COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 25, 1–23 (1984)

Experiments in Segmentation Using a Facet Model Region Grower

TING-CHUEN PONG, LINDA G. SHAPIRO, LAYNE T. WATSON,

AND ROBERT M. HARALICK

Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

Received February 8, 1982; revised September 2, 1982

A region growing scheme based upon the facet model (R. M. Haralick, *Computer Graphics Image Processing* **12**, 1980, 60–73; R. M. Haralick and L. T. Watson, *Computer Graphics Image Processing* **15**, 1981, 113–129) is presented. The process begins with an initial segmentation which preserves much of the detailed resolution of the original image. Next a region property list and a region adjacency graph corresponding to the segmented image are constructed. Global information is then used to merge atomic regions. The region growing algorithm is based upon extensions of the facet model, but it is a higher-level algorithm which treats regions as primitive elements. The basic algorithm and several variations are described, including a version that uses a threshold on the amount a property vector is allowed to change to control the region growing process. The convergence of this thresholded facet iteration is also proved. Finally, the results of comparative experiments are presented.

1. INTRODUCTION

Image segmentation, the process of partitioning a digital image into regions, is important in nearly every application of image processing and pattern recognition. In particular, classification and description of the original scene are usually carried out in terms of regions and properties of the segmented image.

Region growing plays an important role in pictorial segmentation. A survey in this area is given by Zucker [15]. Here we briefly summarize Zucker's paper. The earliest major approach to region growing was the work of Muerle and Allen [10]. In their approach, gray level information was used in the growing process; regions were joined on the basis of similarity in the local statistical distribution of the gray levels. Brice and Fennema [1] developed a more global approach. Region merging was guided by the successive application of two heuristics: phagocyte and weakness. These heuristics use boundary information between pairs of adjacent regions.

In Pavlidis' work [7], functional approximation was adopted as a mathematical foundation for region growing. A two-dimensional picture is sliced into thin strips. Each strip is partitioned into segments and segments are approximated by polynomials. These approximated segments can now be merged into regions by comparing the coefficients of adjacent segments. The "split and merge" principle, proposed by Pavlidis and Horowitz [6], is another method using functional approximation. A two-dimensional picture is segmented into regions which are described in terms of an approximating function. Adjacent regions having similar approximations are merged while those regions that have large error norms are split. The process employs the pyramid data structure [12].

0734-189X/84 $3.00

Another region extraction method has been proposed by Milgram [9]. In his work, candidate object regions are extracted by thresholding the image in a region dependent way. Candidates are then accepted or rejected based on the coincidence of an edge map with the region boundary. Regions that best match the edge map are used to describe the objects in the image.

In the next section, we will discuss a facet model based region growing technique. Ideas related to the facet model first appeared in the papers of Davis *et al.* [2] and Tomita *et al.* [13]. Davis *et al.* suggested a method of region extraction by first averaging the picture and then thresholding it. This method will work well only if the picture contains regions for which the average gray levels lie in no more than two disjoint ranges. Tomita *et al.* proposed an improved method which finds the most homogeneous neighborhood among five rectangular neighborhoods containing a given pixel. The procedure gives the pixel the average gray level of its best-fitting neighborhood. This process can be repeated iteratively. Unfortunately, it does not yield good results for regions with complex shaped boundaries.

Recently, Nagao *et al.* [11] proposed an edge preserving smoothing algorithm which looks for the most homogeneous neighborhood around each pixel in a picture using an elongated rectangular neighborhood, and then assigns to each pixel the average gray level of the selected neighborhood area. Results showed that it removes noise in a flat region without blurring sharp edges or destroying the details of the boundary of the region. Haralick and Watson [3] and Haralick [4] generalized the facet model idea by using higher-order polynomial fitting functions and provided a unified view of edge and region analysis. The facet model is described in the following section.

2. INITIAL PROCESSING

The facet model [4] assumes that the image domain is composed of connected regions called facets. In the ideal image, the gray tone intensities for each facet are a polynomial function of the row and column coordinates of the pixels in the facet. To make the definition precise, let the spatial domain of an image be partitioned into K facets, $\{F(1),\ldots,F(K)\}$. For every pixel (r, c) that belongs to facet $F(k)$, the ideal image gray tone at (r, c) is represented by

$$g(r, c) = \sum_{i=0}^{N} \sum_{j=0}^{i} a_{ij}(k) * r^j * c^{i-j} \tag{1}$$

where the a_{ij}'s are constants. In particular, the flat facet model assumes that the gray level is constant in each ideal region. That is, N is assumed to be zero.

In [4], an image restoration procedure based on the slope facet model was suggested. In the slope facet model, the polynomials are assumed to have degree one; therefore each ideal region has a gray tone surface which is a sloped plane. For the general facet model, the restoration is done by fitting polynomials to the neighborhoods in an image. Given a square neighborhood $R \times C$, the gray tone $g(r, c)$ of each pixel $(r, c) \in R \times C$ is hypothesized to satisfy an equation of the form

$$g(r, c) = \sum_{i=0}^{N} \sum_{j=0}^{i} a_{ij} * r^j * c^{i-j} + n(r, c) \tag{2}$$

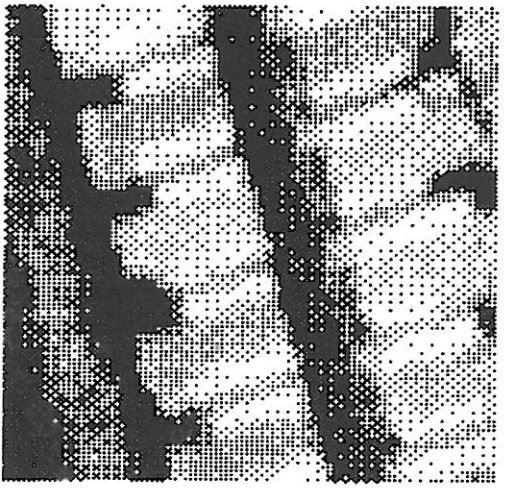


FIG. 1a. A 64 × 64 aerial image of a trailer park.

where the a_{ij}'s are coefficients of the best fit and n represents noise. A least squares procedure can be used to determine the constants a_{ij} that minimize

$$e^2 = \sum_{r \in R} \sum_{c \in C} \left[\left[\sum_{i=0}^{N} \sum_{j=0}^{i} a_{ij} * r^j * c^{i-j} - g(r, c) \right]^2 \right]. \tag{3}$$

For a square window of size M, each pixel is contained in at most M^2 windows. For each pixel in the image, we can find the window that gives the best least square

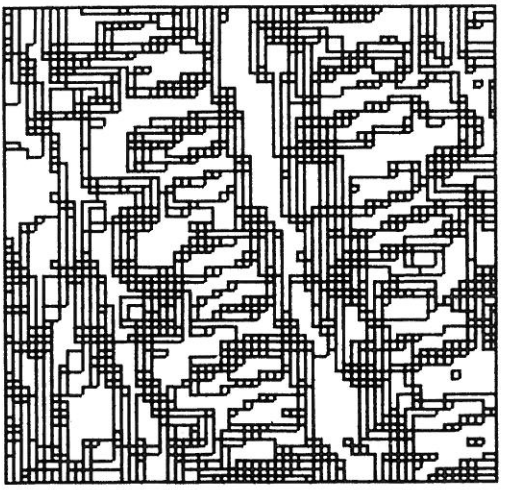


FIG. 1b. The initial segmentation of the image of Fig. 1a based on the flat facet model. There are 1304 regions.

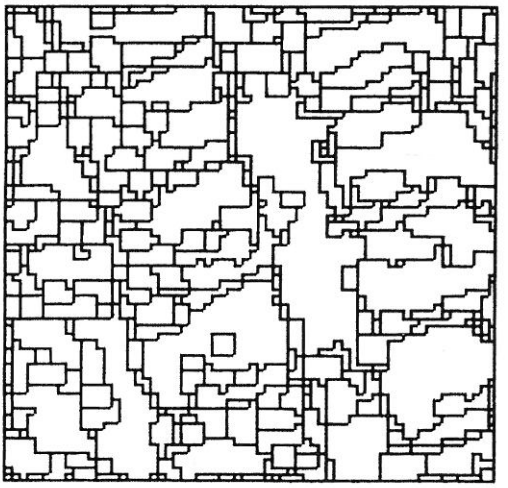


FIG. 1c. The initial segmentation of Fig. 1a based on the slope facet model. There are 350 regions.

fit, and replace the intensity of the pixel in the input image with the intensity of the best fit in the output image. The procedure can be iterated by using the output image as the new input; its convergence was proved in [3].

