COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 29, 286–295 (1985)

# Image Representation Using Voronoi Tessellation*

NARENDRA AHUJA AND BYONG AN

*Coordinated Science Laboratory, University of Illinois, Urbana, Illinois 61801*

AND

BRUCE SCHACHTER

*Columbia, Maryland*

Received August 4, 1983; revised June 21, 1984

The cells of the Voronoi tessellation are used as primitives to represent image regions. The tessellation is derived from a Poisson point process. The random shapes of cells make the representation attractive for secure transmission. © 1985 Academic Press, Inc.

## 1. INTRODUCTION

The major approaches to image representation [1] in terms of its constituent regions may be divided into two broad categories: (1) those which specify the borders of the regions and (2) those which describe their interiors. Most of the approaches and the more interesting ones, belong to the latter category. This may be attributed to the increased dimensionality of information (regions, rather than curves) to be represented. An important subclass of these methods, called medial axis transforms (MAT) [1], involves representation of the regions by a set of maximal 2-dimensional blocks, of a fixed shape. Each maximal block lies completely within a single region, and is not contained in any other such block. The blocks may have any size and may be placed anywhere in the image as determined by the locations of the regions. The simplest choice for the block shape is the circle as originally proposed by Blum [2]. Squares, or, equivalently, diamonds, may be more appropriate for grid images [3].

If the purpose of the image representation is data compression only, the choice of the block shape is not critical for an arbitrary image with unknown region geometry. However, if the image MAT must be transmitted and security is a concern along with compression, the choice of a fixed shape for blocks may not be ideal. A collection of randomly shaped blocks may provide an MAT coding for the image regions that is less likely to be broken.

This paper describes an MAT that uses irregular polygonal blocks whose shapes and sizes are determined by a planar random process. Section 2 defines the Voronoi tessellation of the plane generated by a planar random point process; the cells of the tessellation are used as the blocks of the MAT. Section 3 discussed the derivation of the MAT and a scheme for secure transmission of a given piecewise uniform image. Experimental results on two images are also given. Section 4 describes an adaptive Voronoi partitioning scheme that uses variable spatial resolution to improve the

*This work was supported in part by the National Science Foundation under Grant ECS-8106008



0734-189X/85 $3.00

accuracy of the representation in the vicinity of region boundaries. Section 5 presents concluding remarks.

## 2. VORONOI TESSELLATION

Suppose that we are given a set $S$ of three or more points in the Euclidean plane. Assume that these points are not all collinear, and that no four points are cocircular. Consider an arbitrary pair of points $P$ and $Q$. The bisector of the line joining $P$ and $Q$ is the locus of points equidistant from both $P$ and $Q$ and divides the plane into two halves. The half plane $H_P^Q(H_Q^P)$ is the locus of points closer to $P(Q)$ than to $Q(P)$. For any given point $P$ a set of such half planes is obtained for various choices of $Q$. The intersection $\cap_{Q \in S,\, Q \neq P} H_P^Q$ defines a polygonal region consisting of points closer to $P$ than to any other point. Such a region is called the Voronoi [4] polygon associated with the point.

Voronoi polygons may be viewed as the result of a growth process. Assume all points (nuclei) simultaneously start a uniform outward growth along a circular frontier. The growth stops at points of contact between any two circles which then expand into straight line segments along which growth frontiers meet and freeze. An edge continues elongating until it encounters the border of a third expanding circle. Eventually, only the circles whose nuclei are on the convex hull of $S$ are still expanding. Each of the remaining nuclei is contained in exactly one convex polygon. The set of complete polygons is called the Voronoi diagram of $S$ [5]. The Voronoi diagram together with the incomplete polygons in the convex hull define a Voronoi tessellation of the entire plane. An example of Voronoi tessellation is shown in Fig. 1; $O(N \log N)$ algorithms to construct the Voronoi tessellation of $N$ points are given by Shamos and Hoey [5] and Lee [6].

## 3. IMAGE REPRESENTATION AND SECURE TRANSMISSION

Traditionally, for obtaining the MAT of a given image, the locations and sizes of the blocks are determined by the requirement that the block sizes be locally maximal. Thus, the image regions determine the block locations. Since our goal in this paper is to obtain a representation that is relatively secure for transmission, we choose a random set of locations to generate the Voronoi tessellation. In particular, we use the Poisson point process to randomly drop the nuclei of the Voronoi

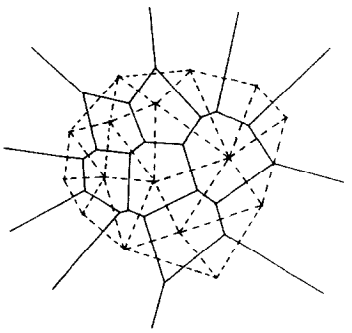

FIG. 1. Voronoi tessellation defined by a given set of points. Dotted lines show the corresponding Delaunay tessellation.

polygons. On a grid, a discrete version of the planar Poisson point process must be used.

