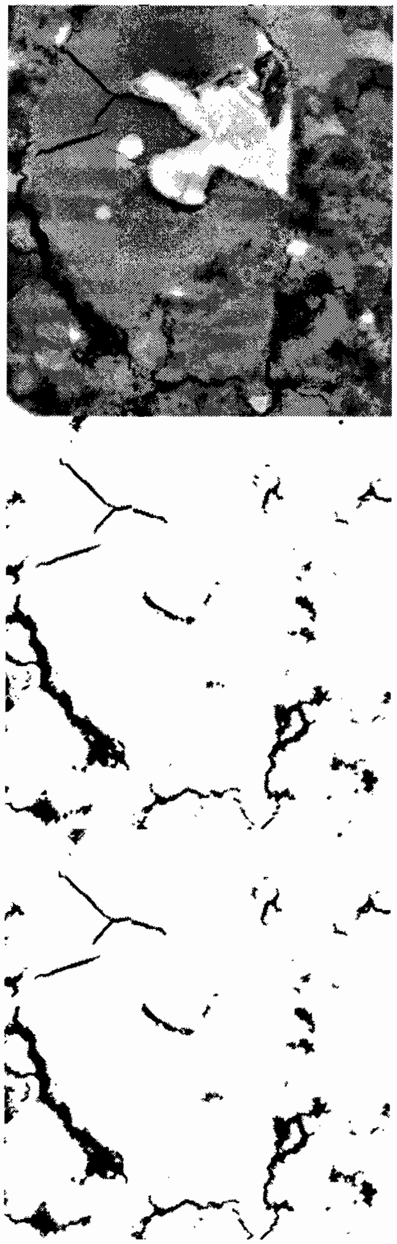

Fig. 12 Experiments on connecting objects. The raw image shows a photo of cement with cracks and other composites. The middle column shows the thresholded result. As observed, cracks (dark linear structures) were broken. By restricted growing the objects (bright regions), we increased dark regions. As a result, cracks were substantiated and connected, as shown in the right column. This allows a more accurate geometric analysis of the cracks.

While our technique is separating objects, it connects the non-objects. Thus, by reversing the roles of the two pixel classes, our technique can be employed to strengthen and link objects, as shown in Fig. 12. The original image is a photo of concrete, broken with cracks and other composites. The objective was to find the cracks to facilitate measurements such as length, angle, and branch factor for stress analysis. To process this image, we treated the cracks as non-objects, and the rest as object regions. By restricted growing the object regions, which would attempt to separate the objects as much as possible, we were actually improving the existence of non-object pixels. Compare the two segmented images. The first one was the direct result of global thresholding. The second one was after applying our technique. As we can see, cracks once broken have been linked and substantiated in size. This application shows that our technique can be used to strengthen and link broken objects.

Fig. 13 shows the results of using different skin images for our restricted growing algorithm. As mentioned in the previous section, our skin image is defined as in Eq. (20). However, this definition does destroy a tiny, unrecoverable portion of the original size and shape. Thus, we devised two alternative implementations: (1) one using the cleaned, direct thresholded result, and (2) one using the uncleaned, direct thresholded result. By cleaning, we mean applying binary morphological closing and opening to the image. The original image in Fig. 13 is a STAR-2 aircraft-SAR sea ice image. The upper-right corner shows the result of using the originally defined skin image. Details were lost, and the floes were slightly shrunk. The lower-left corner shows the result of using the cleaned, direct thresholded result as the skin image; the lower-right corresponds to that of uncleaned, direct thresholded result as the skin image. Comparing the different results visually, the last option yielded the most accurate result as small details and wiggliness of the boundaries have been completely preserved. However, since it deals with the direct thresholded result as the skin image, it also suffers from noise effects, as evident in the interior of the large ice floe. In our opinion, all three methods are applicable, depending on what the objectives of the segmentation process are: if the image is very noisy, we recommend using the skin image as defined in Eq. (20); if the details of the boundaries are important, we recommend using the direct thresholded result as the skin image; if the image is noisy and details are also important, we recommend using the cleaned thresholded result as the skin image.