The facet model tells us that regions are connected sets of resolution cells whose gray tones belong to the same polynomial surface. Given the output of the above iterative scheme, the F statistic can be used to determine the significance of having an edge between two such polynomial surfaces. A segmented image can be produced using an F-statistic threshold which depends on the desired significance of the F test [4]. Figure 1a shows an aerial image of a trailer park. Figures 1b and c show the segmentations based on the flat facet model and the slope facet model, respectively.

Once the initial segmentation has been produced, properties of the initial regions are computed. Among the set of properties measured for each region, the following properties are used in our experiments.

(1) *Size* is simply the number of pixels in a region.

(2) *Mean gray level* is the average gray level intensity in a region.

(3) *Elongation* is a measure of the shape of a figure. It is obtained by finding the covariance matrix M of the distribution of $(r - \bar{r}, c - \bar{c})$ where (r, c) represents the coordinates of a pixel in region R, and $(\bar{r}, \bar{c})$ is the center of mass of R.

The matrix M is defined by

$$M = \begin{bmatrix} \sum_{(r,c)\in R} (r-\bar{r})^2 & \sum_{(r,c)\in R} (r-\bar{r})*(c-\bar{c}) \\ \sum_{(r,c)\in R} (c-\bar{c})*(r-\bar{r}) & \sum_{(r,c)\in R} (c-\bar{c})^2 \end{bmatrix}$$

and $(\bar{r}, \bar{c})$ for a region R is given by

$$\bar{r} = 1/\#R \sum_{(r,c)\in R} r \quad \text{and} \quad \bar{c} = 1/\#R \sum_{(r,c)\in R} c.$$

Two eigenvalues can be obtained from the matrix M. Elongation is defined as the ratio of the larger eigenvalue to the smaller.

Besides the property vector, a region adjacency graph [8], which gives topological information about the regions, is also generated for a segmented image. Two regions R1 and R2 are said to be *adjacent* for a segmented image if there exists some pixel in R1 such that its 4(8)-neighborhood intersects R2. The region adjacency graph has nodes corresponding to regions and edges that connect together nodes representing adjacent regions.

3. A REGION GROWING ALGORITHM

The problem with using any of the above initial segmentations as input to a higher-level algorithm attempting to recognize objects in the scene is that the regions are too small to be meaningful. This problem motivated us to develop our region growing scheme that starts with the F-test segmentation as an initial segmentation and produces a new segmentation having larger, hopefully more useful regions. Such a procedure could be repeated any number of times producing a sequence of rougher and rougher segmentations. The final result or the entire sequence of segmentations might prove useful to a higher-level process.

The initial segmentation was produced by grouping pixels using an iterative scheme. We can extend this idea to that of grouping regions using a similar iterative scheme. Basically each region is represented by a property vector. At each iteration, the property vector of a region can be replaced by some function of the property vectors of the regions constituting its best fitting neighborhood. After convergence of the iterative procedure, connected sets of regions with similar revised property vectors become the new regions.

The merging algorithm has two phases. In phase 1 the properties of each region are updated based upon the properties of its region neighborhood. In phase 2 adjacent regions which have similar updated property values are merged together. We now describe the algorithm and its several variations in detail.

3.1. Phase 1

Suppose that the image spatial domain has been divided into N nonoverlapping regions labeled $r(1),\ldots,r(N)$ with corresponding property vectors $p^k(1),\ldots,p^k(N)$. Define the neighborhood of region r, NBD(r), by

$$\text{NBD}(r) = \{r' | r' \text{ is adjacent to } r\}.$$

Suppose for some region r that $\text{NBD}(r) = \{r'(1),\ldots,r'(m)\}$. Then r is also an element of $\text{NBD}(r'(j))$ for $j = 1,\ldots,m$. Thus r participates in m different neighborhoods.

For a given neighborhood X, we define the variance of X, $\text{var}(X)$, by

$$\text{var}(X) = \sum_{r(j)\in X} \|(p(j) - \bar{p}(X))\|^2/(|X| - 1)$$

where $\bar{p}(X)$ is the mean property vector of X, and $|X|$ is the cardinality of X. The best-fitting neighborhood of region r, BF(r), is that one of the m neighborhoods it participates in that has lowest variance. Thus

$$\mathrm{BF}(r) = X^*$$

where $X^* = \mathrm{NBD}(r')$ for some $r' \in \mathrm{NBD}(r)$ and

$$\mathrm{var}(X^*) = \min_{r' \in \mathrm{NBD}(r)} \mathrm{var}(\mathrm{NBD}(r')).$$

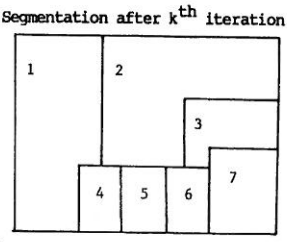
An iteration of the region growing algorithm starts with the set of regions $r(1),\ldots,r(N)$, with property vectors $p^k(1),\ldots,p^k(N)$, and replaces the property vector of each region by some function of the property vectors of its best-fitting neighborhood. That is,

$$p^{k+1}(n) = f(\mathrm{BF}^k(r(n))), \qquad n = 1,\ldots,N$$

where, of course, $\mathrm{BF}^k(r(n))$ depends on $p^k(n)$. The process is repeated until it reaches or approaches a fixed point. Then in phase 2, adjacent regions with identical or close property vectors are merged to form a new set of regions.

3.2. *Phase 2*

Suppose that we start with a segmented image whose regions are labeled $r^i(1),\ldots,r^i(N_i)$. If the process for these regions reaches a fixed point at some



r(i)	p^k(i)	NBD(r(i))	VAR^k(NBD(r(i))	$\bar{p}^k$(i)	p^{k+1}(i)
r(1)	182	1,2,4	13.013	194.67	194.67
r(2)	208	1,2,3,4,5,6	17.940	183.50	194.67
r(3)	187	2,3,6,7	47.021	162.50	183.50
r(4)	194	1,2,4,5	14.477	189.75	194.67
r(5)	175	2,4,5,6	23.051	183.00	189.75
r(6)	155	2,3,5,6,7	41.103	165.00	183.50
r(7)	100	3,6,7	44.004	147.33	165.00

If merging is performed at the k+1st iteration, the number of resulting regions is 4.

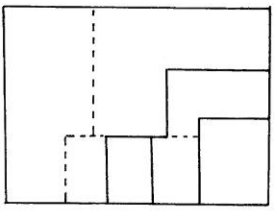


FIG. 2. Method 1 for updating property vectors by using the mean property vector of the best-fitting neighborhood.

iteration, we merge them to form a new set of regions $r^{i+1}(1),\ldots,r^{i+1}(N_{i+1})$ in the following way.

Construct a graph in which the nodes are the regions. Link together all pairs of regions

(a) which are adjacent, and

(b) whose updated property vectors are close enough.

Determine the connected components of the resulting graph. Each connected component corresponds to a subset of regions whose union constitutes one of the merged regions for the next cycle.

For an image with T regions, let $n(i)$ be the number of neighbors for region i. Then the number of computations for an iteration of this algorithm is proportional to $n(1) + n(2) + \cdots + n(T)$, which gives a computational complexity $O(T^*\bar{n})$, where $\bar{n}$ is the average number of neighboring regions for all the regions. In most cases, $\bar{n} \ll T$, which makes this an efficient algorithm.

3.3. Updating the Property Vectors

One of the most important steps in the above region growing algorithm is to update the property vector for each region. Three different alternatives have been tried. The first method uses the mean property vector of its best-fitting neighborhood. At iteration k, the updated property vector of a region R is given by

$$p^{k+1}(R) = \bar{p}(R^*)$$

where $R^* = \mathrm{BF}(R)$. Extensions of the theorem for the flat facet model in [3] guarantee the convergence of this method. Figure 2 illustrates this updating process.

The second way of updating the property vector of a region is to make it take on the original property vector of the region that defines the best-fitting neighborhood. That is, instead of using $p^{k+1}(R) = \bar{p}(R^*)$ as in method (1), the property vector is updated by

$$p^{k+1}(R) = p^k(S)$$

where $\mathrm{NBD}(S) = \mathrm{BF}(R)$. Results showed that the rate of merging by this method is faster than the first. But unfortunately, this method shows oscillatory behavior for some images; it does not always lead to a fixed point. As an example, Fig. 3 shows an oscillation with a length one cycle. This is bad because it will give rise to two (or more) interpretations of the same image.

The third alternative is to calculate the mean by weighting the property vectors of a neighborhood by their region sizes. The algorithm used to update the mean and variance of the weighted property vectors is suggested by West [14]. Our results showed that the first approach, using the mean property vector, is the most reliable method. For the rest of the discussion, the experiments performed employ the first approach.