### *3.1. Digital Poisson Point Process*

In the Euclidean plane, a process is said to be a homogeneous Poisson point process of intensity $\lambda$ iff:

(i) the number of points in any region of area $A$ has a Poisson distribution with parameter $\lambda A$, and

(ii) the random variables corresponding to the numbers of points in disjoint regions are independent.

Any finite region in the Euclidean plane corresponds to a finite number of nodes on an overlying grid. Whereas the actual Poisson process can drop an arbitrarily large number of points in such a region with positive probability, the digital Poisson process can drop at most as many points as there are grid nodes in the region. Each node is at the center of an implicit square of unit area. A node will be selected by the digital Poisson process whenever at least one point is dropped by the Poisson process in the square. From property (i) we have

$$\begin{aligned} p &= Pr\ (\text{a given node on the grid is selected by the digital} \\ &\qquad \text{Poisson process of intensity } \lambda) \\ &= Pr\ (\text{a square of unit area in the Euclidean plane contains at} \\ &\qquad \text{least one point dropped by a process of intensity } \lambda) \\ &= 1 - e^{-\lambda}. \end{aligned}$$

A digital Poisson point process is thus a binomial process with parameter $p = 1 - e^{-\lambda}$, and $Pr\ \{n$ points fall in a region consisting of $N$ grid nodes$\} = \binom{N}{n} p^n (1-p)^{N-n}$. To simulate a Poisson process on the grid thus amounts to making a binary decision at each of its nodes.

### *3.2. Representation*

We will consider representing a facet image, where each region is characterized by a constant gray level or by a mean gray level (color). A set of points is dropped on the image which is then used to derive the Voronoi tessellation. Each cell is examined for homogeneity. If the cell lies completely inside a region (or background) then the region's color is assigned to the cell nucleus. The cells that contain pixels from more than one type of region, i.e., the cells covering region borders, cannot be given a unique color. The nucleus of each such cell is assigned the color of the majority of the pixels in the cell. The set of nuclei along with the assigned colors constitute the representation of the image.

#### *3.2.1. Experimental Results*

Representations were derived for two $128 \times 128$ images shown in Fig. 2. The images were first thresholded to make the cell homogeneity test simple. The digital Poisson process was used to drop points over a range of intensity values. The points were then assigned colors as described above. Each of the resulting representations

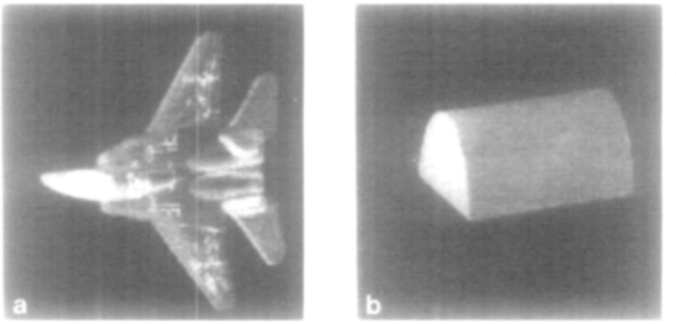


FIG. 2. Two images used in our experiments.

was used to reconstruct the original image by coloring the cells in the tessellation uniformly with the color values assigned to their nuclei. Clearly, the reconstruction is only an approximation to the original as the border cells, which contain more than one color, are assigned the majority color. In fact, the edges of the reconstructed image come from the edges of the tessellation, i.e., the border of a given region in the image is approximated by a sequence of connected tessellation edges. Thus the pixels having minority color in a cell receive incorrect color in the reconstruction. Table 1 lists the experimental results on the two images. $\lambda$ values were used in the range 0.01 to 0.05, with increments of 0.01. Intensity values larger than 0.05 generated points that occasionally fell too close (on neighboring grid nodes). This caused problems in obtaining the tessellation. Values smaller than 0.01 gave too coarse a tessellation and hence were not used. Five different sets of nuclei were dropped for each $\lambda$ value, by varying the seed of the pseudo random number generator. Thus, five different, statistically identical representations were derived for each $\lambda$ value. On each representation the following observations were made: the total number of nuclei (= number of cells), the number of border cells, and the number of error pixels in the reconstruction. Table 1 lists these observations along with the percentage numbers of border cells and error pixels. The percentage number of error pixels is a measure of the mean squared error in the reconstructed image. Figures 3 and 4 show the error pixels in typical reconstructions of the images shown in Fig. 2. The concentration of errors near region borders and the reduction in the amount of error with increasing intensity can be seen. Figures 5 and 6 show the edges of the tessellations that constitute the region borders in the representations, i.e., the tessellation edges separating the Voronoi cells of different colors.

