SEARCH, DEPTH-FIRST

State-space search methods are useful when a problem can be formulated in terms of finding a solution path in a directed graph from an initial node to a goal node. State-space graphs are implicitly represented. They are generated on the fly with the aid of a successor-generator function; given a node of the graph, this function generates its successors. Depth-first search is a name commonly used for various search methods that perform search as follows. The search begins by expanding the initial node, ie, by generating its successors. At each later step, one of the most recently generated nodes is expanded. (In some problems, heuristic information is used to order the successors of an expanded node. This determines the order in which these successors will be visited by the depth-first search method.) If this most recently generated node does not have any successors or if it can be determined that the node will not lead to any solutions, then backtracking (qv) is done, and a most recently generated node from the remaining as yet unexpanded nodes is selected for expansion.

A depth-first search method can be used to find a solution in the search space by simply terminating the algorithm when the first solution is found. It can also be used to find a least-cost solution (by letting the algorithm run until the whole search space is exhausted, and also by keeping track of the best solution seen so far). Following are three search methods that use the depth-first search strategy.

1. Simple backtracking is a depth-first search method that is used to find any one solution and that uses no heuristics for ordering the successors of an expanded node. Heuristics may be used to prune nodes of the search space so that search can be avoided under these nodes.
2. Ordered depth-first search is a depth-first search method that is used to find any one solution and that uses heuristics for ordering the successors oí an expanded node. Heuristics may also be used to prune nodes of the search space so that search can be avoided under these nodes.
3. Depth-first branch-and-bound (DFBB) is a depth-first search method that is used to find an optimal solution. These search methods use a lower bound function (defined over the nodes of the search space) to prune those nodes that cannot lead to a solution that is better than the one already found. They also often use a heuristic to order the successors of an expanded node.

There is a considerable confusion regarding the names that are used by various researchers to label different search techniques. For example, the names depth-first branch-and-bound and backtracking are used by some researchers to refer to the class of algorithms that here are called ordered depth-first search.

If the search space to the left of the first goal node is infinite (or very large), then search would never terminate (or take a very long time). This problem can be corrected by having a bound L on the depth of the space searched. This kind of search is called depth-bounded depth-first search. If there is no goal node at a depth L or earlier, then the search would fail even if there is a goal node at a depth greater than L. In such cases the search will have to be restarted with a larger depth bound.

Usually, a depth-first search procedure has lower storage requirement than a best-first search procedure. If every node has k successors, then the storage requirement of a depth-first procedure for searching to a depth of n is $O(n \times k)$. In best-first search, if the heuristic evaluation function is bad, then the storage requirement can be as much as $O(k^n)$. Furthermore, depth-first search has very little overhead as compared to best-first search in which a priority queue must be rearranged after every node expansion.

BIBLIOGRAPHY

General References

E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms,* Computer Science Press, Rockville, Md., 1978.

L. Kanal and V. Kumar, eds., *Search in Artificial Intelligence.* Springer-Verlag, New York, 1988.

N. J. Nilsson, *Principles of Artificial Intelligence,* Tioga Press, Palo Alto, Calif., 1980.

J. Pearl. *Heuristics—Intelligent Search Strategies for Computer Problem Solving.* Addison-Wesley, Reading, Mass., 1984.

V. Kumar
University of Minnesota

SEGMENTATION

Image segmentation is the partition of an image into a set of nonoverlapping regions whose union is the entire image. The purpose of image segmentation is to decompose the image into parts that are meaningful with respect to a particular application. For example, in two-dimensional part recognition, a segmentation might be performed to separate the two-dimensional object from the background. Figure 1a shows a gray-level image of an industrial part, and Figure 1b shows its segmentation into object and background. In this figure, the object is shown in white and the background in black. In simple segmentations, this article will use gray levels to illustrate the separate regions. In more complex segmentation examples where there are many regions, white lines on a black background will be used to show the separation of the image into its parts.

It is very difficult to tell a computer program what constitutes a meaningful segmentation. Instead, general segmentation procedures tend to obey the following rules.

1. Regions of an image segmentation should be uniform and homogenous with respect to some characteristic such as gray level or texture.
2. Region interiors should be simple and without many small holes.
3. Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform.
4. Boundaries of each segment should be simple, not ragged, and must be spatially accurate.

Reprinted from Encyclopedia of Artificial Intelligence, Second Edition,

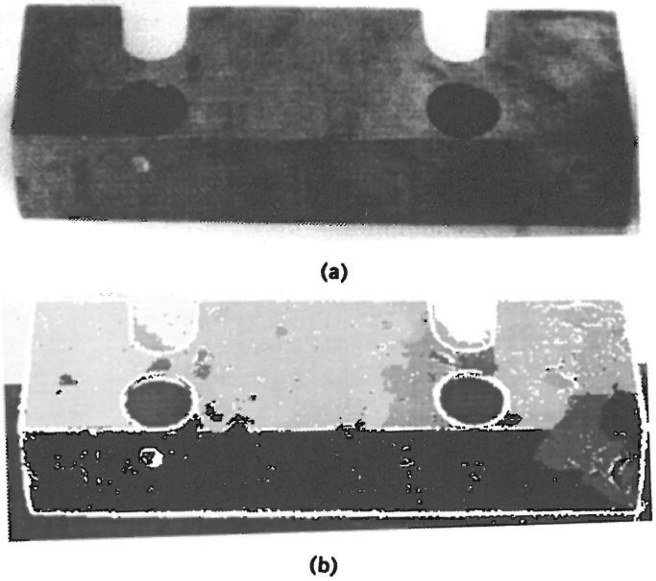


Figure 1. (*a*) A gray-level image of an industrial part and (*b*) a segmentation of the image into object (white) and background (black).

Achieving all these desired properties is difficult because strictly uniform and homogeneous regions are typically full of small holes and have ragged boundaries. Insisting that adjacent regions have large differences in values can cause regions to merge and boundaries to be lost.

Clustering in pattern recognition (qv) is the process of partitioning a set of pattern vectors into subsets called clusters (Young and Calvert, 1974). For example, if the pattern vectors are pairs of real numbers illustrated by the point plot of Figure 2, clustering consists of finding subsets of points that are close to each other in Euclidean two-space. As there is no full theory of clustering, there is no full theory of image segmentation. Image segmentation techniques are basically *ad hoc* and differ precisely in the way they emphasize one or more of the desired properties and in the way they balance and compromise one desired property against another. The difference between image segmentation and clustering is that in clustering, the grouping is done in measurement space. In image segmentation, the grouping is done on the spatial domain of the image and there is an interplay in the clustering between the (possibly overlapping) groups in measurement space and the mutually exclusive groups of the image segmentation.

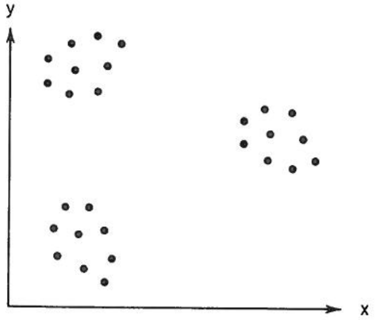


Figure 2. A set of points in a Euclidean measurement space that can be separated into three clusters of points. Each cluster consists of points that are in some sense close to each other.

This article describes the main ideas behind the major image segmentation techniques and gives example results for a number of them. Additional image segmentation surveys have been published (Zucker, 1976; Riseman and Arbib, 1977; Kanade, 1980; Fu and Mui, 1981). This article will view segmentation with respect to the gray-level characteristic. Segmentation on the basis of some other characteristic, such as texture, can be achieved by first applying an operator that transforms local texture to a texture feature value (see also TEXTURE). Texture segmentation can then be accomplished by applying segmentation with respect to the texture pattern value characteristic exactly as if it were a gray-level characteristic.

MEASUREMENT–SPACE GUIDED SPATIAL CLUSTERING

This technique for image segmentation uses the measurement–space clustering process to define a partition in measurement–space. Then each pixel is assigned the label of the cell in the measurement–space partition to which it belongs. The image segments are defined as the connected components of the pixels having the same label.

The segmentation process is, in general, an unsupervised clustering, because no *a priori* knowledge about the number and type of regions present in the image is available. The accuracy of the measurement–space clustering image segmentation process depends directly on how well the objects of interest on the image separate into distinct measurement–space clusters. Typically, the process works well in situations where there are a few kinds of distinct objects having widely different gray-level intensities (or gray-level intensity vectors, for multiband images) and these objects appear on a near uniform background.

Clustering procedures that use the pixel as a unit and compare each pixel value with every other pixel value can require excessively large computation time because of the large number of pixels in an image. Iterative partition rearrangement schemes must go through the image data set many times and if done without sampling can also take excessive computation time. Histogram mode seeking, because it requires only one pass through the data, probably involves the least computation time of the measurement–space clustering techniques, and it is the approach discussed here.

Histogram mode seeking is a measurement–space clustering process in which it is assumed that homogeneous objects on the image manifest themselves as the clusters in measurement–space. Image segmentation is accomplished by mapping the clusters back to the image domain where the maximal connected components of the mapped back clusters constitute the image segments. For images that are single band images, calculation of this histogram in an array is direct. The measurement–space clustering can be accomplished by determining the valleys in this histogram and declaring the clusters to be the interval of values between valleys. A pixel whose value is in the ith interval is labeled with index i and the segment it belongs to is one of the connected components of all pixels whose label is i. Thresholding techniques are examples of histogram mode seeking with bimodal histograms.

Figure 3 illustrates an example image that is the right kind of image for the measurement–space clustering image segmentation process. It is an enlarged image of a polished mineral ore section. The width of the field is about 1 mm. The ore is from Ducktown, Tennessee, and shows subhedral to enhedral pyrite porophyroblests (white) in a matrix of pyrorhotite (gray). The black areas are holes. Figure 4 shows the histogram of this image. The valleys are no trouble to find. The first cluster is from the left end to the first valley. The second cluster is from the first valley to the second valley. The third cluster is from the second valley to the right end. Assigning to each pixel the cluster index of the cluster to which it belongs and then assigning a unique gray level to each cluster label yields the segmentation shown in Figure 5. This is a virtually perfect (meaningful) segmentation.

Figure 6 shows an example image that is not ideal for measurement–space clustering image segmentation. Figure 7 shows its histogram, which has three modes and two

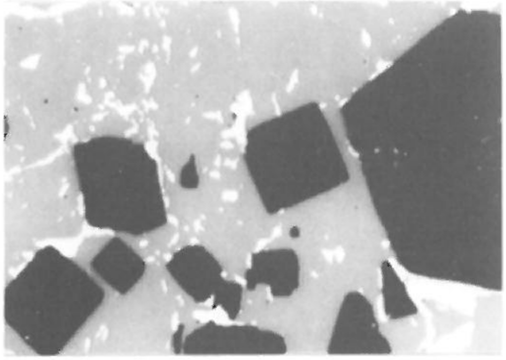


Figure 3. An enlarged raw mineral ore section. The bright areas are grains of pyrite; the gray areas constitute a matrix of pyrorhotite; the black areas are holes.

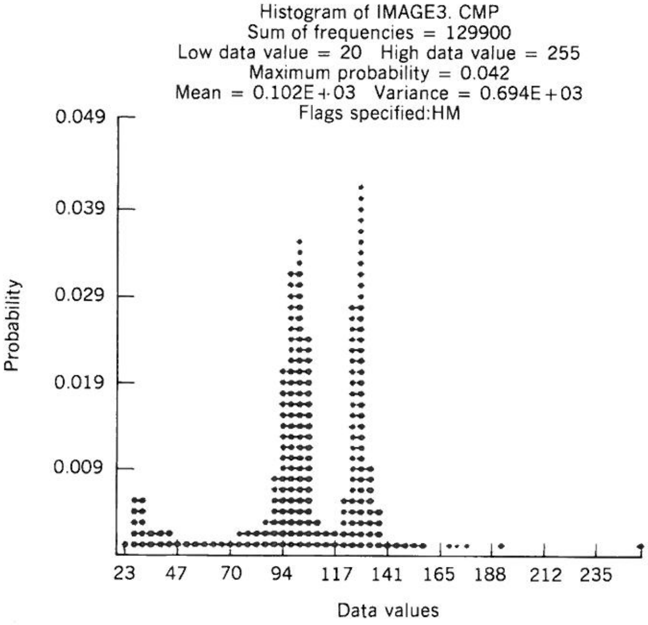


Figure 4. The histogram of the image in Figure 3. The three nonoverlapping modes correspond to the black holes, the pyrorhotite, and the pyrite.