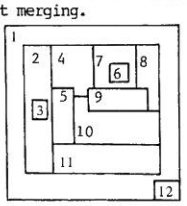
3.4. *Thresholded Flat Facet Iteration*

The updating schemes described in the previous section recompute the property vector of every region at each iteration. To prevent inaccurate segmentations due to property vectors changing too much, we need to inhibit the updating if the new property vector of a region is too different from the original one. To accomplish this, the idea of a thresholded flat facet iteration is introduced. For a flat facet iteration with threshold e, the updated property vector of a region R is given by

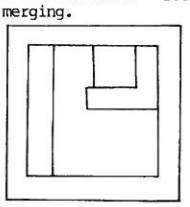
$$\begin{aligned} p^{k+1}(R) &= p^k(R), && \text{if } \|\bar{p}(R^*) - p^k(R)\| > e \\ &= \bar{p}(R^*), && \text{otherwise.} \end{aligned}$$

Experiments show that except for extreme values of e, the differences caused by the choice of e are minor. For simplicity, the value of e is chosen to be a fraction of the intensity range of the image.

Segmentation after k^{th} iteration without merging.

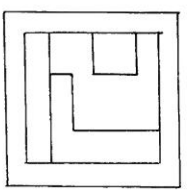


Segmentation after k^{th} iteration with merging.



r(i)	p^k(i)	VAR^i(NBD(r(i))	NBD(r(i))	p^{k+1}(i)	VAR^{k+1}(NBD(r(i))	p^{k+2}(i)
r(1)	100.00	25.71	1,2,4,7,8, 10,11,12	100.00	25.03	100.00
r(2)	137.33	21.10	1,2,3,4,5, 11	137.33	20.62	137.33
r(3)	137.33	0.00	2,3	137.33	0.00	137.33
r(4)	154.50	21.73	1,2,4,5,7 9,10	149.89	20.54	154.50
r(5)	154.50	7.68	2,4,5,10,11	154.50	7.04	154.50
r(6)	166.33	0.00	7,6	166.33	0.00	166.33
r(7)	166.33	24.60	1,4,6,7,8,9	166.33	24.42	166.33
r(8)	149.89	28.60	1,7,8,9	149.89	25.18	149.89
r(9)	149.89	6.73	4,7,8,9,10	149.89	7.35	149.89
r(10)	154.50	23.94	1,4,5,8,9 10,11	149.89	21.24	154.50
r(11)	154.50	23.65	1,2,5,10,11	154.50	23.03	154.50
r(12)	100.00	0.00	1,12	100.00	0.00	100.00

Segmentation after $k+1^{st}$ iteration with merging.



Segmentation after $k+2^{nd}$ iteration with merging.

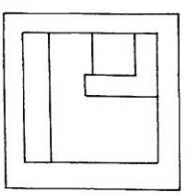


FIG. 3. Method 2 for updating property vectors using the central property vector of the best-fitting neighborhood. The oscillatory behavior is illustrated, since $p^k(i) = p^{k+2}(i)$.

The convergence of the flat facet iteration is guaranteed [3]. As a result of the following theorem, the *thresholded* flat facet iteration also converges.

THEOREM. *Let* $X^0(1),\ldots,X^0(N)$ *be a sequence of real numbers. The thresholded flat facet iteration produces sequences* $X^m(1),\ldots,X^m(N)$, $m = 1,2,\ldots$, *defined by*

$$
\begin{aligned}
X^{m+1}(k) &= X^m(k), && \text{if } \min(|X^m(k-1) - X^m(k)|, \\
&&& |X^m(k) - X^m(k+1)|) > e \\
&= \frac{X^m(k-1) + X^m(k)}{2}, && \text{if } |X^m(k-1) - X^m(k)| \\
&&& \le \min(e, |X^m(k) - X^m(k+1)|) \\
&= \frac{X^m(k) + X^m(k+1)}{2}, && \text{otherwise.}
\end{aligned}
$$

Then,

(1) *The algebraic order of* $X^0(1),\ldots,X^0(N)$ *is preserved by each sequence* $X^m(1),\ldots,X^m(N)$.

(2) *If* $X^0(k)$ *is a local minimum (maximum), then so is* $X^m(k)$ *for every* m.

(3) *For all* k *and* m, $\min_i X^0(i) \le X^m(k) \le \max_i X^0(i)$.

(4) *For all* k, $X^m(k)$ *is either monotone increasing or decreasing.*

(5) $\lim_{m\to\infty}(X^m(1),\ldots,X^m(N)) = (X^\infty(1),\ldots,X^\infty(N))$ *exists.*

A convergence proof for the simplest one-dimensional flat facet iteration is given in the Appendix. That proof contains the essence of the more general result without being obscured by technical details. The theorem can be readily generalized to larger neighborhoods and higher dimensions.

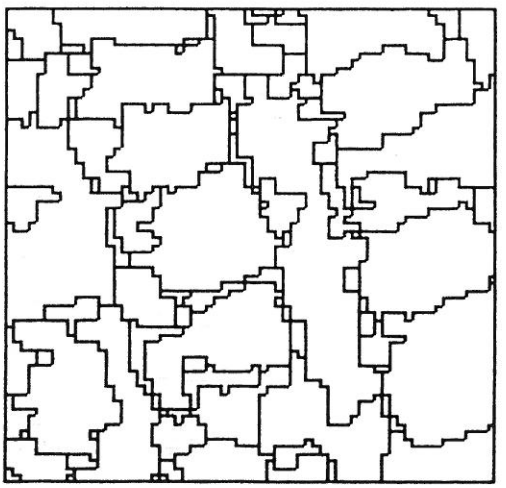


FIG. 4. The results of merging after five iterations of region growing, using replacement method 1, where the property vector of a region is replaced by the mean property vector of its best-fitting neighborhood. Note that some boundaries between two trailers are shifted because a boundary region merged with just one of the trailers. There are 110 regions.

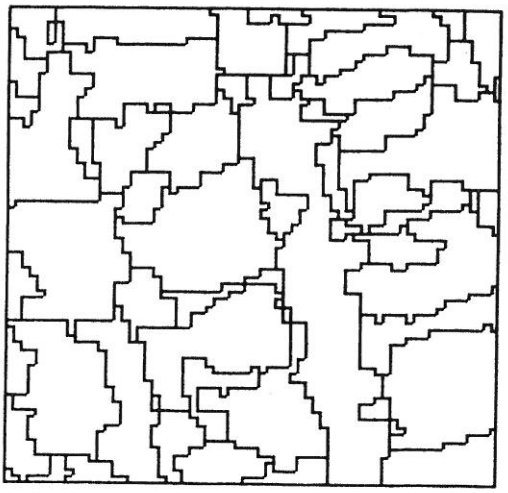


FIG. 5a. The results of merging after six iterations of region growing, using replacement method 2, where the property vector of a region is replaced by the central property vector of its best-fitting neighborhood. There are 58 regions.

4. EDGE HEURISTICS

Our results show that the algorithm suggested above reduces the number of regions considerably; because many undesired regions still exist, more processing is needed. In this section we attempt to achieve a better partition of the picture by using heuristics to guide the merging of regions.

Regions are often characterized by well-defined boundaries. Experiments show that boundaries are often picked up as small regions during the initial facet

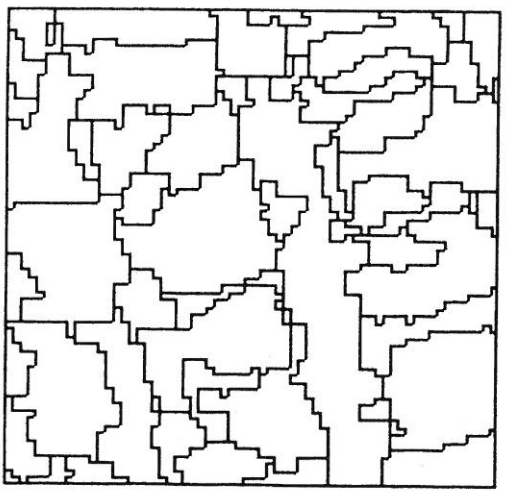


FIG. 5b. The results of merging after seven iterations of region growing, using replacement method 2. There are again 58 regions, but the segmentation is slightly different from Fig. 5a. However, after eight iterations, the results of Fig. 5a are repeated, showing the oscillatory nature of this method.

segmentation. These regions will sometimes create problems during region growing processes. Because boundary regions have width, a shift of boundary between two objects will result if the boundary region between two objects is merged with one of the regions. In this section, a combined use of the proposed region growing algorithm and edge information is investigated.

We define edge regions as regions with mean gradient values greater than a certain threshold. To find edge regions, a gradient image is computed from the coefficients of the first order terms (a_{10} and a_{11} of (3)) of the fitting polynomials. The gradient image can consist of one of three norms of the gradient:

(1) the l^1 norm: sum of the absolute values of a_{10} and a_{11};

(2) the l^2 norm: square root of sum of the squares of a_{10} and a_{11}; or

(3) the l^∞ norm: maximum of the absolute values of a_{10} and a_{11}.