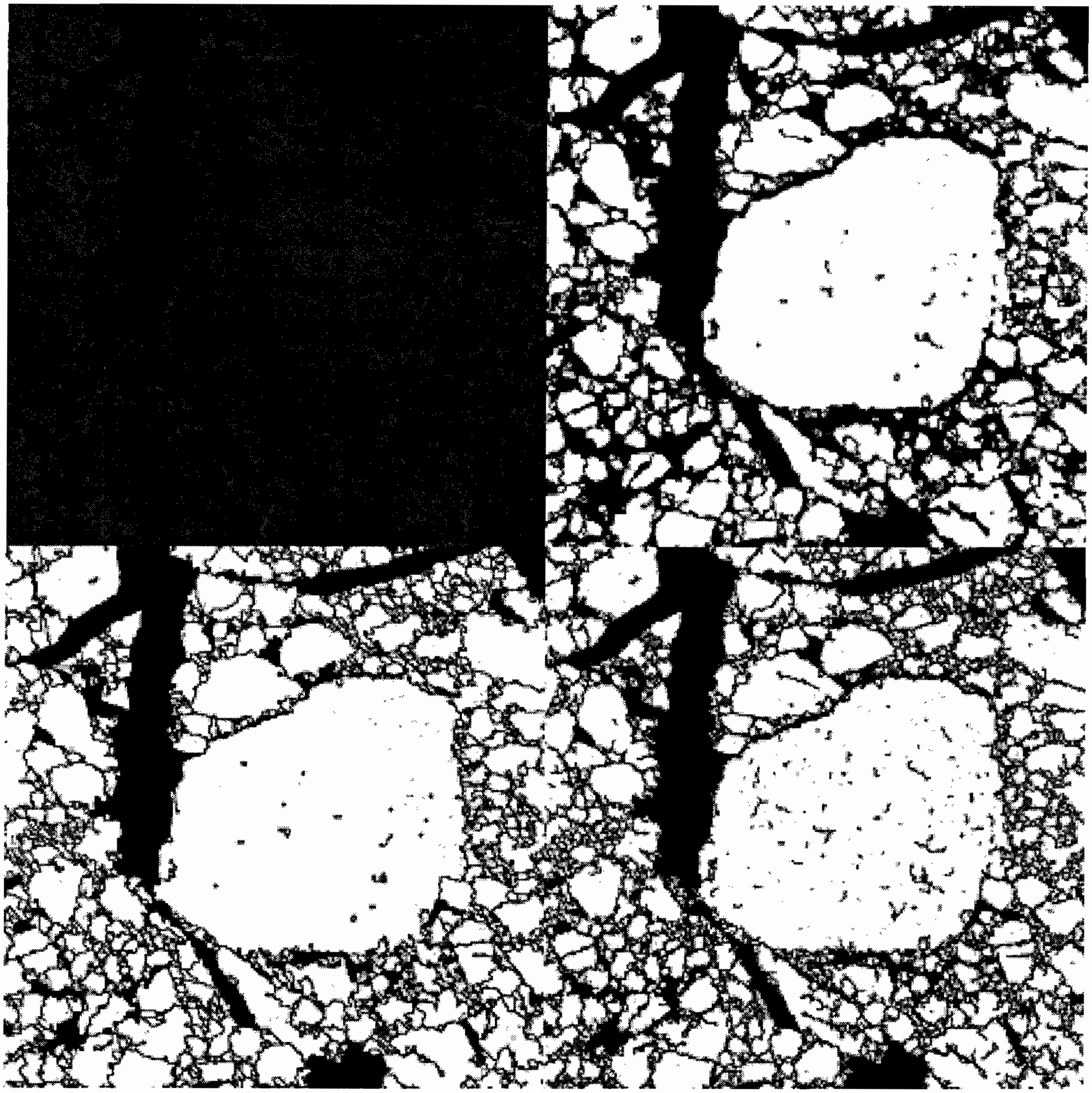

Fig. 13 Experiments on different skin image designs. The raw image shows a STAR-2 aircraft SAR sea ice image (obtained from Dr. J. Comiso of NASA Goddard SFC). The upper-right corner shows the result of using our object separation technique with the skin image as defined in Eq. (20). The lower-left corner shows the result with the morphologically cleaned, direct thresholded result of the original image as the skin image. Finally, the lower-right corner shows the result with the uncleaned, direct thresholded result of the original image as the skin image. Note that objects were separated in all three cases, but more and more small objects and tiny details at the boundaries were preserved. In the last case, it was a complete separation and preservation. In our opinion, all three designs of skin image are applicable based on specific needs of the task.