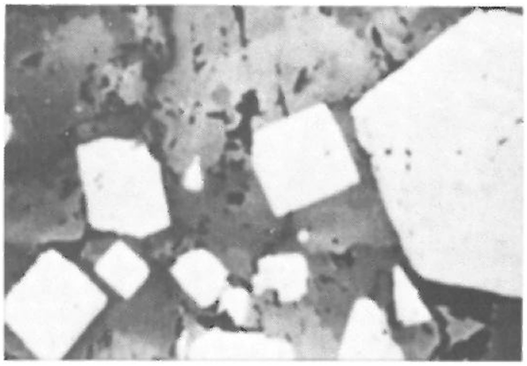


Figure 5. The segmentation of the image of Figure 3, produced by clustering the histogram of Figure 4.

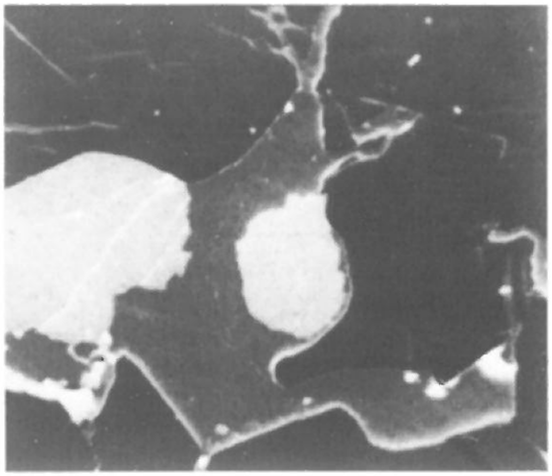


Figure 6. An image similar in some respects to the image of Figure 3. Because some of the boundaries between regions are shadowed, homogeneous region segmentation may not produce the desired segmentation.

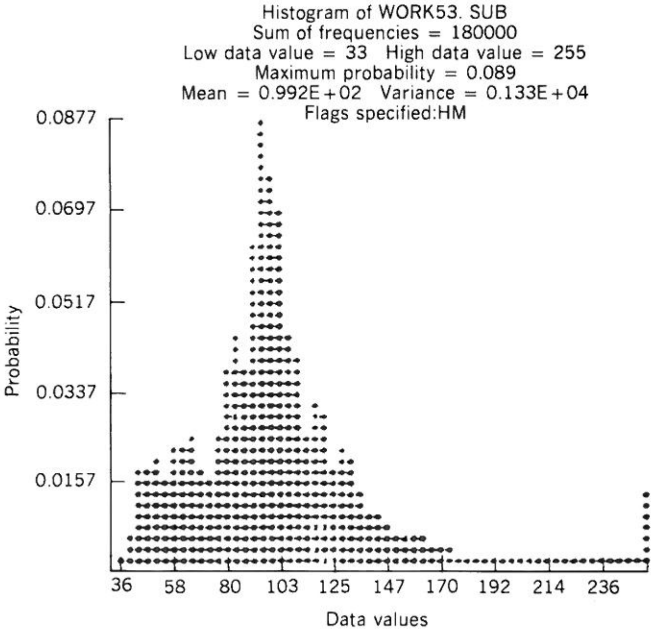


Figure 7. A histogram of the image of Figure 6.

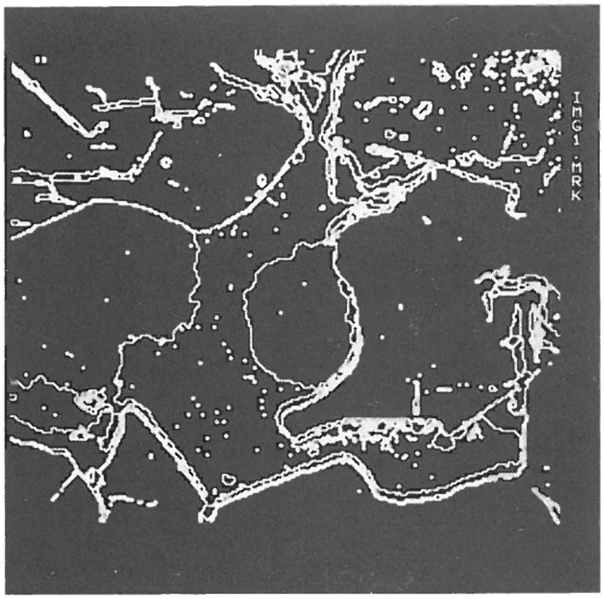


Figure 8. The segmentation of the image of Figure 6, produced by clustering the histogram of Figure 7.

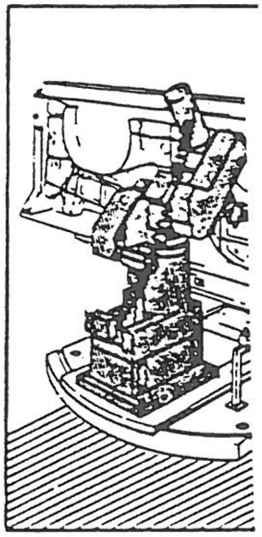


Figure 9. An F-15 bulkhead.

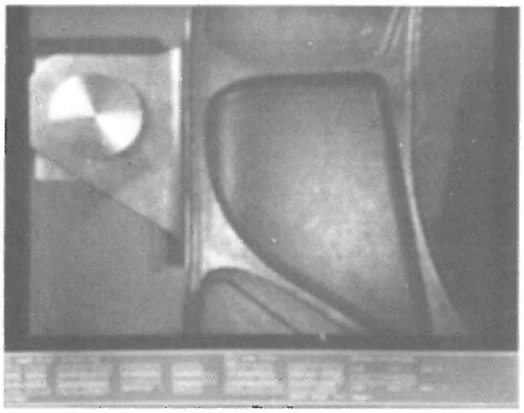


Figure 10. A section of the F-15 bulkhead.

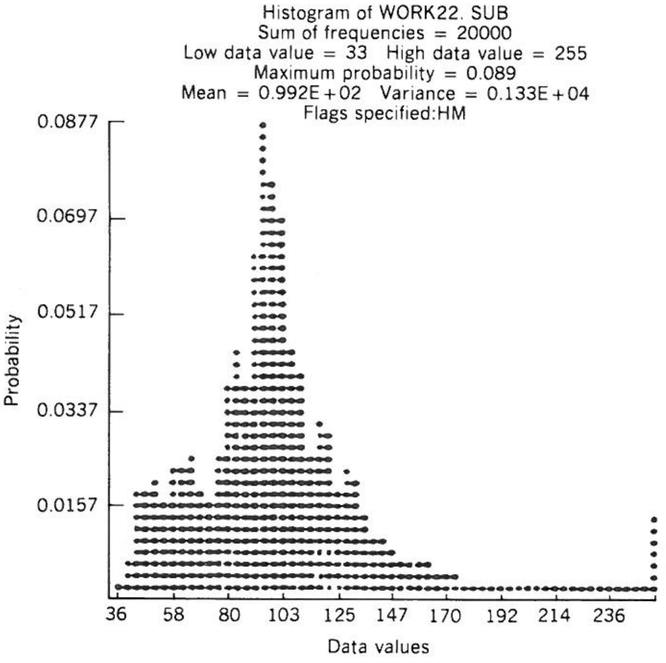


Figure 11. A histogram of the bulkhead image of Figure 10.

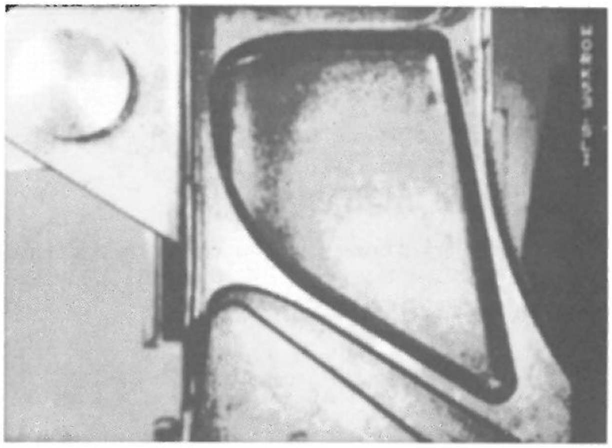


Figure 12. The segmentation of the bulkhead by a measurement–space clustering into five clusters.

valleys, and Figure 8 shows the corresponding segmentation. Notice the multiple boundary area. It is apparent that the boundary between the grain and background is in fact shaded dark, and there are many such border regions that show up as dark segments. In this case, it is not desired that the edge borders be separate regions, and although the segmentation procedure did exactly as it should have done, the results are not what was desired. This illustrates that segmentation into homogeneous regions is not necessarily a good solution to a segmentation problem.

The next example further illustrates the fallacies of measurement–space clustering. Figure 9 is a diagram of an F-15 bulkhead. Images of portions of the bulkhead, which were used as test data for an experimental robot guidance–inspection system, will be used as examples throughout the rest of this article. Figure 10 illustrates an image of a section of the F-15 bulkhead. It is clear that the image has distinct parts such as webs and ribs. Figure 11 shows the histogram of this image. It has two well-separated modes. The narrow one on the right, with a long left tail, corresponds to specular reflection points. The main mode has three valleys on its left side and two valleys on its right side. Defining the depth of a valley to be the probability difference between the valley bottom and the lowest valley side and eliminating the two shallowest valleys produces the segmentation shown in Figure 12. The problem in the segmentation is apparent. Because the clustering was done in measurement space, there was no requirement for good spatial continuation and the resulting boundaries are very noisy and busy. Separating the main mode into its two most dominant submodes produces the segmentation of Figure 13. Here the boundary noise is less, the resulting regions more satisfactory, but the detail provided is much less.

Ohlander and co-workers (1978) refine the clustering idea in a recursive way. They begin by defining a mask selecting all pixels on the image. Given any mask, a histogram of the masked image is computed. Measurement–space clustering enables the separation of one mode of the histogram set from another mode. Pixels on the image are then identified with the cluster to which they belong. If there is only one measurement–space cluster, then the mask is terminated. If there is more than one cluster, then each connected component of all pixels with the same cluster is, in turn, used to generate a mask that is placed on a mask stack. During successive iterations the next mask in the stack selects pixels in the histogram computation process. Clustering is repeated for each new mask until the stack is empty.

Figure 14 illustrates this process, which is called a recursive histogram-directed spatial clustering. Figure 15 illustrates a recursive histogram-directed spatial clustering technique applied to the bulkhead image of Figure 10. It produces a result with boundaries being somewhat busy and many small regions in areas of specular reflectance. Figure 16 illustrates the results of performing a morphological opening with a 3 × 3 square structuring element on the segmentation of Figure 15. The tiny regions are removed in this manner, but several important long, thin regions are also lost.

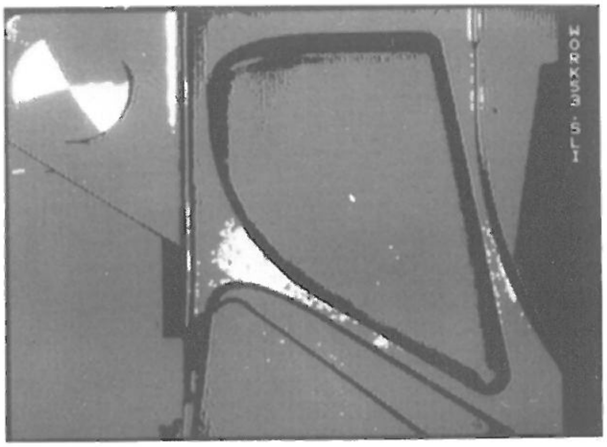


Figure 13. The segmentation of the bulkhead, induced by a measurement–space clustering into three clusters.

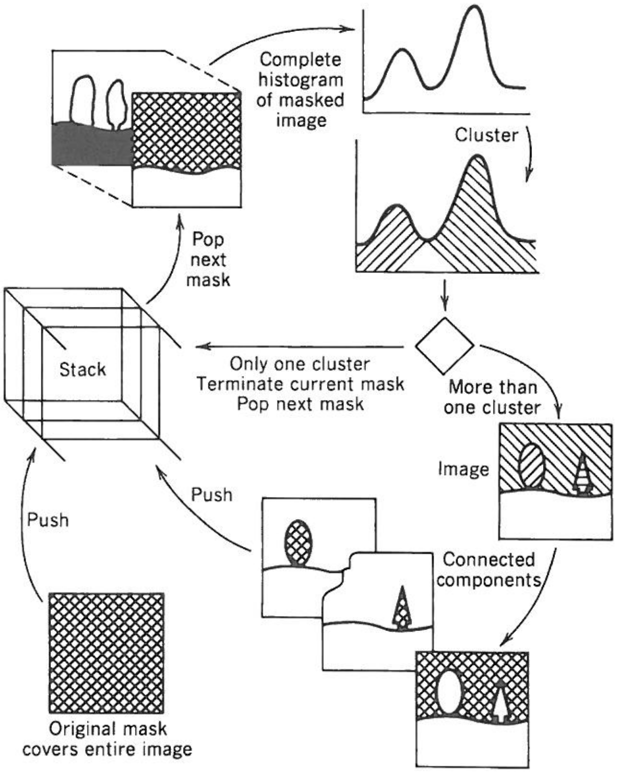


Figure 14. The recursive histogram-directed spatial clustering scheme of Ohlander and co-workers (1978).