The mean gradient value for a region is the mean of the gradient values of the pixels in the region. An appropriate threshold can be found from the histogram of the property vectors.

After the edge regions are identified, the region growing process is divided into two stages. In the first step, the region growing algorithm described in Section 3 is applied except that edge regions are excluded from the atomic regions. That is, edge regions will not participate in any of the neighborhoods. Edge regions, therefore, will remain unmerged after this growing process. The second step is to eliminate edge regions. This is done by a filling algorithm. All the pixels in edge regions are first marked with zeros in the segmented image and then a symmetric fill operation is performed on the zeros.

One advantage to this approach is that it not only prevents the shifting of the boundary, but also helps to stop regions from growing too far. The difficulty with this method is the identification of edge regions from the rest.

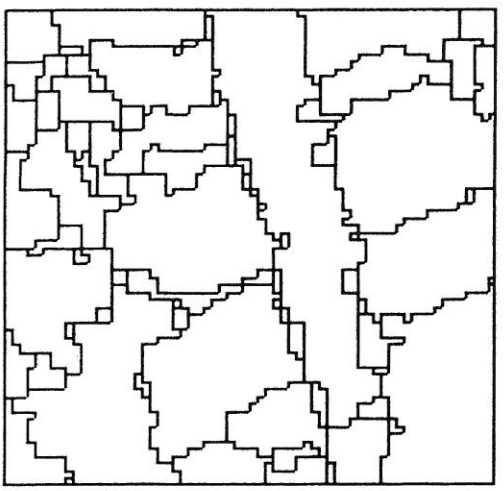


FIG. 6. The results of merging after five iterations of region growing, using replacement method 3, where weighted mean and variance are calculated and the property vector of a region is replaced by the weighted mean property vector of its best-fitting neighborhood. There are 80 regions.

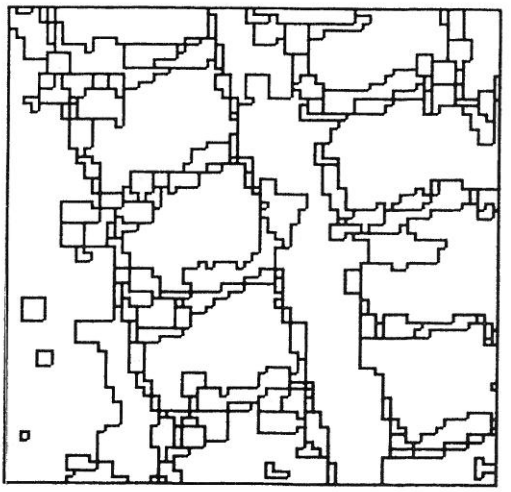


FIG. 7a. The results of merging after five iterations of region growing, using replacement method 1, but refusing to allow edge regions to merge. There are 143 regions.

5. EXPERIMENTAL RESULTS

In this section we discuss the results of applying the region growing algorithm to three images: an artificially generated image, a small aerial image of a trailer park (64 × 64), and a larger aerial image of a commercial/residential area (256 × 256).

The 64 × 64 trailer park image (Fig. 1a) and its initial slope facet segmentation (Fig. 1c) are used to illustrate the three updating methods and the entire process further. Note that most boundaries are picked up as small regions in the initial segmentation. This set of experiments used only mean gray level as the criteria to merge regions. Figure 4 shows the results of merging after five iterations of method 1. Figures 5a and b show the results of merging after six and seven iterations,

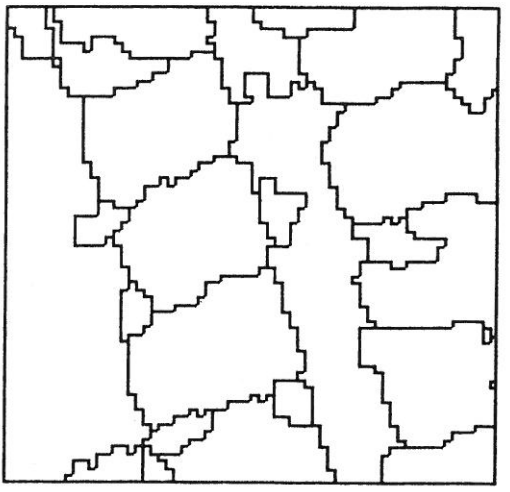


FIG. 7b. The results of performing a symmetric fill operator on the edge regions of Fig. 7a so that the edge regions are merged with the larger regions on all sides of them. There are 29 regions.

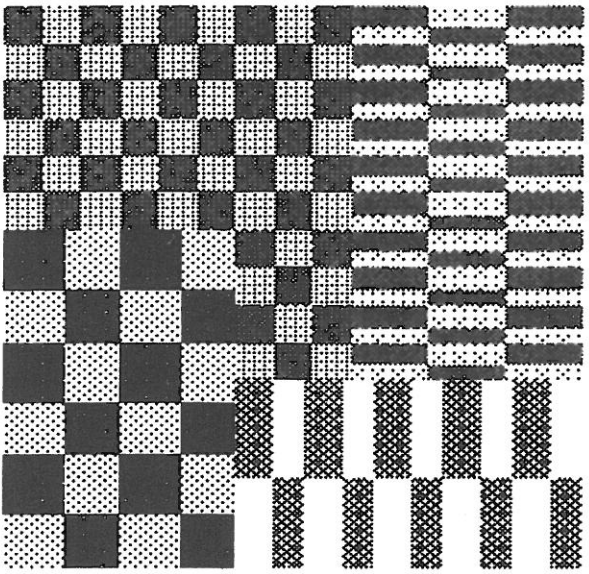


FIG. 8a. A 150 × 150 image of a combination of different checkerboards; the checks are of sizes 10 × 10, 5 × 20, 15 × 15, and 9 × 25 pixels.

respectively, of method 2. After eight iterations, the results after merging are the same as after six iterations, showing the oscillatory behavior. Figure 6 shows the results of merging after five iterations of method 3. Figure 7a shows the results after five iterations of method 1, but where edge regions are unmerged. Figure 7b shows the results of the filling operation on Fig. 7a. The regions in Fig. 7b show a very good correspondence with the trailers in the original image.

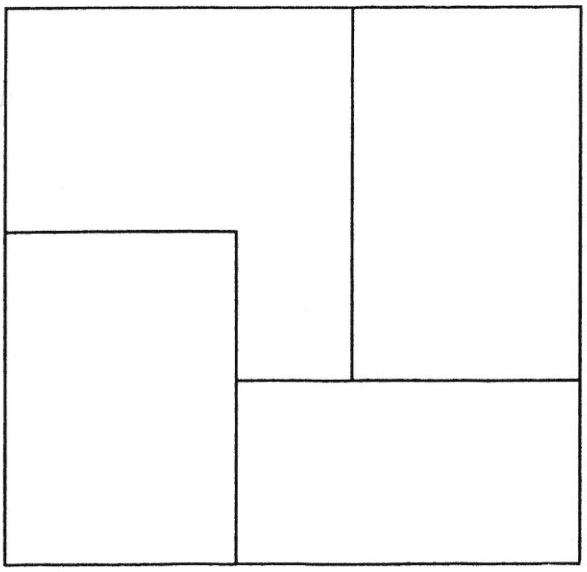


FIG. 8b. The results of merging after five iterations of region growing, using the size and elongation measures.

To understand the performance of the region grower by using properties other than just the gray level intensities, we examined its behavior on an artificially generated image (Fig. 8a). We used a 150 × 150 pixel image of a combination of different checkerboards; the checks are of sizes 10 × 10, 5 × 20, 15 × 15, or 9 × 25 pixels. The slope facet segmentation created a perfect initial segmentation of this image into the separate checks. The goal of the final segmentation was to group checks which belong to the same checkerboard. Since within each of the different checkerboards, size and elongation of checks are constant while gray tone varies, size and elongation were selected as the components of the property vector for each region. The results of applying the region grower are shown in Fig. 8b.