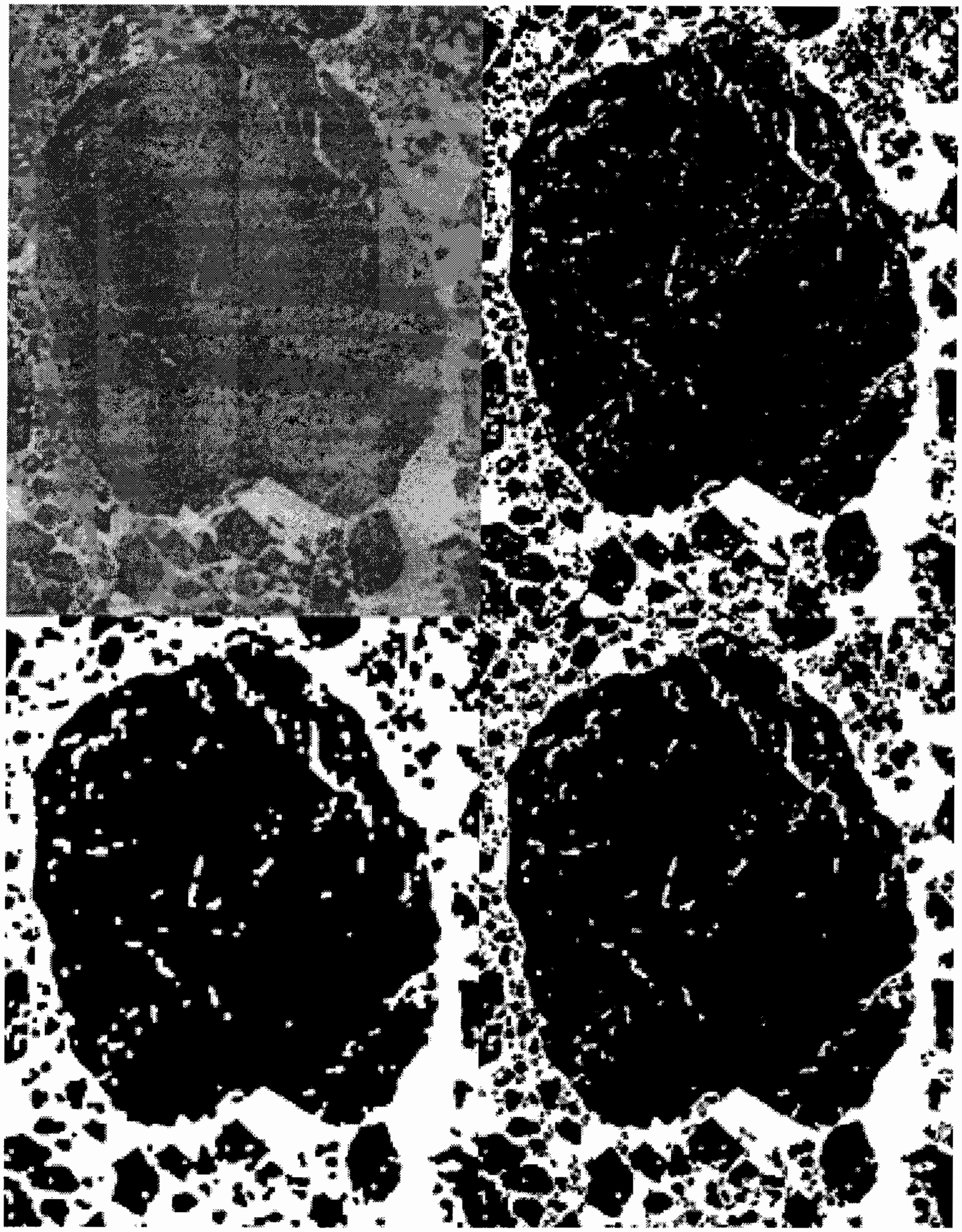

Fig. 14 Various stages of our object-separation technique. The raw image shows a portion of an ERS-1 SAR sea ice image (Copyright ESA). The upper-right corner shows the thresholded result. Note that the large piece of ice floe was connected to its neighboring floes at numerous places. The lower-left corner shows the core image. As observed, all links from the large ice floe to its neighbors were broken. Finally, the lower-right corner shows the grown image. The floes were restored to their original sizes and shapes without destroying the established separation.

Fig. 14 shows a succession of images at different stages of our technique. The original image was a part of an ERS-1 Satellite Synthetic Aperture Radar (SAR) imagery. The upper-right corner shows the result of thresholding the image. Note that the large piece of ice floe (bright region) is connected to its surrounding smaller floes, including the one to its south-east corner. The lower-left corner shows the core image. As we can see, that large piece of ice floe is now a core object and separated from its neighbors. Finally, the lower-right corner shows the result after the application of restricted growing on the core image. Analyzing the image carefully, one can observe that the floe has been restored to its full size and shape while its separation from its neighbors has been established.

Note that there are single-pixel linear artifacts introduced to the images, as observed in Fig. 11. We call these artifacts *hairlines*. Hairlines are results of two possible causes. In the following discussion, we assume that objects are brighter than the background. First, a hairline may be introduced if two non-object neighborhoods are in close proximity with each other. The *attraction* between these two regions allows the pixels in between them to be under the influence of non-object pixels, resulting in a breach in the channel that initially separates the two regions. Second, suppose a object has regions of darkness within itself, which we refer to as a *weak object*. Then, an adjacent, non-object neighborhood may be able to induce those regions to become hairlines, usually in shattery forms. If we view hairlines as errors, then