For ordinary color images, Ohta and co-workers (1980) suggest that histograms not be computed individually on the red, green, and blue (RGB) color variables, but on a set of variables closer to what the Karhunen-Loeve (principal components) transform would suggest. They suggest $(R + G + B)/3$, $(R - B)/2$, and $(2G - R - B)/4$. Figure 17 illustrates a color image. Figure 18 shows two segmentations of the color image: one by recursive histogram-

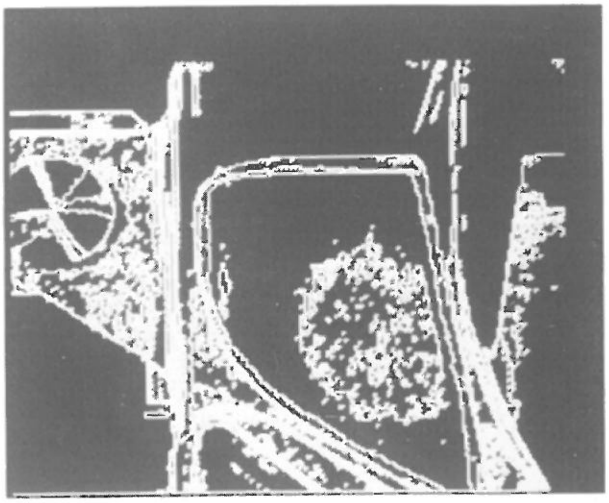


Figure 15. The results of the histogram-directed spatial clustering when applied to the bulkhead image.

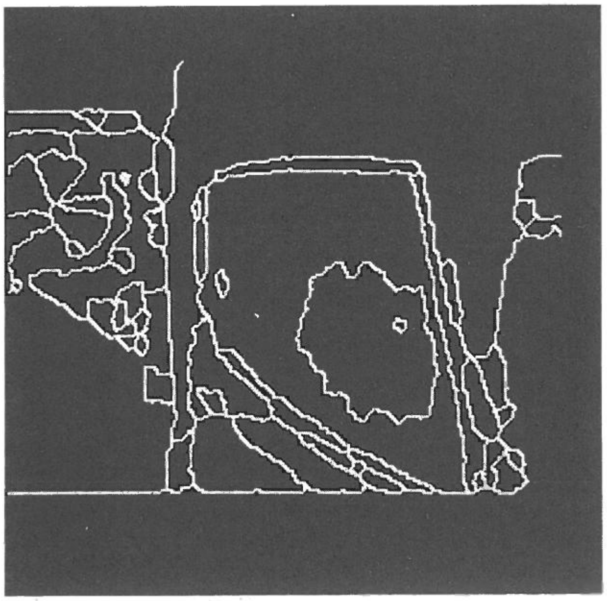


Figure 16. The results of performing a morphological opening with a 3 × 3 square structuring element on the segmentation of Figure 15.

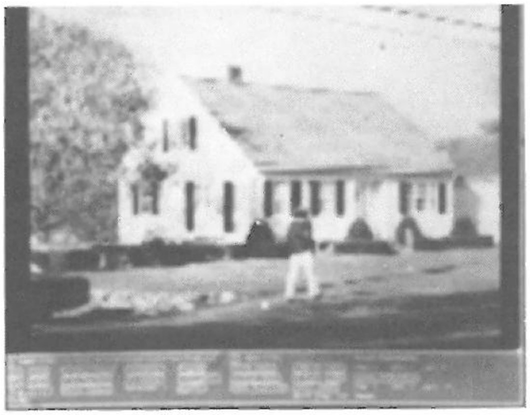


Figure 17. A color image.

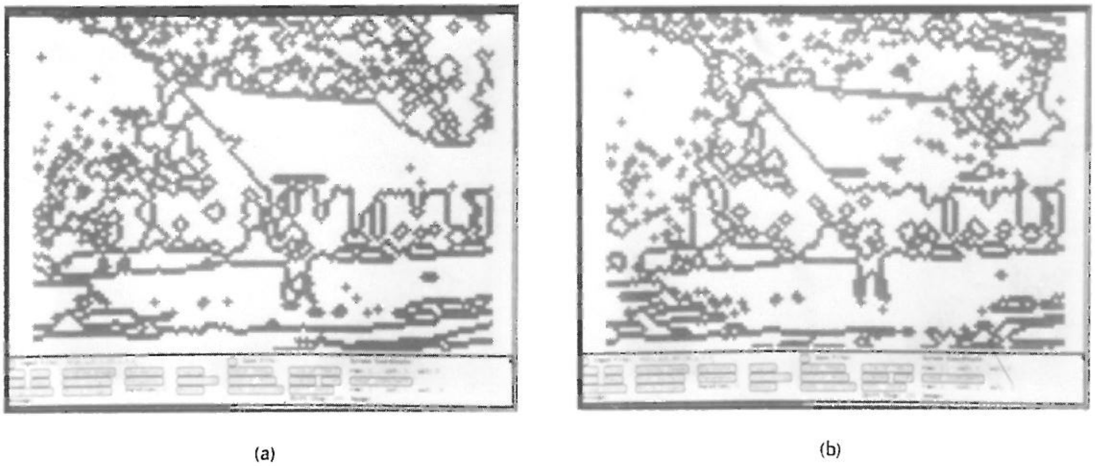
directed spatial clustering using the R, G, and B bands and the second by the same method, but using the transformed bands suggested by Ohta and co-workers (1980).

Thresholding

If the image contains a bright object against a dark background and the measurement–space is one-dimensional, measurement–space clustering amounts to determining a threshold such that all points smaller than or equal to the threshold are assigned to one cluster and the remaining points are assigned to the second cluster. In the easiest cases, a procedure to determine the threshold need only examine the histogram and place the threshold in the valley between the two modes. Unfortunately, it is not always the case that the two modes are nicely separated by a valley. To handle this kind of situation a variety of techniques can be used to combine the spatial information on the image with the gray-level intensity information to help in threshold determination.

Chow and Kaneko (1972) suggest using a threshold that depends on the histogram determined from the spatially local area around the pixel to which the threshold applies. Thus, for example, a neighborhood size of 33 × 33 or 65 × 65 can be used to compute the local histogram. Chow and Kaneko avoided the local histogram computation for each pixel's neighborhood by dividing the image into mutually exclusive blocks, computing the histogram for each block, and determining an appropriate threshold for each histogram. This threshold value can be considered to apply to the center pixel of each block. To obtain thresholds for the remaining pixels, they spatially interpolated the block center pixel thresholds to obtain a spatially adaptive threshold for each pixel.

Weszka and co-workers (1974) suggest determining a histogram for only those pixels having a high Laplacian magnitude. They reason that there will be a shoulder of the gray-level intensity function at each side of the boundary. The shoulder has high Laplacian magnitude. A histo-



(a)

(b)

Figure 18. Two segments of the color image. The left segmentation was achieved by recursive histogram-directed spatial clustering using R, G, and B bands. The right segment was achieved by the same method, but using the transformed bands $(R + G + B)/3$, $(R - B)/2$, and $(2G - R - B)/4$ suggested by Ohta and co-workers (1980).

gram of all shoulder pixels will be a histogram of all interior pixels just next to the interior border of the region. It will not involve those pixels in between regions that help make the histogram valley shallow. It will also have a tendency to involve equal numbers of pixels from the object and from the background. This makes the two histogram modes about the same size. Thus the valley-seeking method for threshold selection has a chance of working on the new histogram.

Weszka and Rosenfeld (1978) describe one method for segmenting white blobs against a dark background by a threshold selection based on busyness. For any threshold, busyness is the percentage of pixels having a neighbor whose thresholded value is different from their own thresholded value. A good threshold is that point near the histogram valley between the two peaks that minimizes the busyness.

Watanabe (1974) suggests choosing a threshold value that maximizes the sum of gradients taken over all pixels whose gray level equals the threshold value. Kohler (1981) suggests a modification of the Watanabe idea. Instead of choosing a threshold that maximizes the sum of gradient magnitudes taken over all pixels whose gray-level intensity equals the threshold value, Kohler suggests choosing that threshold that detects more high contrast edges and fewer low contrast edges than any other threshold.

Kohler defines the set $E(T)$ of edges detected by a threshold T to be the set of all pairs of neighboring pixels one of whose gray-level intensity is less than or equal to T and one of whose gray level intensity is greater than T:

$$E(T) = \{((i, j), (k, l))\, | \tag{1}$$

where pixels (i, j) and (k, l) are neighbors and

$$\min\{I(i, j), I(k, l)\} \leq T < \max\{I(i, j), I(k, l)\}\}$$

The total contrast $C(T)$ of edges detected by threshold T is given by

$$C(T) = \sum_{((i,j),(k,l))\in E(T)} \min\{|I(i, j) - T|_{,}\ |I(k, l) - T|\} \tag{2}$$

The average contrast of all edges detected by threshold T is then given by $C(T)/\#E(T)$. The best threshold T_b is determined by that value that maximizes $C(T_b)/\#E(T_b)$.

Milgram and Herman (1979) reason that pixels that are in between regions probably have in between gray-level intensities. If it is these pixels that are the cause of the shallow valleys, then it should be possible to eliminate their effect by only considering pixels having small gradients. They take this idea further and suggest that by examining clusters in the two-dimensional measurement space consisting of gray-level intensity and gradient magnitude, it is even possible to determine multiple thresholds when more than one kind of object is present.

Panda and Rosenfeld (1978) suggest a related approach for segmenting a white blob against a dark background. Consider the histogram of gray levels for all pixels that have small gradients. If a pixel has a small gradient, then it is not likely for it to be an edge. If it is not an edge, then it is either a dark background pixel or a bright blob pixel. Hence, the histogram of all pixels having small gradients will be bimodal and for pixels with small gradients, the valley between the two modes of the histogram is an appropriate threshold point. Next consider the histogram of gray levels for all pixels that have high gradients. If a pixel has a high gradient, then it is likely for it to be an edge. If it is an edge separating a bright blob against a dark background and if the separating boundary is not sharp but somewhat diffuse, then the histogram will be unimodal, the mean being a good threshold separating the dark background pixels from the bright blob pixels. Thus Panda and Rosenfeld suggest determining two thresholds: one for low gradient pixels and one for high gradient pixels. By this means they perform the clustering in the two-dimensional measurement–space consisting of gray-level intensity and gradient. A survey of threshold techniques can be found in Weszka (1978).

Multidimensional Measurement–Space Clustering

A LANDSAT image comes from a satellite and consists of seven separate images called bands. The bands are registered so that pixel (i, j) in one band corresponds to pixel (i, j) in each of the other bands. Each band represents a particular range of wavelengths. For multiband images such as LANDSAT or Thematic Mapper, determining the histogram in a multidimensional array is not feasible. For example, in a six-band image where each band has intensities between 0 and 99, the array would have to have $100^6 = 10^{12}$ locations. A large image might be 10,000 pixels per row by 10,000 rows. This only constitutes 10^8 pixels, a sample too small to estimate probabilities in a space of 10^{12} values were it not for some constraints of reality: (*1*) there is typically a high correlation between the band-to-band pixel values and (*2*) there is a large amount of spatial redundancy in image data. Both these factors create a situation in which the 10^8 pixels can be expected to contain only between 10^4 and 10^5 distinct 6-tuples. Based on this fact, the counting required for the histogram is easily done by mapping the 6-tuples into array indexes. The programming technique known as *hashing*, which is described in most data structures texts, can be used for this purpose.

Clustering using the multidimensional histogram is more difficult than univariate histogram clustering, because peaks fall in different places in the different histograms. Goldberg and Shlien (1977, 1978) threshold the multidimensional histogram to select all N-tuples situated on the most prominent modes. Then they perform a measurement–space connected components on these N-tuples to collect together all the N-tuples in the top of the most prominent modes. These measurement–space connected sets form the cluster cores. The clusters are defined as the set of all N-tuples closest to each cluster core.

An alternate possibility (Narendra and Goldberg, 1977) is to locate peaks in the multidimensional measurement space and region grow around them, constantly descending from each peak. The region growing includes all successive neighboring N-tuples whose probability is no higher than the N-tuple from which it is growing. Adjacent mountains meet in their common valleys.