Another set of experiments were performed on a more complex image. Figure 9 shows an aerial image of an urban area. The flat facet model F test produced an initial segmentation of 13,602 regions (Fig. 10). The property vector for each region consisted of only the mean gray level. The merging incorporating the edge heuristic

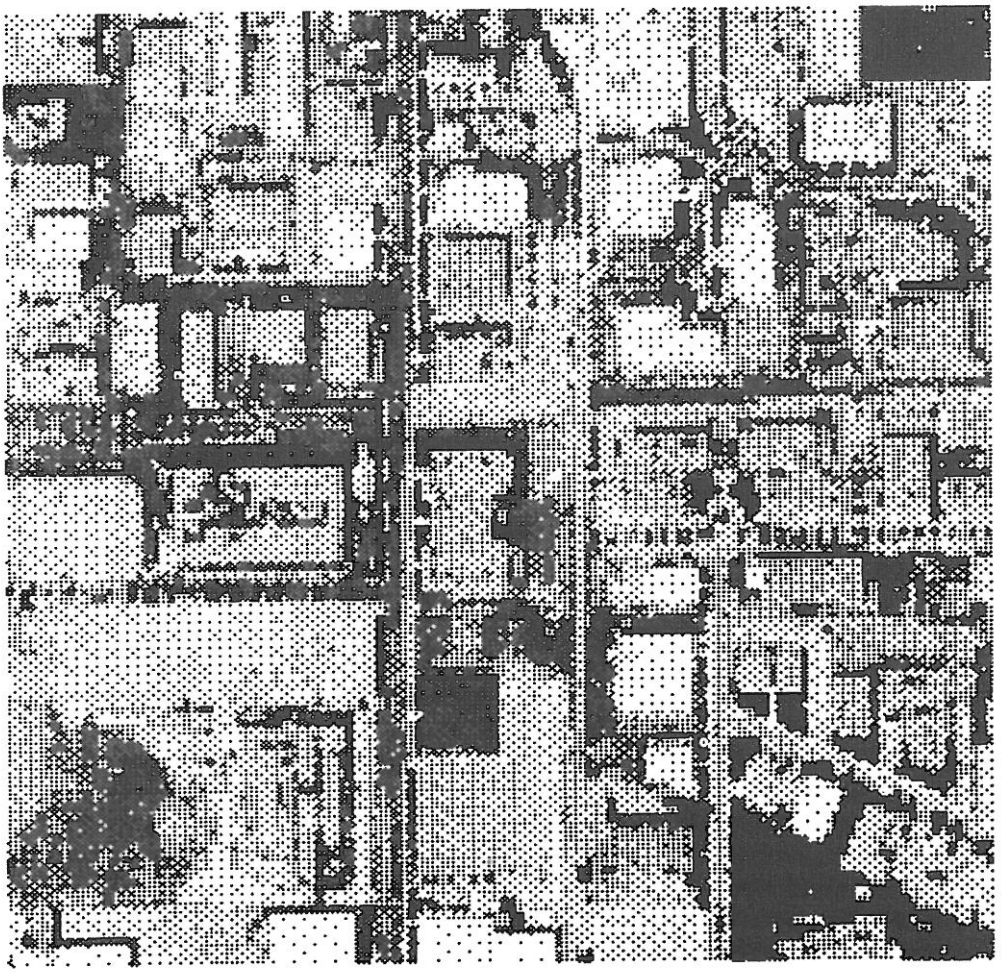


FIG. 9. A 256 × 256 aerial image of a commercial residential area.

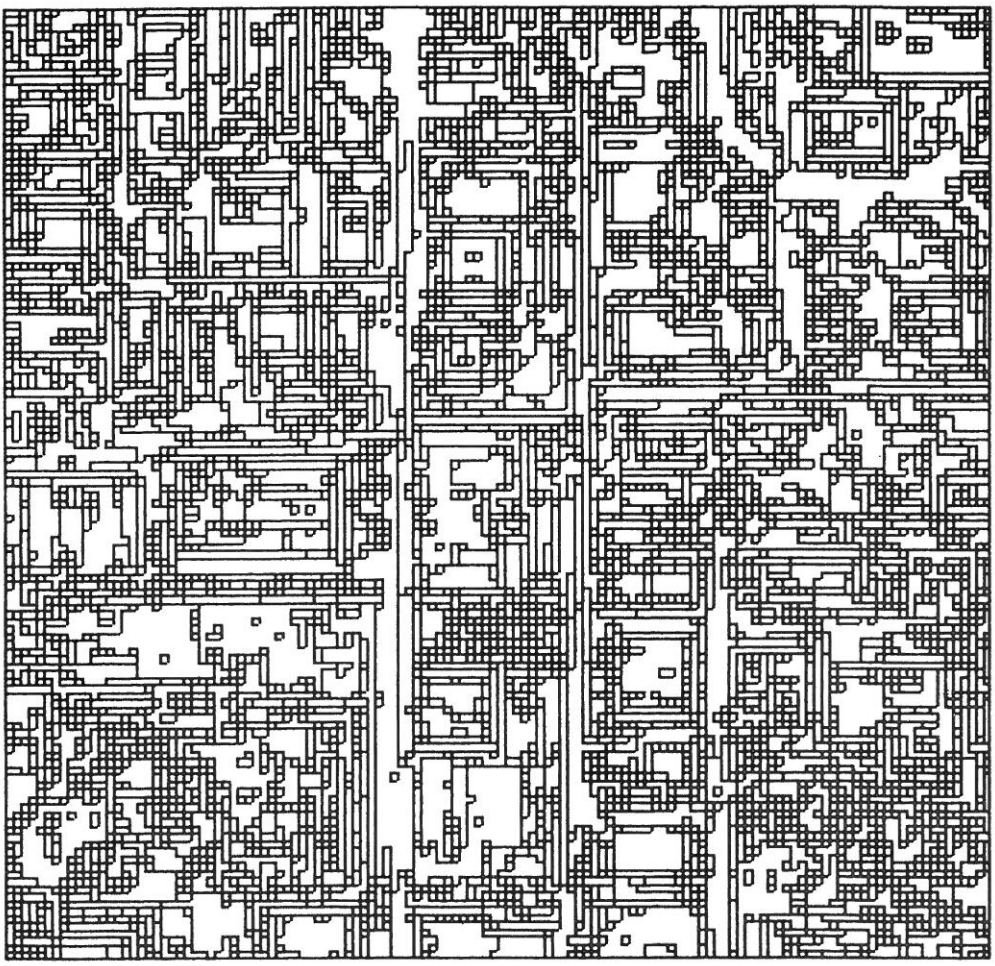


FIG. 10. The initial flat facet segmentation of the image of Fig. 9.

was done after five flat facet iterations of updating method 1, with and without a threshold. The results are shown in Figs. 11 and 12, respectively. As a comparison between the two kinds of facet iterations, we have examined those regions which correspond to the buildings in this image. We found that over 50% of the buildings were picked up closely as individual regions by both methods. Around 30 and 20% of the buildings were split into smaller units in Figs. 11 and 12, respectively. Close to 20% of the buildings were erroneously merged with their surroundings as a result of the regular facet iteration; around 10% were merged with their surroundings by the thresholded facet iteration.

As a comparison, similar experiments were performed on the initial segmentation produced from the slope facet model (Figs. 13–15). Results were found to be equally good when the buildings in the image were examined. As we can see from Figs. 10 and 13, the flat facet model produces a lot more initial regions. Thus the region growing process starting with the flat facet model is more expensive.

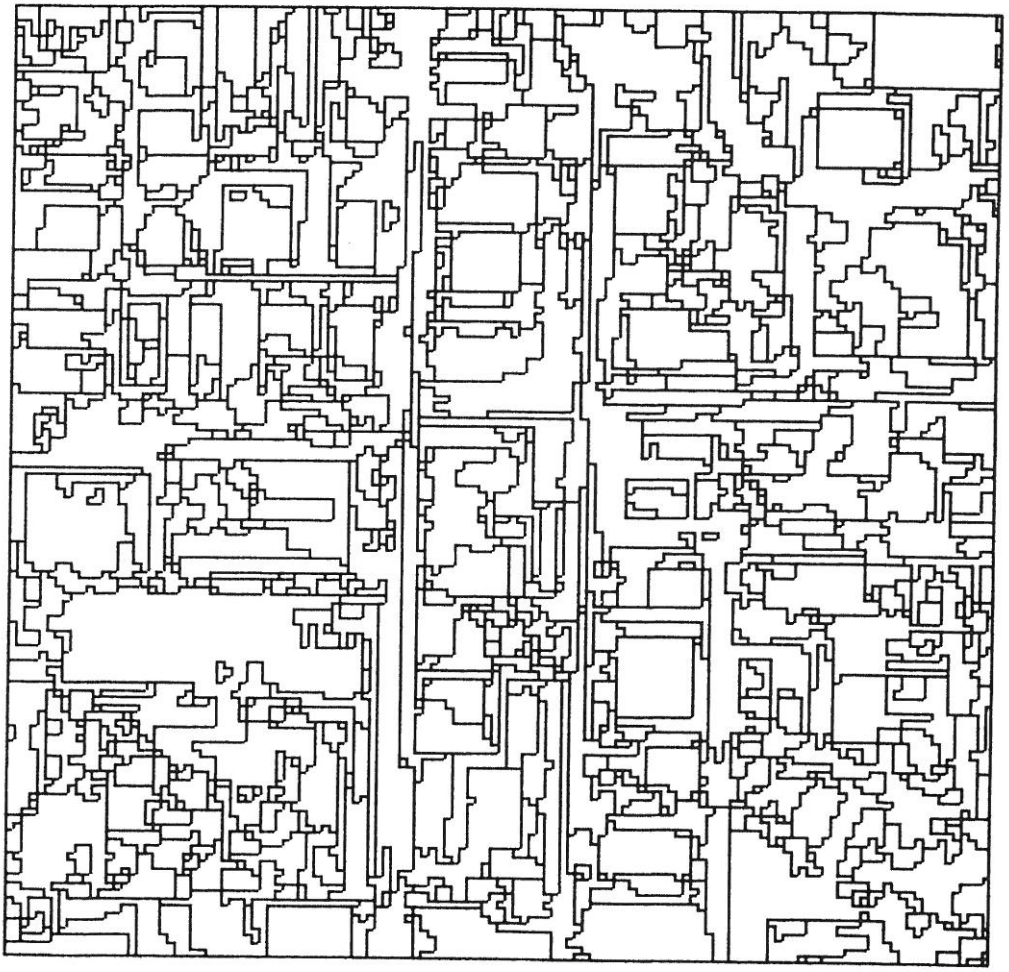


FIG. 11. The results of merging after five regular facet iterations of region growing, using replacement method 1 on the segmentation of Fig. 10, but incorporating the edge heuristic.