before we apply restricted region growing, we should pre-process the image to remove noise, which in turn reducing the number of hairlines. We can also appply other types of smoothing filter to solidify weak objects. From another viewpoint, a hairline is a version of a locally best-search path from a non-object to another. If the cost of the path is the sum of intensity levels of pixels and neighborhoods between two non-objects, then the restricted growing algorithm will find the path that consists of pixels with low intensity levels and neighborhoods. Thus, our algorithm searches for dark pixels to signify separations among regions even though such separations might be caused by noise effects.

7. DISCUSSION AND CONCLUSIONS

The restricted growing concept consists of three main procedures: generating the skin objects, obtaining the core objects, and restricted growing, and all can be implemented differently from what we have presented here. In addition to the three alternatives of generating the skin image that we have provided, one can experiment with other types of segmentation process as long as it yields objects with preserved shapes and sizes. In our opinion, approaches, such as relaxation [6][14][27] and multiple resolution [3][36], that are able to erode an object based on its boundary intensities can be used to obtain core objects Moreover, spatial and textural statistics can also be used instead of probability of intensities in the neighborhood for determining the status of each pixel as a core or skin pixel. Similarly, one can design a different restricted growing algorithm as long as it grows core objects within the boundary of skin objects and does not destroy the existing separation during the process. Instead of sequentially, for example, one implementation may grow each object outward simultaneously ensuring consistent restoration of shape and size. A host of thinning algorithms designed for different applications [18][20] or faster computational speed [1][23][24] can be modified to improve the convergence rate of the growing process. Because of the generality and modularity of the restricted growing concept, one can design his or her own object separation algorithm to suit specific applications and imagery.