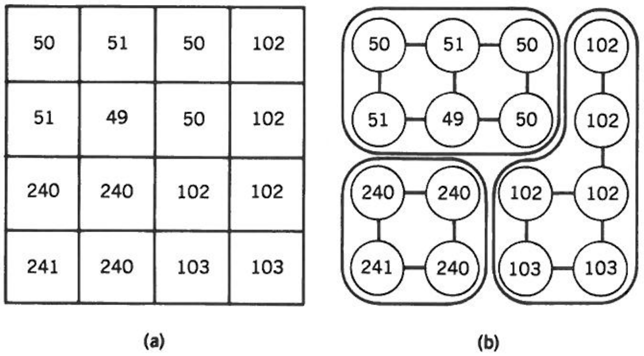
Rather than accomplish the clustering in the full measurement–space, it is possible to work in multiple lower order projection spaces and then reflect these clusters back to the full measurement–space. Suppose, for example, that the clustering is done on a four-band image. If the clustering done in bands 1 and 2 yields clusters c_1, c_2, c_3 and the clustering done in bands 3 and 4 yields clusters c_4 and c_5 then each possible 4-tuple from a pixel can be given a cluster label from the set $\{(c_1, c_4), (c_1, c_5), (c_2, c_4), (c_2, c_5), (c_3, c_4), (c_3, c_5)\}$. A 4-tuple (x_1, x_2, x_3, x_4) gets the cluster label (c_2, c_4) if (x_1, x_2) is in cluster c_2 and (x_3, x_4) is in cluster c_4.

REGION GROWING

Single Linkage Region Growing

Single linkage region growing schemes regard each pixel as a node in a graph. Neighboring pixels whose properties are similar enough are joined by an arc. The image segments are maximal sets of pixels all belonging to the same connected component. Figure 19 illustrates this idea with a simple image and the corresponding graph with the connected components circled. In this example, two pixels are connected by an edge if their values differ by less than five and they are 4-neighbors. Single linkage image segmentation schemes are attractive for their simplicity. They do, however, have a problem with chaining, because it takes only one arc leaking from one region to a neighboring one to cause the regions to merge.

As illustrated in Figure 19, the simplest single linkage scheme defines "similar enough" by pixel difference. Two neighboring pixels are similar enough if the absolute value of the difference between their gray-level intensity values is small enough. Bryant (1979) defines similar enough by normalizing the difference by the quantity (square root of 2) times the root mean square value of neighboring pixel differences taken over the entire image. For the image of Figure 19, the normalization factor is 99.22. The random variable that is the difference of two neighboring pixels normalized by the factor 1/99.22 has a normal distribution with mean 0 and standard deviation 99.22. A threshold can now be chosen in terms of the standard deviation instead of as an absolute value. For pixels having vector values, the obvious generalization is to use a vector norm of the pixel difference vector.



50	51	50	102
51	49	50	102
240	240	102	102
241	240	103	103

(a)

(b)

Figure 19. A simple gray-level image and the graph resulting from defining "similar enough" to be differing in gray level by less than five and using the 4-neighborhood to determine connected components.

Hybrid Linkage Region Growing

Hybrid single linkage techniques are more powerful than the simple single linkage technique. The hybrid techniques seek to assign a property vector to each pixel where the property vector depends on the $K \times K$ neighborhood of the pixel. Pixels that are similar are so because their neighborhoods in some special sense are similar. Similarity is thus established as a function of neighboring pixel values and this makes the technique better behaved on noisy data.

One hybrid single linkage scheme relies on an edge operator to establish whether two pixels are joined with an arc. Here an edge operator is applied to the image labeling each pixel as edge or nonedge. Neighboring pixels, neither of which are edges, are joined by an arc. The initial segments are the connected components of the nonedge labeled pixels. The edge pixels can either be left assigned edges and be considered as background or they can be assigned to the spatially nearest region having a label.

The quality of this technique is highly dependent on the edge operator used. Simple operators such as the Roberts and Sobel operators may provide too much region linkage, for a region cannot be declared as a segment unless it is completely surrounded by edge pixels. Haralick and Dinstein (1975), however, do report some success using this technique on LANDSAT data. They perform a dilation of the edge pixels in order to close gaps before performing the connected components operator. Perkins (1980) uses a similar technique.

Haralick (1982, 1984) discusses a very sensitive zero-crossing of second directional derivative edge operator. In this technique, each neighborhood is least squares fitted with a cubic polynomial in two variables. The first and second partial derivatives are easily determined from the polynomial. The first partial derivatives at the center pixel determine the gradient direction. With the direction fixed to be the gradient direction, the second partials determine the second directional derivative. If the gradient is high enough and if in the gradient direction, the second directional derivative has a negatively sloped zero-crossing inside the pixel's area, then an edge is declared in the neighborhood's center pixel.

Figure 20 shows the edges resulting from the second directional derivative zero-crossing operator using a gradient threshold of 4, a 9 × 9 neighborhood, and a zero-crossing radius of 0.85. The edges are well placed and a careful examination of pixels on perceived boundaries that are not classified as edge pixels will indicate the step edge pattern to be either nonexistent or weak. A connected components of the nonedge pixels accomplishes the initial segmentation. After the connected components operation, the edge pixels are assigned to their spatially closest component by a region filling operation. Figure 21 shows the boundaries from the region filled image. Obvi-

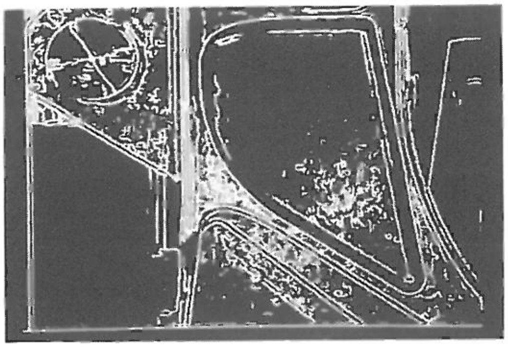


Figure 20. The second directional derivative zero-crossing operator using a gradient threshold of 4, a 9 × 9 neighborhood, and a zero-crossing radius of 0.85 applied to the bulkhead image of Figure 10.

ously, there are some regions that have been merged together. However, those boundaries that are present are placed correctly and they are reasonably smooth. Lowering the gradient threshold of the edge operator could produce an image with more edges and thereby reduce the edge gap problem. But this solution does not really solve the gap problem in general.

Yakimovsky (1976) assumes regions are normally distributed and uses a maximum likelihood test to determine edges. Edges are declared to exist between pairs of contiguous and exclusive neighborhoods if the hypothesis that their means are equal and their variances are equal has to be rejected. For any pair of adjacent pixels with mutually exclusive neighborhoods R_1 and R_2 having N_1 and N_2 pixels, respectively, the maximum likelihood technique computes the mean

$$\overline{X}_i = \frac{1}{N_i} \sum_{X \in R_i} X \tag{3}$$

and the scatter

$$S_i = \sum_{X \in R_i} (X - \overline{X}_i)^2 \tag{4}$$

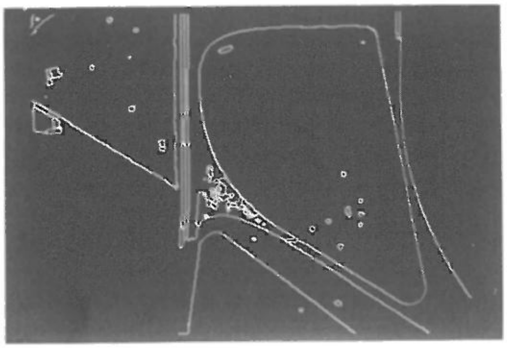


Figure 21. A hybrid linkage region growing scheme in which any pair of neighboring pixels, neither of which are edge pixels, can link together. The resulting segmentation consists of the connected components of the nonedge pixels and where edge pixels are assigned to their nearest connected component. This result was obtained from the edge image of Figure 20.

as well as the grand mean

$$\overline{X} = \frac{1}{N_1 + N_2} \sum_{X \in R_i \cup R_2} X \tag{5}$$

and grand scatter

$$S = \sum_{X \in R_1 \cup R_2} (X - \overline{X})^2 \tag{6}$$

The likelihood ratio test statistic T is given by

$$T = \frac{[S^2/(N_1 + N_2)]^{N_1 + N_2}}{[S_1^2/N_1]^{N_1}[S_2^2/N_2]^{N_2}} \tag{7}$$

Edges are declared between any pair of adjacent pixels when the T statistic from their neighborhoods is high enough. As N_1 and N_2 get large, 2 log T is asymptotically distributed as a chi-squared variate with 2 degrees of freedom.

If it can be assumed that the variances of the two regions are identical, then the statistic

$$F = \frac{(N_1 + N_2 - 2)N_1 N_2}{N_1 + N_2} \frac{(\overline{X}_1 - \overline{X}_2)^2}{S_1^2 + S_2^2} \tag{8}$$

has an F distribution with 1 and $N_1 + N_2 - 2$ degrees of freedom under the hypothesis that the means of the regions are equal. For an F value that is sufficiently large, the hypothesis can be rejected and an edge declared to exist between the regions.

Haralick (1981) suggests fitting a plane to the neighborhood around the pixel, and testing the hypothesis that the slope of the plane is zero. Edge pixels correspond to pixels between neighborhoods in which the zero slope hypothesis must be rejected. To determine a roof or V-shaped edge, Haralick suggests fitting a plane to the neighborhoods on either side of the pixel and testing the hypothesis that the coefficients of fit, referenced to a common framework, are identical.

Another hybrid technique first used by Levine and Leemet (1976) is based on the Jarvis and Patrick (1973) shared nearest neighbor idea. Using any kind of reasonable notion for similarity, each pixel examines its $K \times K$ neighborhood and makes a list of the N pixels in the neighborhood most similar to it. Call this list the similar neighbor list, where it is understood that a neighbor is any pixel in the $K \times K$ neighborhood. An arc joins any pair of immediately neighboring pixels if each pixel is in the other's shared neighbor list and if there are enough pixels common to their shared neighbor lists, that is, if the number of shared neighbors is high enough.

To make the shared neighbor technique work well, each pixel can be associated with a property vector consisting of its own gray-level intensity and a suitable average of the gray level intensity of pixels in its $K \times K$ neighborhood. For example, (x, a) and (y, b) can denote the property vectors for two pixels if x is the gray-level intensity value, a is the average gray-level intensity value in the neighborhood of the first pixel, y is the gray-level intensity value, and b is the average gray-level intensity value in the neighborhood of the second pixel. Similarity can be established by computing

$$S = w_1(x - y)^2 + w_2(x - b)^2 + w_3(y - a)^2 \qquad (9)$$

where w_1, w_2, and w_3 are nonnegative weights. Thus the quantity S takes into account the difference between the gray levels of the two pixels in question and the difference between the gray level of each pixel and the average gray level of the neighborhood of the other pixel. The weights w_1, w_2, and w_3 can be learned from training data for a particular class of images. The pixels are called similar enough for small enough values of S.

Pong and co-workers (1984) suggest an approach to segmentation based on the facet model of images. The procedure starts with an initial segmentation of the image into small regions. The initial segmentations used by Pong group together pixels that have similar facet fitting parameters, but any initial segmentation can be used. For each region of the initial segmentation, a property vector, which is a list of values of a set of predefined attributes, is computed. The attributes consist of such properties of a region as its area, its mean gray level, its elongation, and so on. Each region with associated property vector is considered a unit. In a series of iterations, the property vector of a region is replaced by a property vector that is a function of its neighboring regions. (The function that worked best in Pong's experiments replaced the property vector of a region with the property vector of the best-fitting neighborhood of that region.) Then adjacent regions having simlar final property vectors are merged. This gives a new segmentation that can then be used as input to the algorithm. Thus a sequence of coarser and coarser segmentations are produced. Useful variations are to prohibit merging across strong edge boundaries or when the variance of the combined region becomes too large. Figures 22, 23, and 24 illustrate the results of the Pong approach on the image of Figure 10 for one, two, and three iterations, respectively. Figure 25 illustrates the result of removing regions of size 25 or fewer pixels from the segmentation of Figure 24.

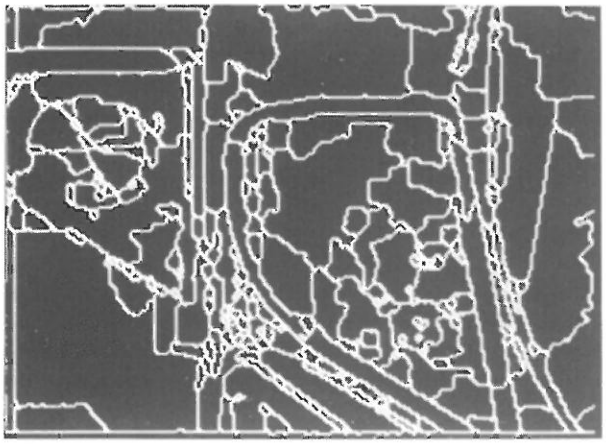


Figure 22. One iteration of the Pong algorithm on the bulkhead image of Figure 10.