6. DISCUSSION AND CONCLUSION

It is worthwhile to compare our proposed region growing method to the split-and-merge technique of Horowitz and Pavlidis [6]. Our procedure starts with an initial segmentation into many small irregular regions where the pixels of a region belong to the same sloped plane and a significant edge exists between the planes of each pair of adjacent regions. The split-and-merge technique is defined as starting with an arbitrary partition of the image. However, the implementation of split-and-merge reported in [6] used square regions. Our procedure updates the property vectors of each region up to (or close to) a fixed point and then merges adjacent regions having similar property vectors. The resulting regions are again of arbitrary shape. The split-and-merge implementation merges four square blocks to produce a new square block or splits a single square block into four children (also square). Only after the splitting and merging is over does a grouping procedure take over to partition the

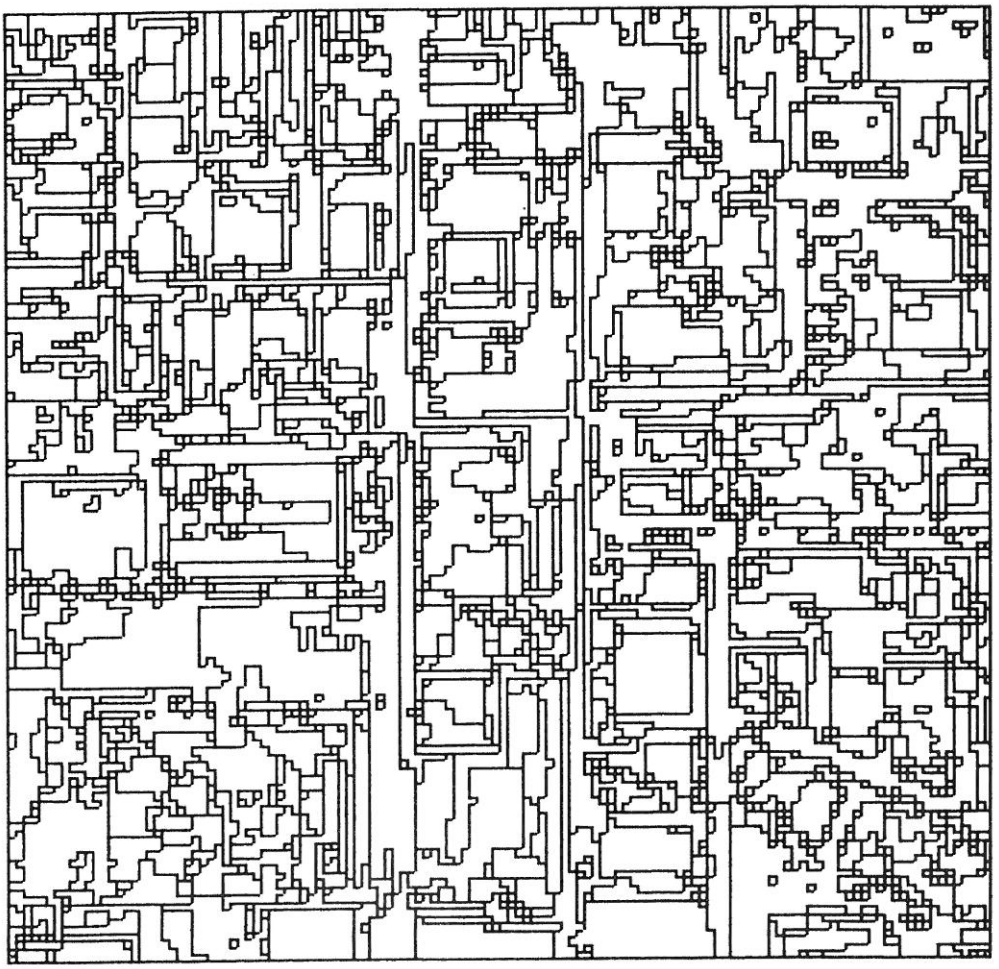


FIG. 12. The results of merging after five thresholded facet iterations of region growing, using replacement method 1 on the segmentation of Fig. 10, but incorporating the edge heuristic.

current cut set of the segmentation quad tree into equivalence classes that form the final segmentation.

Our procedure only merges, using arbitrary shaped regions. The Horowitz and Pavlidis procedure splits and merges using square regions and finally merges again to get more arbitrary regions. We feel that the use of arbitrary regions throughout is important in working with natural images; however, adding the possibility of splitting to our procedure might be worth trying.

We have presented, in this paper, a scheme for region growing. Merging regions using the mean property vector of the best neighborhood gave better results than the two other alternatives tried. The thresholded facet iteration, which was used to control the region growing process, introduced a larger number of regions, but had the advantage of separating nonhomogeneous neighborhoods of regions. More experiments are needed in order to understand how to select appropriate properties for the regions.

Due to the nature of the lower degree facet models, edge areas which have significant width become small regions. Applying the heuristic of eliminating small regions with high gradient values gave better results. Results show that both the flat and the slope facet models give good fits to the interiors of the regions in the image, but do not fit as well at the edges. Although the F tests for the higher-order facet models are mathematically more complicated, initial segmentations based upon higher-order facet models need to be attempted. Haralick [5] reports some success on edge detection using a local cubic facet model which is different from the piecewise facet model of [3]. Ways to incorporate the higher-order facet model edge detection technique into our region grower are worth exploring. We hope to address these issues in future papers.

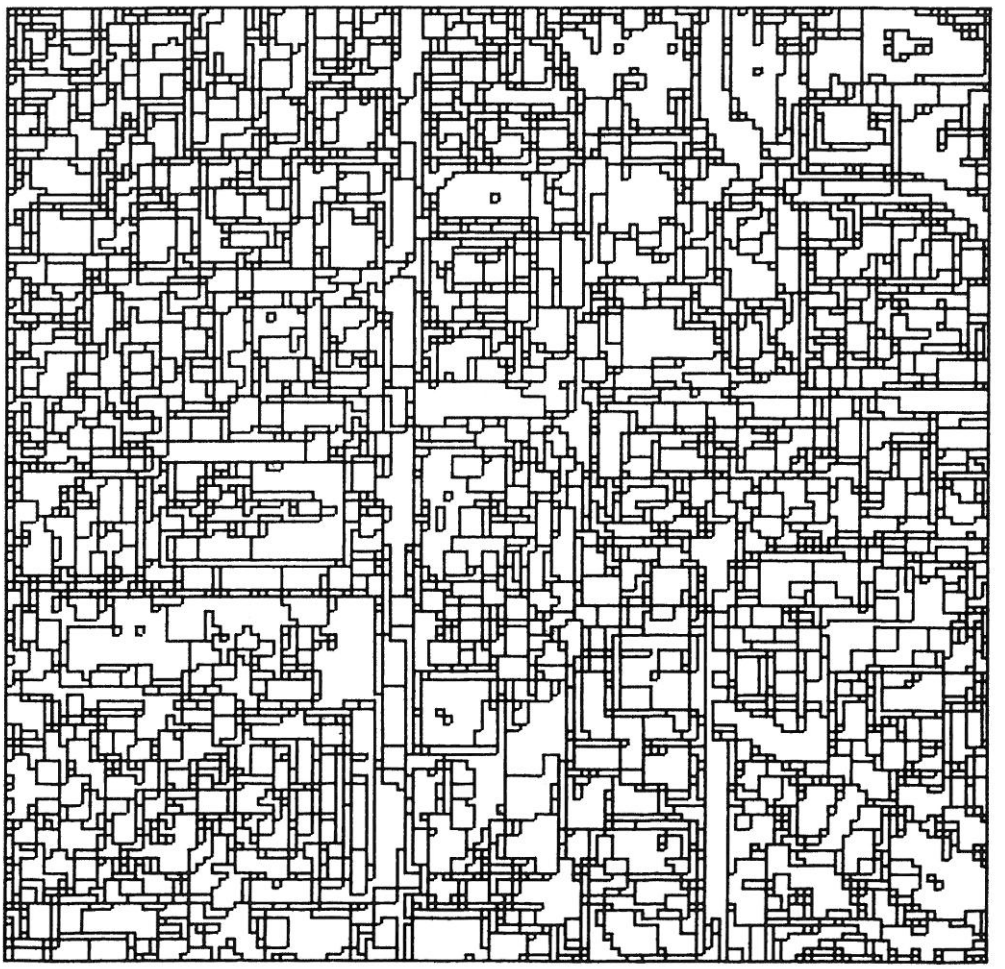


FIG. 13. The initial slope facet segmentation of the image of Fig. 9.