In conclusion, we have presented a concept that is able to achieve both object separation and preservation of size and shape of objects. This concept is both intuitive and general. It solves the problem of object separation by addressing its two most important issues: separation among objects and preservation of objects. The restricted growing concept is a framework with which scientists can design image processing techniques tailored to specific domains and applications such as pattern recognition, feature extraction, object solidification, segmentation, and other tasks. Our implementation which includes probabilistic morphology and a restricted growing algorithm outperformed the techniques based on traditional binary morphology. In addition, our implementation is able to handle different types of imagery and applications without any parameter adjustments after they have been determined. On the other hand, the parametric design allows flexibility in adapting to diverse image domains as well.

We have also examined our technique's performance along criteria such as preservation of small objects and breaking of strong connections. Our experiments showed that our technique was able to retain small objects which would have been sacrificed for separation in other conventional approaches. Moreover, our technique was able to break strong connections in one fully-automated iteration. This allows batch running of the program without having to manually determine the optimal number of iterations as in morphology-based erosion. Also, coupled with a noise-removal filter (e.g. median filter), our technique was able to handle noise as much as 25% acceptably.

REFERENCES

[1] C. Arcelli and G. Sanniti di Baja, A Width-Independent Fast Thinning Algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 463–474, 1985.

[2] J. Banfield and A. E. Raftery, Ice Floe Identification in Satellite Images Using Mathematical Morphology and Clustering about Principal Curves, *Journal of the American Statistical Association*, vol. 87, no. 417, pp. 7–16, 1992.

[3] F. Bergholm, Edge Focusing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 6, pp. 726–741, 1987.

[4] H. Blum, "A Transformation for Extracting New Descriptors of Shape," in *Models for the Perception of Speech and Visual Form*, Wathen-Dunn, W., ed., Cambridge, MA: MIT Press, 1967.

[5] J. Daida, R. Samadani, and J. F. Vesecky, "Object-Oriented Feature-Tracking Algorithms for SAR Images of the Marginal Ice Zone," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 4, pp. 573–589, 1990.

[6] A. J. Danker and A. Rosenfeld, "Blob Detection by Relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 1, pp. 79–92, 1981.

[7] R. C. Gonzalez, Industrial Computer Vision, in *Advances in Information Systems Science* (J. T. Tou, Ed.), pp.345–385, Plenum, New York, 1985.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1992.

[9] R. M. Haralick, S. R. Sternberg, and X. Zhuang, Image Analysis Using Mathematical Morphology, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, 1987.

[10] T. Hastie and W. X. Stuetzle, Principal Curves, *Journal of the American Statistical Association*, vol. 84, pp. 502–516, 1989.

[11] D. Haverkamp, L.-K. Soh, and C. Tsatsoulis, A Comprehensive, Automated Approach to Determining Sea Ice Thickness from SAR Data, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 1, pp. 46-57, 1995.

[12] H. J. A. M. Heijmans and C. Ronse, The Algebraic Basis of Mathematical Morphology I. Dilations and Erosions, *Computer Vision, Graphics, and Image Processing*, vol. 50, no. 3, pp. 245-295, 1990.

[13] C. J. Hilditch, Comparison of Thinning Algorithms on A Parallel Processor, *Image and Vision Computing*, vol. 1, no. 3, pp. 115–132, 1983.

[14] R. A. Hummel and S. W. Zucker, On the Foundations of Relaxation Labeling Processes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 3, pp. 267–287, 1983.