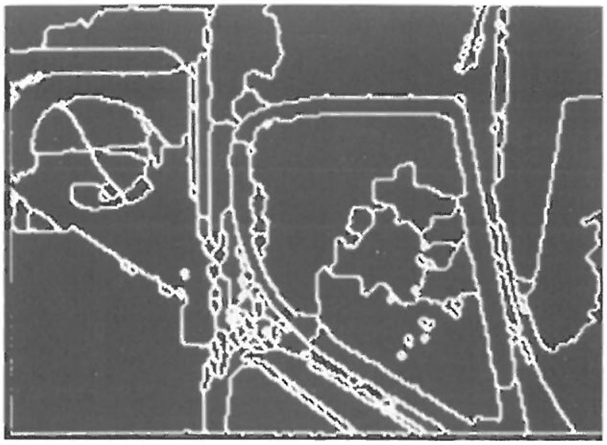


Figure 23. The second iteration of the Pong algorithm.

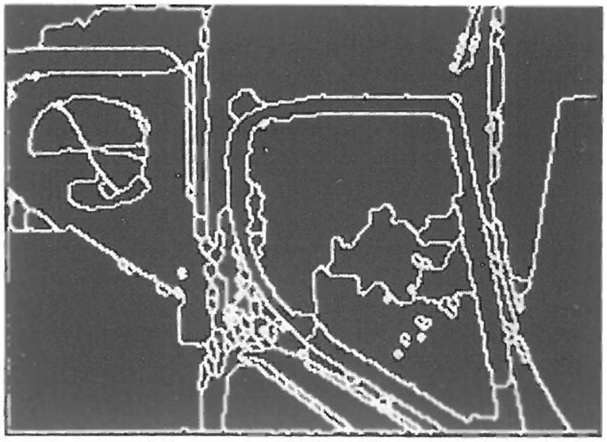


Figure 24. The third iteration of the Pong algorithm.

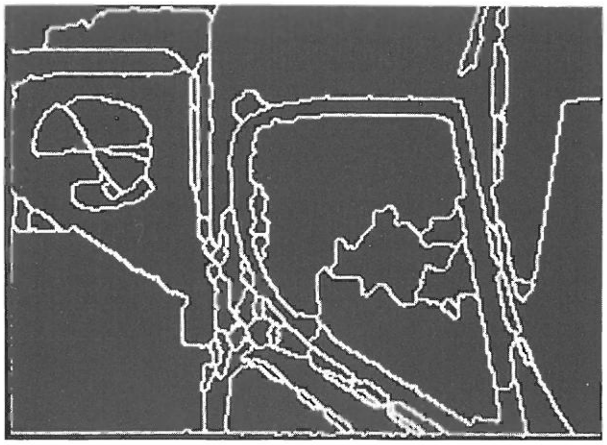


Figure 25. The segmentation obtained by removing regions smaller than size 25 from the segmentation of Figure 24.

Centroid Linkage Region Growing

In centroid linkage region growing, in contrast with single linkage region growing, pairs of neighboring pixels are not compared for similarity. Rather, the image is scanned in some predetermined manner such as left-right top-bottom. A pixel's value is compared to the mean of an already existing but not necessarily completed neighboring segment. If its value and the segment's mean value are close enough, then the pixel is added to the segment and the segment's mean is updated. If there is more than one region that is close enough, then it is added to the closest region. However, if the means of the two competing regions are close enough, the two regions are merged and the pixel is added to the merged region. If no neighboring region has its mean close enough, then a new segment is established having the given pixel's value as its first member. Figure 26 illustrates the geometry of this scheme.

Keeping track of the means annd scatters for all regions as they are being determined does not require large amounts of memory space. There cannot be more regions active at one time than the number of pixels in a row of the image. Hence, a hash table mechanism with the space of a small multiple of the number of pixels in a row can work well.

Another possibility is a single band region growing technique using the T-test. Let R be a segment of N pixels neighboring a pixel with gray-level intensity y. Define the mean $\overline{X}$ and scatter S^2 by

$$\overline{X} = \frac{1}{N} \sum_{(r,\, c) \in R} I(r, c) \qquad (10)$$

and

$$S^2 = \sum_{(r,\, c) \in R} (I(r, c) - X)^2 \qquad (11)$$

Under the assumption that all the pixels in R and the test pixel y are independent and have identically distributed normals, the statistic

$$T = \left[\frac{(N-1)N}{(N+1)} (y - \overline{X})^2 / S^2 \right]^{\frac{1}{2}} \qquad (12)$$

has a T_{N-1} distribution. If T is small enough y is added to region R and the mean and scatter are updated using y. The new mean and scatter are given by

$$\overline{X}_{\text{new}} \leftarrow (N\overline{X}_{\text{old}} + y)/(N + 1) \qquad (13)$$

and

$$S^2_{\text{new}} \leftarrow S^2_{\text{old}} + (y - \overline{X}_{\text{new}})^2 + N(\overline{X}_{\text{new}} - \overline{X}_{\text{old}})^2 \qquad (14)$$

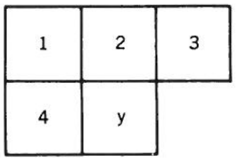


Figure 26. The region growing geometry for one-pass scan, left-right, top-bottom region growing. Pixel i belongs to region R_i whose mean is X_i, $i = 1, 2, 3$, and 4. Pixel y is added to a region R_j if by a T-test the difference between y and $\overline{X}_j$ is small enough. If for two regions R_i and R_j the difference is small enough, and if the difference between $\overline{X}_i$ and $\overline{X}_j$ is small enough, regions R_i and R_j are merged and y is added to the merged region. If the difference between $\overline{X}_i$ and $\overline{X}_j$ is significantly different, then y is added to the closest region.

If T is too high the value y is not likely to have arisen from the population of pixels in R. If y is different from all of its neighboring regions then it begins its own region. A slightly stricter linking criterion can require that not only must y be close enough to the mean of the neighboring regions, but that a neighboring pixel in that region must have a close enough value to y. This combines a centroid linkage and single linkage criterion. The next section discusses a more powerful combination technique, but first it is necessary to develop the concept of "significantly high."

To give a precise meaning to the notion of too high a difference, an α-level statistical significance test is used. The fraction α represents the probability that a T statistic with $N - 1$ degrees of freedom will exceed the value $t_{N-1}(\alpha)$. If the observed T is larger than $t_{N-1}(\alpha)$, then the difference is declared to be significant. If the pixel and the segment really come from the same population, the probability that the test provides an incorrect answer is α.

The significance level α is a user-provided parameter. The value of $t_{N-1}(\alpha)$ is higher for small degrees of freedom and lower for larger degrees of freedom. Thus, for region scatters considered to be equal, the larger a region is, the closer a pixel's value must be to the region's mean to merge into the region. This behavior tends to prevent already large regions from attracting to it many other additional pixels and tends to prevent the drift of the region mean as the region gets larger.

Note that all regions initially begin as one pixel in size. To avoid the problem of division by 0 (for S^2 is necessarily 0 for one pixel regions as well as for regions having identically valued pixels) a small positive constant can be added to S^2. One convenient way of determining the constant is to decide on a prior variance $V > 0$ and an initial segment size N. The initial scatter for a new one-pixel region is then given by NV and the new initial region size is given by N. This mechanism keeps the degrees of freedom of the T-statistic high enough so that a significant difference is not the huge difference required for a T-statistic with a small number of degrees of freedom. Figure 27 illustrates a second image of the F-15 bulkhead. Figure 28 illustrates the resulting segmentation of the bulkhead image for a 0.2% significance level test after all region smaller than 25 pixels have been removed.

Pavlidis (1972) suggests a more general version of this idea. Given an initial segmentation where the regions are

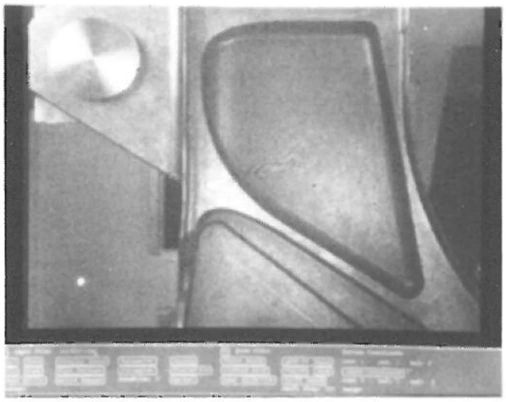


Figure 27. A second image of the F-15 bulkhead.

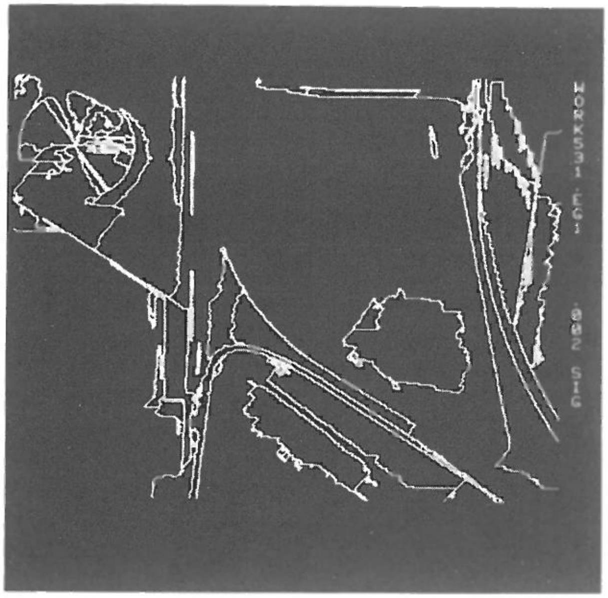


Figure 28. The one-pass centroid linkage segmentation of the bulkhead image of Figure 27. A significance level of 0.2% was used.

approximated by some functional fit guaranteed to have a small enough error, pairs of neighboring regions can be merged, if for each region the sum of the squares of the differences between the fitted coefficients for this region and the corresponding averaged coefficients, averaged over both regions, is small enough. Pavlidis gets his initial segmentation by finding the best way to divide each row of the image into segments with a sufficiently good fit. He also describes a combinatorial tree search algorithm to accomplish the merging that guarantees the best result. Kettig and Landgrebe (1975) successively merge small image blocks using a statistical test, They avoid much of the problem of zero scatter by considering only cells containing a 2 × 2 block of pixels.

Gupta and co-workers (1973) suggest using a T-test based on the absolute value of the difference between the pixel and the nearest region as the measure of dissimilarity. Kettig and Landgrebe (1975) discuss the multiband situation leading to the F-test and report good success with LANDSAT data.

Nagy and Tolaba (1972) just examine the absolute value between the pixel's value and the mean of a neighboring region formed already. If this distance is small enough, the pixel is added to the region. If there is more than one region, then the pixel is added to that region with the smallest distance.

The Levine and Shaheen scheme (1981) is simlar. The difference is that Levine and Shaheen attempt to keep regions more homogeneous and try to keep the region scatter from getting too high. They do this by requiring the differences to be more significant before a merge takes place if the region scatter is high. For a user-specified value θ, they define a test statistic T where

$$T = |y - \overline{X}_{\text{new}}| - (1 - S/\overline{X}_{\text{new}})\theta \qquad (15)$$

If $T < 0$ for the neighboring region R in which $|y - \overline{X}|$ is the smallest, then y is added to R. If $T > 0$ for the neighboring region in which $|y - \overline{X}|$ is the smallest, then y begins a new region. It should be noted that there are misprints in the formulas given for region scatter and region scatter updating in the Levine and Shaheen (1981) paper.

Brice and Fennema (1970) accomplish the region growing by partitioning the image into initial segments of pixels having identical intensity. They then sequentially merge all pairs of adjacent regions if a significant fraction of their common border has a small enough intensity difference across it.

Simple single-pass approaches that scan the image in a left-right, top-down manner are, of course, unable to make the left and right sides of a V-shaped region belong to the same segment. To be more effective, the single pass must be followed by some kind of connected components merging algorithm in which pairs of neighboring regions having means that are close enough are combined into the same segment. This is easily accomplished by using the two-pass label propagation logic of the Lumia and coworkers (1983) connected components algorithm.

After the top-bottom, left-right scan, each pixel has already been assigned a region label. In the bottom-up, right-left scan, the means and scatters of each region can be recomputed and can be kept in a hash table. Whenever a pair of pixels from different regions neighbor one another, a T-test can check for the significance of the difference between the region means. If the means are not significant, then they can be merged. A slightly stricter criterion would insist not only that the region means be similar, but also that the neighboring pixels from the different regions must be similar enough. Figure 29 shows the resulting segmentation of the bulkhead image for a 0.2% significance level after one bottom-up, right-left merging pass and after all regions smaller than 25 pixels have been removed.

One potential problem with region growing schemes is their inherent dependence on the order in which pixels and regions are examined. A left-right, top-down scan does not yield the same initial regions as a right-left, bottom-up scan or for that matter a column major scan. Usually, however, differences caused by scan order are minor.