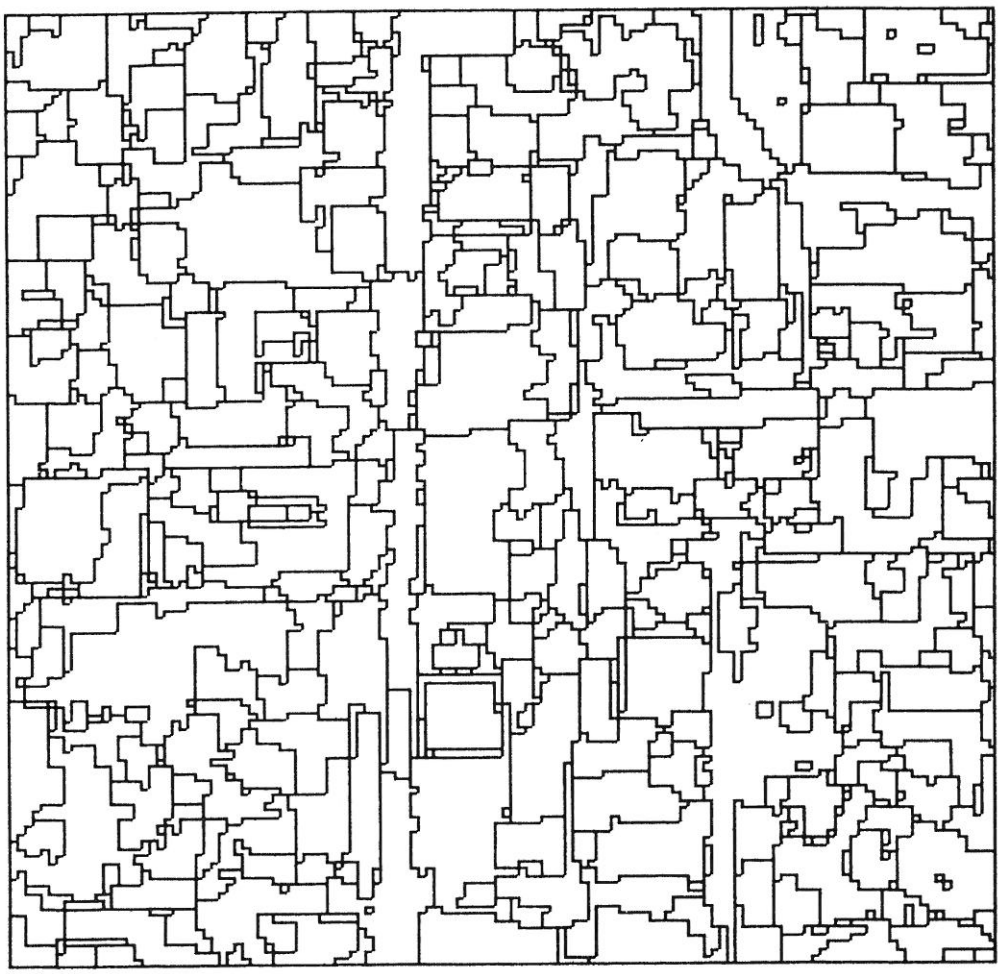


FIG. 14. The results of merging after five regular facet iterations of region growing, using replacement method 1 on the segmentation of Fig. 13, but incorporating the edge heuristic.

APPENDIX

In this appendix, we will prove the theorem on the convergence of the thresholded flat facet iteration.

Proof. We will tacitly use the lemmas for the flat facet iteration in [3], since they are trivially true for thresholded flat facet iteration also.

(1) The proof is by induction on m. Suppose $X^m(k) \leq X^m(k+1)$. Then from the definition of the $(m+1)$st iterates,

$$X^{m+1}(k) \leq \frac{X^m(k) + X^m(k+1)}{2} \leq X^{m+1}(k+1).$$

Similarly, if $X^m(k) \geq X^m(k+1)$, then

$$X^{m+1}(k) \geq \frac{X^m(k) + X^m(k+1)}{2} \geq X^{m+1}(k+1).$$

(2) Since order relationships between neighboring points are preserved, local extrema must remain extrema.

(3) If $X^0(k)$ is a local minimum, then by (2) $X^m(k)$ remains a local minimum for all m. By construction, $X^{m+1}(k)$ must either be the same as $X^m(k)$ or greater than $X^m(k)$. Therefore the sequence $X^m(k)$, $m = 0, 1, 2, \ldots$ must be monotonically increasing, and by property (1), $\min_i X^0(i) \leq X^m(j)$ for all j and m. If $X^0(k)$ is a local maximum, then $X^m(k)$ is also a local maximum for all m. From the definition of the thresholded flat facet iteration, $X^{m+1}(k)$ must either be the same as $X^m(k)$ or less than $X^m(k)$. Therefore the sequence $X^m(k)$ is monotonically decreasing, and $X^m(j) \leq \max_i X^m(i) \leq \max_i X^0(i)$ for all j and m.

(4) If $X^m(k)$ is a local extremum, then the result follows from the proof of (3). For $X^m(k)$ not a local extremum, the proof is by induction on m. There are two possibilities: either

$$X^{m-1}(k-1) < X^{m-1}(k) < X^{m-1}(k+1)$$

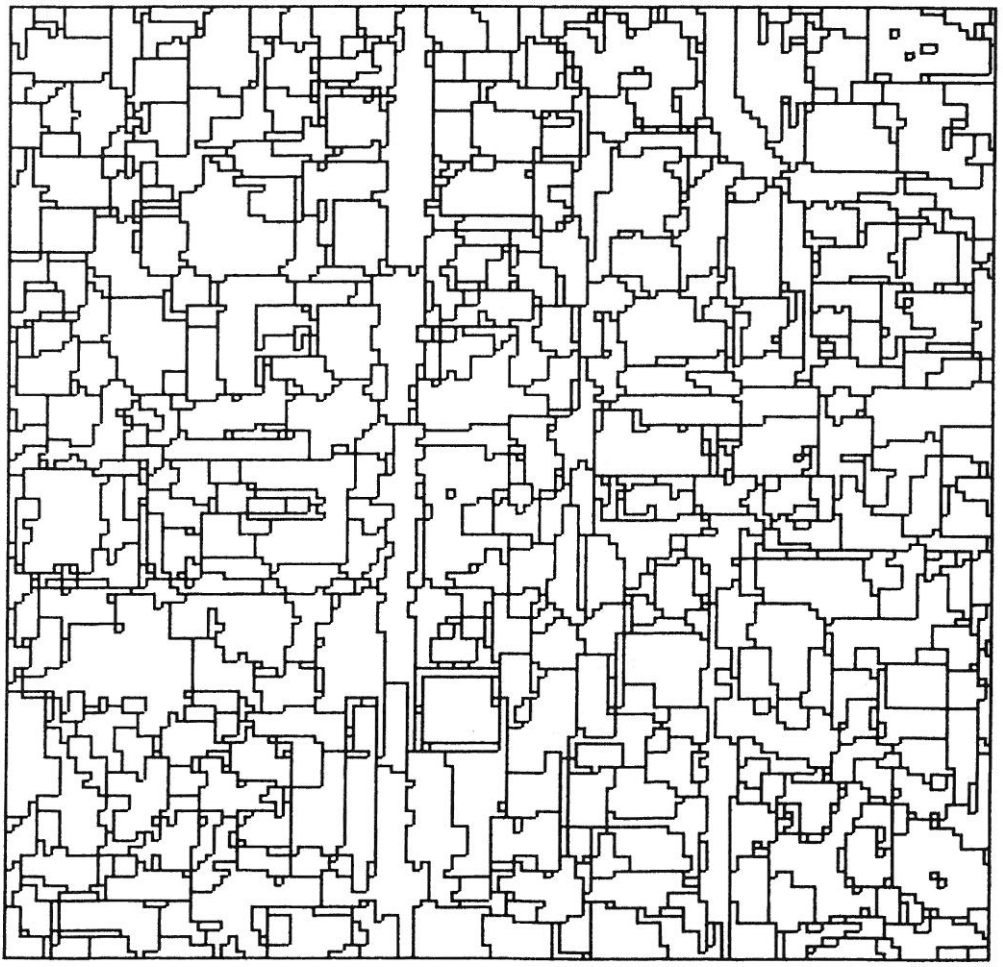


FIG. 15. The results of merging after five thresholded facet iterations of region growing, using replacement method 1 on the segmentation of Fig. 13, but incorporating the edge heuristic.

or

$$X^{m-1}(k-1) > X^{m-1}(k) > X^{m-1}(k+1).$$

Since the second case follows from the first by reversing the indices, it may be assumed without loss of generality that $X^{m-1}(k-1) < X^{m-1}(k) < X^{m-1}(k+1)$.