[15] B.-J. Jang and R. T. Chin, Analysis of Thinning Algorithms Using Mathematical Morphology, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 541–551, 1990.

[16] L. Ji and J. Piper, Fast Homotopy-Preserving Skeletons Using Mathematical Morphology, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 6, pp. 653–664, 1992.

[17] L. Lam, S. W. Lee, and C. Y. Suen, Thinning Methodologies—A Comprehensive Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, 1992.

[18] L. Lam and C. Y. Suen, An Evaluation of Parallel Thinning Algorithms for Character Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 914–919, 1995.

[19] P. A. Maragos and R. W. Schafer, Morphological Skeleton Representation and Coding of Binary Images, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1228–1244, 1986.

[20] P. A. Maragos and R. D. Ziff, Threshold Superposition in Morphological Image Analysis Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 498–504, 1990.

[21] G. Matheron, *Random Sets and Integral Geometry*, Wiley, New York, 1975.

[22] D. M. McKeown, W. A. Harvey, and J. McDermott, Rule-Based Interpretation of Aerial Imagery, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 5, 1985, pp. 570–585, 1985.

[23] L. O'Gorman, $k \times k$ Thinning, *Computer Vision, Graphics, and Image Processing*, vol. 51, no. 2, 1990, pp. 195–215, 1990.

[24] J. Piper, Efficient Implementation of Skeletonization Using Interval Coding, *Pattern Recognition Letter*, vol. 3, pp. 389–397, 1985.

[25] C. Ronse and H. J. A. M. Heijmans, The Algebraic Basis of Mathematical Morphology II. Openings and Closings, *Computer Vision, Graphics, and Image Processing*, vol. 54, no. 1, pp. 74–97, 1991.

[26] A. Rosenfeld, Connectivity in Digital Pictures, *Journal of ACM*, vol. 17, no. 1, pp. 146–160, 1970.

[27] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, Scene Labelling by Relaxation Operations, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 420–433, 1976.

[28] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[29] J. Serra, Introduction to Mathematical Morphology, *Computer Vision, Graphics, and Image Processing*, vol. 35, pp. 283–305, 1986.

[30] F. Y.-C. Shih and O. R. Mitchell, Threshold Decomposition of Gray-Scale Morphology into Binary Morphology, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 31–42, 1989.

[31] L.-K. Soh and C. Tsatsoulis, A Multistage Feature Extraction Technique for Synthetic Aperture Radar (SAR) Sea Ice Imagery, Part I: Combination of Algorithmic and Heuristic Methods, Technical Report, *CECASE TR 9710-02*, The University of Kansas, Lawrence, Kansas, 1993.

[32] L.-K. Soh, C. Tsatsoulis, and B. Holt, Identifying Ice Floes and Computing Ice Floe Distributions in SAR Images, accepted for publication in *Recent Advances in the Analysis of SAR Sea Ice Data*, Springer-Verlag, to appear in 1998.

[33] J. Song and E. J. Delp, The Analysis of Morphological Filters with Multiple Structuring Elements, *Computer Vision, Graphics, and Image Processing*, vol. 50, no. 3, 1990, pp. 308–328, 1990.

[34] S. R. Sternberg, Grayscale Morphology, *Computer Vision, Graphics, and Image Processing*, vol. 35, 1986, pp. 333–355, 1986.

[35] I. D. Svalbe, The Geometry of Basis Sets for Morphologic Closing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 12, pp. 1214–1224, 1991.

[36] D. Terzopoulos, Image Analysis Using Multigrid Relaxation Methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 2, pp. 129–139, 1986.

[37] T. Y. Zhang and C. Y. Suen, A Fast Parallel Algorithm for Thinning Digital Patterns, *Communications of ACM*. vol. 27, no. 3, pp. 236–239, 1984.

[38] S. W. Zucker, Region Growing: Childhood and Adolescence, *Computer Graphics and Image Processing*, vol. 5, pp. 382–399, 1976.