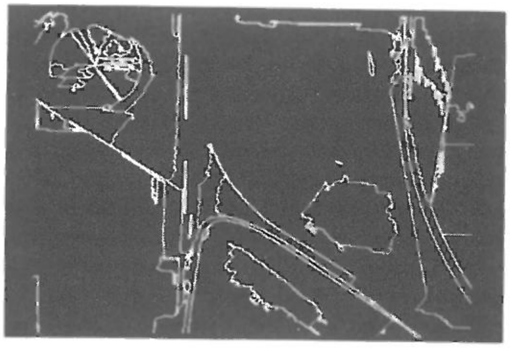


Figure 29. The two-pass centroid segmentation of the bulkhead image of Figure 27. A significance level of 0.2% was used on both passes.

HYBRID LINKAGE COMBINATIONS

The previous section mentioned the simple combination of centroid linkage and single linkage region growing. In this section the more powerful hybrid linkage combination techniques are discussed.

The centroid linkage and the hybrid linkage can be combined in a way that takes advantage of their relative strengths. The strength of the single linkage is that boundaries are placed in a spatially accurate way. Its weakness is that edge gaps result in excessive merging. The strength of centroid linkage is its ability to place boundaries in weak gradient areas. it can do this because it does not depend on a large difference between the pixel and its neighbor to declare a boundary. It depends on a large difference between the pixel and the mean of the neighboring region to declare a boundary.

The combined centroid hybrid linkage technique does the obvious thing. Centroid linkage is only done for non-edge pixels; that is, region growing is not permitted across edge pixels. Thus if the parameters of centroid linkage were set so that any difference, however large, between pixel value and region mean was considered small enough to permit merging, the two-pass hybrid combination technique would produce a connected components of the non-edge pixels. As the difference criterion is made more strict, the centroid linkage will produce boundaries in addition to those produced by the edges.

Figure 30 illustrates a one-pass scan combined centroid and hydrid linkage segmentation scheme using a significance level test of 0.2%. Edge pixels are assigned to their closest labeled neighbor, and regions having fewer than 25 pixels are eliminated. Notice that the resulting segmentation is much finer than that shown in Figures 28 and 29. Also the dominant boundaries are nicely curved and smooth. Figure 31 illustrates the two-pass scan combined centroid and hybrid linkage region growing scheme using a significance level test of 0.2%. The regions are somewhat simpler because of the merging done in the second pass.

SPATIAL CLUSTERING

It is possible to determine the image segments by simultaneously combining clustering in measurement–space with a spatial region growing. Such a technique is called spatial clustering. In essence, spatial clustering schemes combine the histogram mode seeking technique with a region growing or a spatial linkage technique.

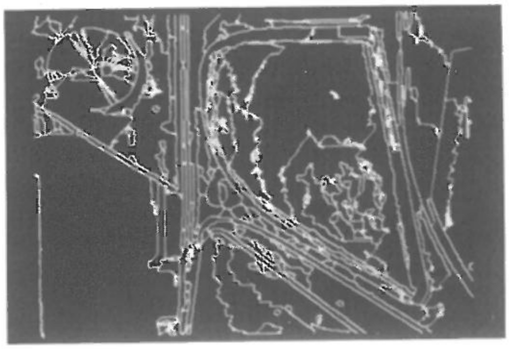


Figure 30. One-pass combined centroid and hybrid linkage segmentation of the bulkhead image of Figure 27. A significance level of 0.2% was used.

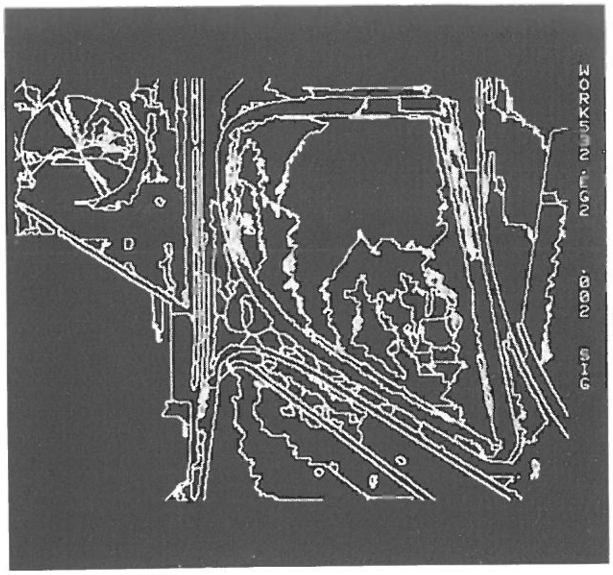


Figure 31. The two-pass combined centroid and hybrid linkage segmentation of the bulkhead image of Figure 27. A significance level of 0.2% was used on both passes.

Haralick and Kelly (1969) suggest that segmentation be done by first locating, in turn, all the peaks in the measurement–space histogram, and then determining all pixel locations having a measurement on the peak. Next, beginning with a pixel corresponding to the highest peak not yet processed, both spatial and measurement–space region growing are simultaneously performed in the following manner. Initially, each segment is the pixel whose value is on the current peak. Consider for possible inclusion into this segment a neighbor of this pixel (in general, the neighbors of the pixel that is being grown from) if the neighbor's value (an N-tuple for an N band image) is close enough in measurement–space to the pixel's value and if its probability is not larger than the probaiblity of the value of the pixel that is being grown from. Matsumoto and co-workers (1981) discuss a variation of this idea. Milgram (1979) defines a segment for a single band image to be any connected component of pixels, all of whose values lie in some interval I and whose border has a higher coincidence with the border created by an edge operator than for any other interval I. The technique has the advantage over the Haralick and Kelly technique in that it does not require the difficult measurement space exploring done in climbing down a mountain. However, it must try many different intervals for each segment. Extending it to efficient computation in multiband images appears difficult. However, Milgram does report good results of segmenting white blobs against a black background. Milgram and Kahl (1979) discuss embedding this technique into the Ohlander and co-workers (1978) recursive control structure.

Minor and Sklansky (1981) make more active use of the gradient edge image than Milgram, but restrict themselves to the more constrained situation of small convex-like segments. They begin with an edge image in which each edge pixel contains the direction of the edge. The orientation is so that the higher valued gray level is to the right of the edge. Then each edge sends out for a limited distance a message to nearby pixels and in a direction orthogonal to the edge direction. The message indicates what is the sender's edge direction. Pixels that pick up these messages from enough different directions must be interior to a segment.

The spoke filter of Minor and Sklansky counts the number of distinct directions appearing in each 3 × 3 neighborhood. If the count is high enough they mark the center pixel as belonging to an interior of a region. Then the connected components of all marked pixels is obtained. The gradient-guided segmentation is then completed by performing a region growing of the components. The region growing must stop at the high gradient pixels, thereby ensuring that no undesired boundary placements are made.

Burt and co-workers (1981) describe a spatial clustering scheme that is a spatial pyramid constrained ISODATA kind of clustering. The bottom layer of the pyramid is the original image. Each successive higher layer of the pyramid is an image having half the number of pixels per row and half the number of rows of the image below it. Initial links between layers are established by linking each parent pixel to the spatially corresponding 4 × 4 block of child pixels. Each pair of adjacent parent pixels has 8 child pixels in common. Each child pixel is linked to a 2 × 2 block of parent pixels. The iterations proceed by assigning to each parent pixel the average of its child pixels. Then each child pixel compares its value with each of its parent's values and links itself to its closest parent. Each parent's new value is the average of the children to which it is linked, etc. The iterations converge reasonably quickly for the same reason the ISODATA iterations converge. If the top layer of the pyramid is a 2 × 2 block of great grandparents, then these are at most 4 segments that are the respective great grandchildren of these 4 great grandparents. Pietikainen and Rosenfeld (1981) extend this technique to segment an image using textural features.

SPLIT AND MERGE

A splitting method for segmentation begins with the entire image as the initial segment. Then it successively splits each of its current segments into quarters if the segment is not homogeneous enough; that is, if the difference between the largest and smallest gray-level intensities is large. A merging method starts with an initial segmentation and successively merges regions that are similar enough.

Splitting algorithms were first suggested by Robertson (1973) and Klinger (1973). Kettig and Landgrebe (1975) try to split all nonuniform 2 × 2 neighborhoods before beginning the region merging. Fukada (1980) suggests successively splitting a region into quarters until the sample variance is small enough. Efficiency of the split and merge method can be increased by arbitrarily partitioning the image into square regions of a user selected size and then splitting these further if they are not homogeneous.

Because segments are successively divided into quarters, the boundaries produced by the split technique tend to be squarish and slightly artificial. Sometimes adjacent quarters coming from adjacent split segments need to be joined rather than remain separate. Horowitz and Pavlidis (1976) suggest the split-and-merge strategy to take care of this problem. They begin with an initial segmentation achieved by splitting into rectangular blocks of a prespecified size. The image is represented by a segmentation tree, which is a quadtree data structure (a tree whose nonleaf nodes each have four children). The entire image is represented by the root node. The children of the root are the regions obtained by splitting the root into four equal pieces, and so on. A segmentation is represented by a cutset, a minimal set of nodes separating the root from all of the leaves. In the tree structure, the merging process consists of removing four nodes from the cutset and replacing them with their parent. Splitting consists of removing a node from the cutset and replacing it with its four children. The two processes are mutually exclusive; all of the merging operations are followed by all of the splitting operations. The splitting and merging in the tree structure is followed by a final grouping procedure that can merge adjacent unrelated blocks found in the final cutset. Figure 32 illustrates the result of a Horowitz and Pavlidis type split-and-merge segmentation of the bulkhead image. Muerle and Allen (1968) suggest merging a pair of adjacent regions if a statistical test determines that their gray-level intensity distributions are similar enough. They recommend the Kolmogorov-Smirnov test.

Chen and Pavlidis (1980) suggest using statistical tests for uniformity rather than a simple examination of the difference between the largest and smallest gray-level intensities in the region under consideration for splitting. The uniformity test requires that there be no significant difference between the mean of the region and each of its

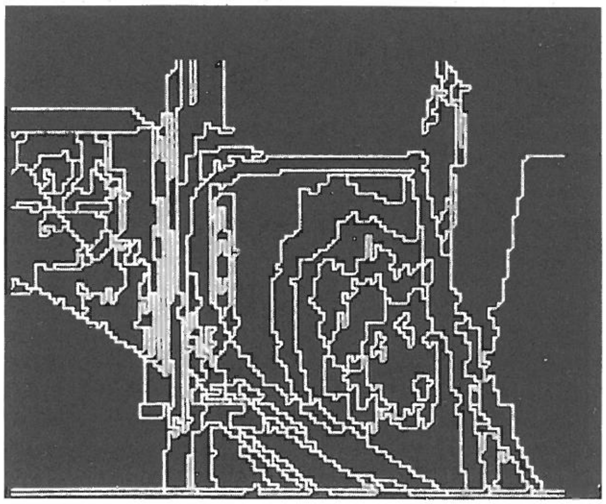


Figure 32. A split-and-merge segmentation of the bulkhead image of Figure 10.

quarters. The Chen and Pavlidis tests assume that the variances are equal and known.

Let each quarter have K pixels, X_{ij} be the jth pixel in the ith region, X_i be the mean of the ith quarter and $X_{..}$ be the grand mean of all the pixels in the four quarters. Then in order for a region to be considered homogeneous, Chen and Pavlidis require that

$$|X_i - X_{..}| \le \varepsilon, \qquad i = 1, 2, 3, 4 \tag{16}$$

where ε is a given threshold parameter.

The F-test for testing the hypothesis that the mean and variances of the quarters are identical is given here. This is the optimal test when the randomness can be modeled as arising from additive Gaussian-distributed variates. The value of variance is not assumed known. Under the assumption that the regions are independent and have identically distributed normals, the optimal test is given by the statistic F which is defined by

$$F = \frac{K \sum_{i=1}^{4} (X_{i.} - X_{..})^2/3}{\sum_{i=1}^{4} \sum_{k=1}^{K} (X_{ik} - X_{i.})^2/4(K - 1)} \tag{17}$$

It has a $F_{3,4(K-1)}$ distribution. If F is too high the region is declared not uniform.

The data structures required to do a split-and-merge on images larger than 512 × 512 are extremely large. Execution of the algorithm on virtual memory computers results in so much paging that the dominant activity may be paging rather than segmentation. Browning and Tanimoto (1982) give a description of a space-efficient version of the split-and-merge scheme that can handle large images, using only a small amount of main memory.

RULE-BASED SEGMENTATION

The rules behind each of the methods discussed so far are encoded in the procedures of the method. Thus it is not easy to try different concepts without complete reprogramming. Nazif and Levine (1984) solve this problem with a rule-based expert system for segmentation. The knowledge in the system is not application domain specific, but includes general-purpose, scene-independent knowledge about images and grouping criteria.

The Nazif and Levine system contains a set of processes, the initializer, the line analyzer, the region analyzer, the area analyzer, the focus of attention, and the scheduler, plus two associate memories, the short-term memory (STM) and the long-term memory (LTM). The short-term memory holds the input image, the segmentation data, and the output. The long-term memory contains the model representing the system knowledge about low level segmentation and control strategies. A system process matches rules in the LTM against the data stored in the STM. When a match occurs, the rule fires, and an action, usually involving data modification, is performed.