First the following observation must be established:

Claim. If $\min(X^r(k) - X^r(k-1), X^r(k+1) - X^r(k)) > e$, then $\min(X^j(k) - X^j(k-1), X^j(k+1) - X^j(k)) > e$, for $j = 0, 1, \ldots, r-1$.

Proof of claim. An equivalent proposition is that if $\min(X^s(k) - X^s(k-1), X^s(k+1) - X^s(k)) \le e$, then $\min(X^{s+1}(k) - X^{s+1}(k-1), X^{s+1}(k+1) - X^{s+1}(k)) \le e$. Suppose that $X^s(k) - X^s(k-1) \le \min(e, X^s(k+1) - X^s(k))$. Recalling the assumption $X^s(k-1) < X^s(k) < X^s(k+1)$ and using the definition of facet iteration gives

$$X^s(k-1) - \frac{X^s(k) - X^s(k-1)}{2} \le X^{s+1}(k-1) \le X^{s+1}(k) = \frac{X^s(k) + X^s(k-1)}{2}$$

which implies

$$\begin{aligned} X^{s+1}(k) - X^{s+1}(k-1) &\le \frac{X^s(k) + X^s(k-1)}{2} - \left[X^s(k-1) - \frac{X^s(k) - X^s(k-1)}{2} \right] \\ &= X^s(k) - X^s(k-1) \le e. \end{aligned}$$

Similarly $X^s(k+1) - X^s(k) < X^s(k) - X^s(k-1)$ and $X^s(k+1) - X^s(k) \le e$ implies $X^{s+1}(k+1) - X^{s+1}(k) \le e$ which establishes the claim.

Now if $\min(X^{m-1}(k+1) - X^{m-1}(k), X^{m-1}(k) - X^{m-1}(k-1)) > e$, then by the preceding observation $\min(X^j(k+1) - X^j(k), X^j(k) - X^j(k-1)) > e$ for $j = 0, 1, \ldots, m-2$ and $X^0(k) = \cdots = X^{m-1}(k) = X^m(k)$, so regardless of where $X^{m+1}(k)$ is the sequence is monotone. Henceforth assume that $\min(X^{m-1}(k+1) - X^{m-1}(k), X^{m-1}(k) - X^{m-1}(k-1)) \le e$.

Suppose that $X^{m-1}(k) \le X^m(k)$. Then just using the definition of the facet iteration and assumed inequalities yields

$$\begin{aligned} X^m(k-1) &\le \frac{X^{m-1}(k-1) + X^{m-1}(k)}{2} < X^{m-1}(k) \\ &\le X^m(k) = \frac{X^{m-1}(k) + X^{m-1}(k+1)}{2} \\ &\le X^m(k+1) \le X^{m-1}(k+1) + \frac{X^{m-1}(k+1) - X^{m-1}(k)}{2}. \end{aligned}$$

Using pieces of this inequality gives

$$\begin{aligned} X^m(k) - X^m(k-1) &\geq \frac{X^{m-1}(k) + X^{m-1}(k+1)}{2} - \frac{X^{m-1}(k-1) + X^{m-1}(k)}{2} \\ &= \frac{X^{m-1}(k) - X^{m-1}(k-1)}{2} \\ &\quad + X^{m-1}(k+1) - \frac{X^{m-1}(k+1) + X^{m-1}(k)}{2} \\ &> \frac{X^{m-1}(k+1) - X^{m-1}(k)}{2} \\ &\quad + X^{m-1}(k+1) - X^m(k) \geq X^m(k+1) - X^m(k), \end{aligned}$$

which establishes that $X^{m+1}(k) = \dfrac{X^m(k) + X^m(k+1)}{2} \geq X^m(k)$.

To complete the proof, suppose that $X^{m-1}(k) \geq X^m(k)$. Then

$$\begin{aligned} X^m(k+1) &\geq \frac{X^{m-1}(k) + X^{m-1}(k+1)}{2} \\ &> X^{m-1}(k) \geq X^m(k) = \frac{X^{m-1}(k-1) + X^{m-1}(k)}{2} \\ &\geq X^m(k-1) \geq X^{m-1}(k-1) - \frac{X^{m-1}(k) - X^{m-1}(k-1)}{2} \end{aligned}$$

which in turn implies

$$\begin{aligned} X^m(k) - X^m(k-1) &\leq \frac{X^{m-1}(k-1) + X^{m-1}(k)}{2} \\ &\quad - \left[X^{m-1}(k-1) - \frac{X^{m-1}(k) - X^{m-1}(k-1)}{2} \right] \\ &< \frac{X^{m-1}(k) - X^{m-1}(k-1)}{2} + \frac{X^{m-1}(k+1) - X^{m-1}(k)}{2} \\ &= \frac{X^{m-1}(k) + X^{m-1}(k+1)}{2} \\ &\quad - \frac{X^{m-1}(k-1) + X^{m-1}(k)}{2} \leq X^m(k+1) - X^m(k). \end{aligned}$$

Hence the next iteration satisfies

$$X^m(k) \geq \frac{X^m(k-1) + X^m(k)}{2} = X^{m+1}(k),$$

which completes the inductive proof.

(5) For fixed k, $X^m(k)$ is a bounded monotone sequence by (3) and (4).

Therefore, by the Bolzano–Weierstrass theorem, $\lim_{m\to\infty} X^m(k) = X^\infty(k)$ exists.

Note that for the flat facet iteration, every point $X^{\infty}(k)$ is a local extremum. This is not necessarily true for *thresholded* flat facet iteration, since wild points may remain unchanged throughout the iteration. In other words, every point is not necessarily attracted to a "strong and consistent neighbor."

REFERENCES

1. C. Brice, and C. Fennema, Scene analysis using regions, *Artif. Intell.* **1**, 1970, 205–226.
2. L. S. Davis, A. Rosenfeld, and J. S. Weszka, Region extraction by averaging and thresholding, *IEEE Trans. Syst. Man Cybern.* **SMC-5**, 1975, 383–388.
3. R. M. Haralick and L. Watson, A facet model for image data, *Computer Graphics Image Processing* **15**, 1981, 113–129.
4. R. M. Haralick, Edge and region analysis for digital image data, *Computer Graphics Image Processing* **12**, 1980, 60–73.
5. R. M. Haralick, The digital step edge, *IEEE Trans. Pattern Anal. Mach. Intell.*, in press.
6. S. L. Horowitz and T. Pavlidis, Picture Segmentation by a Directed Split-and-Merge Procedure, Proceedings, 2nd International Joint Conference on Pattern Recognition, pp. 424–433, 1974.
7. T. Pavlidis, Segmentation of pictures and maps through functional approximation, *Computer Graphics Image Processing* **1**, 1972, 360–372.
8. T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, Berlin, 1977.
9. D. L. Milgram, Region extraction using convergent evidence, *Computer Graphics Image Processing* **11**, 1979, 1–12.
10. J. L. Muerle and D. C. Allen, Experimental evaluation of techniques for automatic segmentation of objects in a complex scene, in *Pictorial Pattern Recognition* (G. C. Cheng *et al.*, Eds.), pp. 3–13, Thompson, Washington, 1968.
11. M. Nagao and T. Matsuyama, Edge preserving smoothing, *Computer Graphics Image Processing* **9**, 1979, 394–407.
12. S. Tanimoto and T. Pavlidis, A hierarchical data structure for image processing, *Computer Graphics Image Processing* **4**, 1975.
13. F. Tomita and S. Tsuji, Extraction of multiple regions by smoothing selected neighborhoods, *IEEE Trans. Syst. Man Cybern.* **SMC-7**, 1977, 107–109.
14. D. H. D. West, Updating mean and variance estimates: An improved method, *Commun. ACM* **22**, 1979, 532–535.
15. S. W. Zucker, Region growing: Childhood and adolescence, *Computer Graphics Image Processing* **5**, 1976, 382–399.