The model stored in the LTM has three levels of rules. At level 1 are knowledge rules that encode information about the properties of regions, lines, and areas in the form of situation–action pairs. The specific actions include splitting a region; merging two regions; adding, deleting, or extending a line; merging two lines; and creating or modifying a focus of attention area. Knowledge rules are classified by their actions. At level 2 are the control rules that are divided into two categories: focus-of-attention rules and inference rules. Focus-of-attention rules find the next data entry to be considered: a region, a line, or an entire area. These rules control the focus-of-attention strategy. The inference rules are metarules in that their actions do not modify the data in the STM. Instead, they alter the matching order of different knowledge rule sets. Thus they control which process will be activated next. At level 3, the highest rule level, are strategy rules that select the set of control rules that executes the most appropriate control strategy for a given set of data.

The conditions of the rules in the rule base are made up of (1) a symbolic qualifier depicting a logical operation to be performed on the data, (2) a symbol denoting the data entry on which the condition is to be matched, (3) a feature of this data entry, (4) an optional NOT qualifier, and (5) an optional DIFFERENCE qualifier that applies the operation to differences in feature values. Table 1 shows the different types of data entries allowed. Table 2 shows the different kinds of features, and Table 3 shows the possible actions that can be associated with a rule. Table 4 illustrates several rules from the system.

The Nazif and Levine approach to segmentation is useful because it is general, but allows more specific strategies to be incorporated without changing the code. Other rule-based segmentation systems tend to use high level knowledge models of the expected scene instead of general rules. The work of McKeown takes this approach for aerial images of airport scenes.

MOTION-BASED SEGMENTATION

In time-varying image analysis the data are a sequence of images instead of a single image. One paradigm under which such a sequence can arise is with a stationary camera viewing a scene containing moving objects. In each frame of the sequence after the first frame, the moving objects appear in different positions of the image than in

Table 1. Allowable Data-Entry Types in the Nazif and Levine Rule-Based Segmentation System

Data Entry	Symbol
Current region	REG
Current line	LINE
Current area	AREA
Region *adjacent* to current region	REGA
Region to the *left* of current line	REGL
Region to the *right* of current line	REGR
Line *near* the current line	LINEN
Line in *front* of current line	LINEF
Line *behind* current line	LINEB
Line *parallel to* current line	LINEP
Line *intersecting* current region	LINEI

Table 2. The Different Kinds of Features That Can Be Associated with the Condition Part of a Rule

	Numerical Descriptive Features	
Feature 1	Feature 2	Feature 3
Variance 1	Variance 2	Variance 3
Intensity	Intensity variance	Gradient
Gradient variance	X-centroid	Y-centroid
Minimum X	Minimum Y	Maximum X
Maximum Y	Starting X	Starting Y
Ending X	Ending Y	Starting direction
Ending direction	Average direction	Length
Start-End distance	Size	Perimeter
Histogram bimodality	Circularity	Aspect ratio
Uniformity 1	Uniformity 2	Uniformity 3
Region contrast 1	Region contrast 2	Region contrast 3
Line contrast 1	Line contrast 2	Line contrast 3
Line connectivity	Number of regions	Number of lines
Number of areas		

Numerical Spatial Features	
Number of *adjacent* regions	Adjacency values
Number of *intersecting* regions	Line content between regions
Distance to line in *front*	Nearest point on line in *front*
Distance to line *behind*	Nearest point of line *behind*
Distance to *parallel* line	Number of *parallel* points
Adjacency of *left* region	Adjacency of *right* region
Number of lines in *front*	Number of lines *behind*
Number of *parallel* lines	Number of regions to the *left*
Number of regions to the *right*	

Logical Features	
Histogram is bimodal	Region is bisected by line
Line is open	Line is closed
Line is loop	Line end is open
Line start is open	Line is clockwise
Area is smooth	Area is textured
Area is bounded	Area is new
One region to the *left*	One region to the *right*
Same region to the *left* and *right* of line	
Same region *left* of line 1 and line 2	
Same region *right* of line 1 and line 2	
Same region to the *left* of line 1 and *right* of line 2	
Same region to the *right* of line 1 and *left* of line 2	
Two lines are touching (8-connected)	
Areas are absent	Regions are absent
Lines are absent	System is starting
Process was regions	Process was lines
Process was areas	Process was focus
Process was generate areas	Process was active

the previous frame. Thus the motion of the objects creates a change in the images that can be used to help locate the objects and thus to segment the images.

Jain and co-workers (1979) used differencing operations to identify areas containing moving objects. The images of the moving objects were obtained by focusing the segmentation processes on these restricted areas. In this way, motion was used as a cue to the segmentation process. Thompson (1980) developed a method for partitioning a scene into regions corresponding to surfaces with distinct velocities. He first computed velocity estimates for each point of the scene and then performed the segmentation by a region-merging procedure that combined regions based on similarities in both intensity and motion.

Jain (1984) handled the more complex problem of segmenting dynamic scenes using a moving camera. He used the known location of the focus of expansion to transform the original frame sequence into another camera-centered sequence. The ego-motion polar transform (EMP) works as follows.

Suppose that A is a point in three space having coordinates (x, y, z), and the camera at time 0 is located at (x_0, y_0, z_0). During the time interval between frames, the camera undergoes displacement (dx_0, dy_0, dz_0), and the point

Table 3. The Different Kinds of Actions That Can Be Associated with a Rule

	Area Analyzer Actions	
Create smooth area	Add to smooth area	Save smooth area
Create texture area	Add to texture area	Save texture area
Create bounded area	Add to bounded area	Save bounded area
Relabel area to smooth	Relabel area to texture	
Relabel area to bounded	Delete area	
	Region Analyzer Actions	
Slit a region by histogram	Merge two regions	
Split region at lines		
	Line Analyzer Actions	
Extend line forward	Extend line backward	
Join lines forward	Join lines backward	
Insert line forward	Insert line backward	
Merge lines forward	Merge lines backward	
Delete line		
	Focus of Attention Actions	
Region with highest adjacency	Largest *adjacent* region	
Region with lowest adjacency	Smallest *adjacent* region	
Region with higher label	Next scanned region	
Region to the *left* of line	Region to the *right* of line	
Closest line in front	Closest line *behind*	
Closest *parallel* line	Shortest line that is near	
Longest line that is near	Strongest line that is near	
Weakest line that is near	Line with higher label	
Next scanned line	Line *intersecting* region	
Defocus (focus on whole image)	Focus on areas	
Clear region list	Clear line list	
Freeze area	Next area (any)	
Next smooth area	Next texture area	
Next bounded area		
	Supervisor Actions	
Initialize regions	Initialize lines	Generate areas
Match region rules	Match line rules	Match area rules
Match focus rules	Start	Stop

A undergoes displacement (dx, dy, dz). When the projection plane is at $z = 1$, the focus of expansion is at $(dx_0/dz_0, dy_0/dz_0)$. The projection A' of point A after the displacements is at (X, Y) in the image plane where

$$X = \frac{(x + dx - x_0 - dx_0)}{(z + dz - z_0 - dz_0)}$$

and

$$Y = \frac{(y + dy - y_0 - dy_0)}{(z + dz - z_0 - dz_0)}$$

The point A' is converted into its polar coordinates (r, θ) with the focus of expansion being the origin in the image plane. The polar coordinates are given by

$$\theta = tan^{-1}\left(\frac{dz_0(y + dy - y_0) - dy_0(z + dz - z_0)}{dz_0(x + dx - x_0) - dx_0(z + dz - z_0)}\right)$$

and

$$r = ((X - dx_0)^2 + (Y - dy_0)^2)^{\frac{1}{2}}$$

In (r, θ) space, the segmentation is simplified. Assume that the transformed picture is represented as a two-dimensional image have θ along the vertical axis and r along the horizontal axis. If the camera continues its motion in the same direction, then the focus of expansion remains the same, and θ remains constant. Thus the radial motion of the stationary point A' in the image plane due to the motion of the camera is converted to horizontal motion in (r, θ) space. If the camera has only a translational component to its motion, then all the regions that show only horizontal velocity in the (r, θ) space can be classified as due to stationary surfaces. The regions having a vertical velocity component are due to nonstationary surfaces. The segmentation algorithm first separates the stationary and nonstationary components on the basis of

Table 4. Several Examples of Rules from the Nazif and Levine System

A Region Merging Rule

IF: 1. The REGION SIZE is VERY LOW
2. The ADJACENCY with another REGION is HIGH
3. The DIFFERENCE in REGION FEATURE 1 is NOT HIGH
4. The DIFFERENCE in REGION FEATURE 2 is NOT HIGH
5. The DIFFERENCE in REGION FEATURE 3 is NOT HIGH
THEN: 1. MERGE the two REGIONS

A Region-Splitting Rule

IF: 1. The REGION SIZE is NOT LOW
2. The REGION AVERAGE GRADIENT is HIGH
3. The REGION HISTOGRAM is BIMODAL
THEN: 1. SPLIT the REGION according to the HISTOGRAM

A Line-Merging Rule

IF: 1. The LINE END point is OPEN
2. The LINE GRADIENT is NOT VERY LOW
3. The DISTANCE to the LINE IN FRONT is NOT VERY HIGH
4. The two LINES have the SAME REGION to the LEFT
5. The two LINES have the SAME REGION to the RIGHT
THEN: 1. JOIN the LINES by FORWARD expansion

A Control Rule

IF: 1. The LINE GRADIENT is HIGH
2. The LINE LENGTH is HIGH
3. SAME REGION LEFT and RIGHT of the LINE
THEN: 1. GET the REGION to the LEFT of the LINE

their velocity components in (r, θ) space. The stationary components are then further segmented into distinct surfaces by using the motion to assign relative depths to the surfaces.

SUMMARY

The place of segmentation in vision algorithms has been surveyed as well as common techniques of measurement–space clustering, single linkage, hybrid linkage, region growing, spatial clustering, and split and merge used in image segmentation. The single linkage region growing schemes are the simplest and most prone to the unwanted region merge errors. The hybrid and centroid region growing schemes are better in this regard. The split-and-merge technique is not as subject to the unwanted region merge error. However, it suffers from large memory usage and excessively blocky region boundaries. The measurement–space guided spatial clustering tends to avoid both the region merge errors and the blocky boundary problems because of its primary reliance on measurement space. But the regions produced are not smoothly bounded, and they often have holes, giving the effect of salt-and-pepper noise. The spatial clustering schemes may be better in this regard, but they have not been well enough tested. The hybrid linkage schemes appear to offer the best compromise between having smooth boundaries and few unwanted region merges. When the data form a time sequence of images, instead of a single image, motion-based segmentation techniques can be used. All the techniques can be made to be more powerful if they are based on some kind of statistical test for equality of means and more flexible if part of a rule-based system.

Not discussed as part of image segmentation is the fact that it might be appropriate for some segments to remain apart or to be merged not on the basis of the gray-level distributions, but on the basis of the object sections that they represent. The use of this kind of semantic information in the image segmentation process is essential for the higher level image understanding work. The work of McKeown describes a system that uses domain-specific knowledge in this manner.

BIBLIOGRAPHY

C. Brice and C. Fennema, "Scene Analysis Using Regions," *Artif. Intell.* **1,** 205–226 (1970).

J. D. Browning and S. L. Tanimoto, "Segmentation of Pictures into Regions with a Tile by Tile Method," *Patt. Recogn.* **15,** 1–10 (1982).

J. Bryant, "On the Clustering of Multidimensional Pictorial Data," *Patt. Recogn.* **11,** 115–125 (1979).

P. J. Burt, T. H. Hong, and A. Rosenfeld, "Segmentation and Estimation of Image Region Properties through Cooperative Hierarchical Computeration," *IEEE Trans. Sys. Man Cybernet.* **11,** 802–809 (1981).

P. C. Chen and T. Pavlidis, "Image Segmentation as an Estimation Problem," *Comput. Graphics Image Process.* **12,** 153–172 (1980).

C. K. Chow and T. Kaneko, "Boundary Detection of Radiographic Images by a Thresholding Method," in S. Wanatabe, ed., *Frontiers of Pattern Recognition,* Academic Press, Inc., New York, 1972, pp. 61–82.

K. S. Fu and J. K. Mui, "A Survey on Image Segmentation," *Patt. Recogn.* **13,** 3–16 (1981).

Y. Fukada, "Spatial Clustering Procedures for Region Analysis," *Patt. Recogn.* **12,** 395–403 (1980).

M. Goldberg and S. Shlien, "A Four-Dimensional Histogram Approach to the Clustering of LAND-SAT Data," in *Machine Processing of Remotely Sensed Data,* IEEE CH 1218-7 MPRSD, Purdue University, West Lafayette, Ind., 1977, pp. 250–259.

M. Goldberg and S. Shlien, "A Clustering Scheme for a Multispectral Image," *IEEE Trans. Sys. Man Cybernet.* **8,** 86–92 (1978).

J. N. Gupta, R. L. Kettig, D. A. Landgrebe, and P. A. Wintz, "Machine Boundary Finding and Sample Classification of Remotely Sensed Agricultural Data," in *Machine Processing of Remotely Sensed Data,* IEEE 73 CHO 834-2GE, Purdue University, West Lafayette, Ind., 1973, pp. 4B-25–4B-35.

R. M. Haralick, "Edge and Region Analysis for Digital Image Data," *Comput. Graphics Image Process.* **12,** 60–73 (1980).

R. M. Haralick, "Zero-Crossing of Second Directional Derivative Edge Operator," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers Technical Symposium East,* Arlington, Va., Vol. 336, 1982.

R. M. Haralick, "Digital Step Edges from Zero Crossing of Second

Directional Derivative," *IEEE Trans. Patt. Anal. Machine Intell.* **6**, 58–68 (1984).

R. M. Haralick and I. Dinstein, "A Spatial Clustering Procedure for Multi-Image Data," *IEEE Trans. Circuits Sys.* **22**, 440–450 (1975).

R. M. Haralick and G. L. Kelly, "Pattern Recognition with Measurement-Space and Spatial Clustering for Multiple Images," *Proc. IEEE* **57**, 654–665 (1969).

S. L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," *J. ACM* **23**, 368–388 (1976).

R. C. Jain, "Segmentation of Frame Sequences Obtained by a Moving Observer," *IEEE Trans. Patt. Anal. Machine Intell.* **6**, 624–629 (1984).

R. Jain, W. N. Martin, and J. K. Aggarwal, "Extraction of Moving Object Images through Change Detection," in *Proceedings of the Sixth IJCAI,* Tokyo, Morgan-Kaufmann, San Mateo, Calif., 1979, pp. 425–428.

R. A. Jarvis and E. A. Patrick, "Clustering Using a Similarity Measure Based on Shared Near Neighbors," *IEEE Trans. Comput.* **22**, 1025–1034 (1973).

T. Kanade, "Region Segmentation: Signal vs. Semantics," *Comput. Graphics Image Process.* **13**, 279–297 (1980).

R. L. Kettig and D. A. Landgrebe, "Computer Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," LARS Information Note 050975, Purdue University, West Lafayette, Ind., 1975.

K. Klinger, "Data Structures and Pattern Recognition," in *Proceedings of the First International Joint Conference on Pattern Recognition,* Washington, D.C., 1973, pp. 497–498.

R. Kohler, "A Segmentation System Based on Thresholding," *Comput. Graphics Image Process.* **15**, 319–338 (1981).

M. D. Levine and J. Leemet, "A Method for Non-Purposive Picture Segmentation," *Proceedings of the Third International Joint Conference on Pattern Recognition,* 1976, pp. 494–497.

M. D. Levine and S. I. Shaheen, "A Modular Computer Vision System for Picture Segmentation and Interpretation," *IEEE Trans. Patt. Anal. Machine Intell.* **3**, 540–556 (1981).

R. Lumia, L. G. Shapiro, and O. Zemiga, "A New Connected Components Algorithm for Virtual Memory Computers," *Comput. Vision Graphics Image Process.* **22**, 287–300 (1983).

K. Matsumoto, M. Naka, and H. Yanamoto, "A New Clustering Method for LANDSAT Images Using Local Maximums of a Multidimensional Histogram," in *Machine Processing of Remotely Sensed Data,* IEEE CH 1637-8 MPRSD, Purdue University, West Lafayette, Ind., 1981, pp. 321–325.

D. L. Milgram and M. Herman, "Clustering Edge Values for Threshold Selection," *Comput. Graphics Image Process.* **10**, 272–280 (1979).

D. L. Milgram and D. J. Kahl, "Recursive Region Extraction," *Comput. Graphics and Image Process.* **9**, 82–88 (1979).

L. G. Minor and J. Sklansky, "The Detection and Segmentation of Blobs in Infrared Images," *IEEE Trans. Sys. Man Cybernet.* **11**, 194–201 (1981).

J. Muerle and D. Allen, "Experimental Evaluation of Techniques for Automatic Segmentation of Objects in a Complex Scene," in G. Cheng and co-workers, eds., *Pictorial Pattern Recognition,* Thompson, Washington, D.C., 1968, pp. 3–13.

G. Nagy and J. Tolaba, "Nonsupervised Crop Classification Through Airborne Multispectral Observations," *IBM J. Res. Develop.* **16**, 138–153 (1972).

P. M. Narendra and M. Goldberg, "A Non-Parametric Clustering Scheme, for LANDSAT," *Patt. Recogn.* **9**, 207–215 (1977).

A. M. Nazif and M. D. Levine, "Low-Level Image Segmentation: An Expert System," *IEEE Trans. Patt. Anal. Machine Intell.* **6**(5), 555–557 (1984).

R. Ohlander, K. Price, and D. R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," *Comput. Graphics Image Process.* **8**, 313–333 (1978).

Y. Ohta, T. Kanade, and T. Sakai, "Color Information for Region Segmentation," *Comput. Graphics Image Process.* **13**, 222–241 (1980).

D. P. Panda and A. Rosenfeld, "Image Segmentation by Pixel Classification in (Gray Level, Edge Value) Space," *IEEE Trans. Comput.* **27**, 875–879 (1978).

T. Pavlidis, "Segmentation of Pictures and Maps through Functional Approximation," *Comput. Graphics Image Process.* **1**, 360–372 (1972).

W. A. Perkins, "Area Segmentation of Images Using Edge Points," *IEEE Trans. Patt. Anal. Machine Intell.* **2**, 8–15 (1980).

M. Pietikainen and A. Rosenfeld, "Image Segmentation by Texture Using Pyramid Node Linking," *IEEE Trans. Sys. Man Cybernt.* **11**, 822–825 (1981).

T. C. Pong, L. G. Shapiro, L. T. Watson, and R. M. Haralick, "Experiments in Segmentation Using a Facet Model Region Grower," *Comput. Vision Graphics Image Process.* **25**, 1–23 (1984).

E. Riseman and M. Arbib, "Segmentation of Static Scenes," *Comput. Graphics Image Process.* **6**, 221–276 (1977).

T. V. Robertson, "Extraction and Classification of Objects in Multispectral Images," *Machine Processing of Remotely Sensed Data,* IEEE 73 CHO 837-2GE, Purdue University, West Lafayette, Ind., 1973, pp. 3B-27–3B-34.

W. B. Thompson, "Combining Motion and Contrast for Segmentation," *IEEE Trans. Patt. Anal. Machine Intell.* **2**(6), 543–549 (1980).

S. Wanatabe and the CYBEST Group, "An Automated Apparatus for Cancer Prescreening: CYBEST," *Comput. Graphics Image Process.* **3**, 350–358 (1974).

J. S. Weszka, "A Survey of Threshold Selection Techniques," *Comput. Graphics Image Process.* **7**, 259–265 (1978).

J. S. Weszka, R. N. Nagel, and A. Rosenfeld, "A Threshold Selection Technique," *IEEE Trans. Comput.* **23**, 1322–1326 (1974).

J. S. Weszka and A. Rosenfeld, "Threshold Evaluation Techniques," *IEEE Trans. Sys. Man Cybernet.* **8**, 622–629 (1978).

Y. Yakimovsky, Y. "Boundary and Object Detection in Real World Image," *J. ACM* **23**, 599–618 (1976).

T. Y. Young, and T. W. Calvert, *Classification, Estimation, and Pattern Recognition,* Elsevier Science Publishing Co., Inc., New York, 1974.

S. Zucker, "Region Growing: Childhood and Adolescence," *Comput. Graphics Image Process.* **5**, 382–399 (1976).

General References

Y. G. Leclerc, "Constructing Simple Stable Descriptions of Image Partitioning," *Int. J. Comput. Vision* **3**, 73–102 (1989).

D. L. Milgram, "Region Extraction Using Convergent Evidence," *Comput. Graphics Image Process.* **11**, 1–12 (1979).

Robert Haralick
University of Washington

SELF-REPLICATION

In machine self-reproduction, an instruction-obeying device (such as a general-purpose computer) is augmented with physical manipulation capability (as in an industrial robot), supplied with raw materials, and programmed to produce a duplicate of itself. A theoretical model of this process was proposed by von Neumann (1951) in which the initial machine resides in an environment of spare parts (switching, sensing, cutting, fusing elements, etc). The parent machine plucks parts at random from its surroundings, identifies them, and following stored instructions, assembles the parts into a duplicate of itself.

This informally described kinematic model was superseded by von Neumann's (1951, 1966) cell-space model (Burks, 1970; Thatcher, 1970). [Conway's "Game of Life" is an example of an extremely simple cell-space system (Gardner, 1983; Berlekamp and co-workers, 1982).] In von Neumann's cell-space model machine reproduction takes place in an indefinitely extended, two-dimensional rectangular array, each square of which contains an identical automaton in direct communication with its four cardinal-direction neighbors. Each cell automaton is capable of being in any one of 29 different states. These states determine the way in which a cell automaton interacts with its neighbors. Depending on its state and the state of its neighbors, a cell automaton can transmit, switch, or store information or can undergo a change of state. Configurations of cell automata can be designed to form higher order information-processing devices, such as pulsers (units that when stimulated emit a stream of pulses) and decoders (units activated only on receipt of particular patterns of pulses). These and other higher order units can be combined to form a self-reproducing machine consisting of a general-purpose computer with an indefinitely expandable memory unit and a constructor (a device containing banks of pulsers that can emit signals that cause a cell automaton to assume any one of the 29 states).

The self-reproducing process proceeds as follows: the parent machine, reading instructions from memory, first directs the constructor to produce trains of pulses that transform cell automata at the periphery of the original machine, so that a constructing-arm pathway of newly activated cells is created and extended out into an undifferentiated region of the cell space. Then the parent machine, making use of a stored description of itself, directs the arm to move and to emit pulses so as to produce a configuration of cells that is identical to that of the original machine (although as yet lacking the memory contents of the original). The parent machine then reads its memory a second time and loads a copy of the contents into the memory of the offspring machine, turns on the new machine, and withdraws the constructing arm. This completes the self-reproduction.

This process of self-reproduction thus has two principal phases: first, the memory unit contents are read and interpreted as instructions for construction; next, the memory is read a second time in order to load a copy into the new memory. This action parallels the biological process of reading nucleic acids twice, once to carry out protein synthesis and again to replicate the genetic message.

Theoretical research since von Neumann has taken several directions. Alternative (usually simpler) cell spaces have been shown capable of supporting the reproductive process (Codd, 1968; Banks, 1970). Hybrid cellular-kinematic systems have been devised that make machine movement a more direct process (Arbib, 1966) (in the original von Neumann cell space, a machine movement is implemented by erasing a configuration of cells in one location and recreating it in another).

Other hybrid systems emphasize machine capacity for identifying system componentry (Laing, 1975). In such systems a machine may initially possess less than complete knowledge of itself but may still be able to reproduce itself because the deficiency can be made up by self-inspection (Laing, 1977). This also means that a machine can undertake partial self-repair: the machine compares its present configuration (obtained through self-inspection) with what its configuration should be (as contained in a stored description of itself) and uses its constructor to reduce the discrepancy. This strategy can be generalized to enable robotic machines to exhibit intentional goal seeking and evolution (Burks, 1984).

In a machine evolutionary process successive offspring machines cannot be mere exact duplicates of parents but must in some respect come to be both different and superior. One approach to machine evolution is to mimic the natural evolutionary processes of random variation of type and subsequent selection of better adapted types, but Myhill (1974) has shown that an indefinitely continued sequence of reproducing machines, each offspring superior to its parent, can be produced in an entirely deterministic fashion.

Machines that accept inputs and produce outputs can be viewed as implementing mathematical functions. In self-reproduction machines read or otherwise refer to or act on themselves to produce their outputs. Recursive-function theory is the abstract and general study of such self-reference computations and is thus an important tool for the precise investigation of self-reproducing systems. For example, the results and techniques of recursive-function theory were employed in the Myhill result cited above and also in establishing the conditions under which a reproducing machine system will eventually have a sterile descendant, will continue to produce descendants indefinitely in a periodic manner, or will produce descendants indefinitely but aperiodically (Case, 1974).

The processes by which artificial machines can exhibit various forms of reproduction, self-inspection, repair, and evolution can serve as explanatory models of similar processes in natural biological systems as well as contributing to the development of a broad theoretical biology of the possible organisms of possible universes.

Although complete physical artificial-machine self-reproduction has not yet been achieved, automation in which computer-controlled machines carry out the manufacture of other machines (including computing machines) has been moving steadily in that direction. Full exploitation of the concept will take advantage of the exponential nature of the reproductive process. Environmental concerns and the cost of energy and raw materials severely constrain Earth-based manufacturing of such an explo-