*3.2.2. Data Compression*

To transmit an $n \times n$ image, $n^2$ color values need to be transmitted, one for each pixel. If the chosen Voronoi representation uses $N$ nuclei, then the information to be transmitted would consist of $N$ colors, the seed for the pseudo-random number generator and the value of $\lambda$. The number of bits necessary to transmit the seed and $\lambda$ can be ignored in comparison with the number of bits necessary to transmit the $N$ colors. Thus a data compression of $n^2/N$ is achieved as a result of using the Voronoi representation. The last column of Table 1 lists the compression rates achieved for various values of mean squared error in the reconstructions.

TABLE 1

Voronoi Representation of the $n \times n$ ($n = 128$) Images in (a) Fig. 2a and (b) Fig. 2b

| | Intensity ($\lambda$) | Total Number of Cells ($N$) | Number of Border Cells ($N_b$) | Number of Error Pixels | Average Percentage Number of Border Cells | Average Percentage Number of Error Pixels = Percentage Mean Squared Error | Average Data Compression ($= n^2/\bar{N}$) |
|---|---|---|---|---|---|---|---|
| (a) | | 154 | 54 | 1099 | | | |
| | | 164 | 57 | 1084 | | | |
| | 0.01 | 159 | 51 | 1241 | 33.6 | 7.20 | 104.09 |
| | | 155 | 51 | 1238 | | | |
| | | 155 | 51 | 1235 | | | |
| | | 277 | 79 | 921 | | | |
| | | 302 | 83 | 757 | | | |
| | 0.02 | 304 | 85 | 871 | 27.3 | 5.08 | 54.04 |
| | | 299 | 76 | 832 | | | |
| | | 334 | 90 | 783 | | | |
| | | 491 | 95 | 752 | | | |
| | | 452 | 99 | 716 | | | |
| | 0.03 | 453 | 92 | 675 | 20.3 | 4.42 | 34.41 |
| | | 488 | 109 | 715 | | | |
| | | 497 | 88 | 761 | | | |
| | | 639 | 101 | 615 | | | |
| | | 660 | 112 | 635 | | | |
| | 0.04 | 637 | 108 | 711 | 17.6 | 3.95 | 26.11 |
| | | 600 | 114 | 630 | | | |
| | | 601 | 114 | 646 | | | |
| | | 769 | 124 | 533 | | | |
| | | 771 | 132 | 561 | | | |
| | 0.05 | 779 | 122 | 590 | 16.1 | 3.52 | 21.06 |
| | | 787 | 121 | 608 | | | |
| | | 783 | 128 | 591 | | | |
| (b) | | 145 | 25 | 815 | | | |
| | | 167 | 27 | 753 | | | |
| | 0.01 | 173 | 31 | 536 | 18.4 | 4.17 | 101.39 |
| | | 154 | 31 | 626 | | | |
| | | 169 | 35 | 688 | | | |
| | | 320 | 43 | 484 | | | |
| | | 255 | 36 | 527 | | | |
| | 0.02 | 331 | 47 | 480 | 14.1 | 2.95 | 53.68 |
| | | 304 | 45 | 437 | | | |
| | | 316 | 44 | 488 | | | |

TABLE 1—*Continued*

| Intensity ($\lambda$) | Total Number of Cells ($N$) | Number of Border Cells ($N_b$) | Number of Error Pixels | Average Percentage Number of Border Cells | Average Percentage Number of Error Pixels = Percentage Mean Squared Error | Average Data Compression ($= n^2/\bar{N}$) |
|---|---|---|---|---|---|---|
| | 425 | 50 | 381 | | | |
| | 472 | 56 | 433 | | | |
| 0.03 | 473 | 54 | 411 | 11.6 | 2.47 | 35.16 |
| | 472 | 51 | 426 | | | |
| | 488 | 58 | 371 | | | |
| | 646 | 69 | 326 | | | |
| | 657 | 67 | 335 | | | |
| 0.04 | 638 | 73 | 325 | 10.5 | 2.03 | 25.54 |
| | 635 | 62 | 354 | | | |
| | 632 | 66 | 325 | | | |
| | 784 | 65 | 324 | | | |
| | 793 | 67 | 278 | | | |
| 0.05 | 789 | 80 | 311 | 8.8 | 1.92 | 21.11 |
| | 711 | 62 | 374 | | | |
| | 803 | 70 | 284 | | | |

*Note.* Five samples of the representation were generated for each value of $\lambda$. Here $\bar{A}$ denotes the average value of $A$.

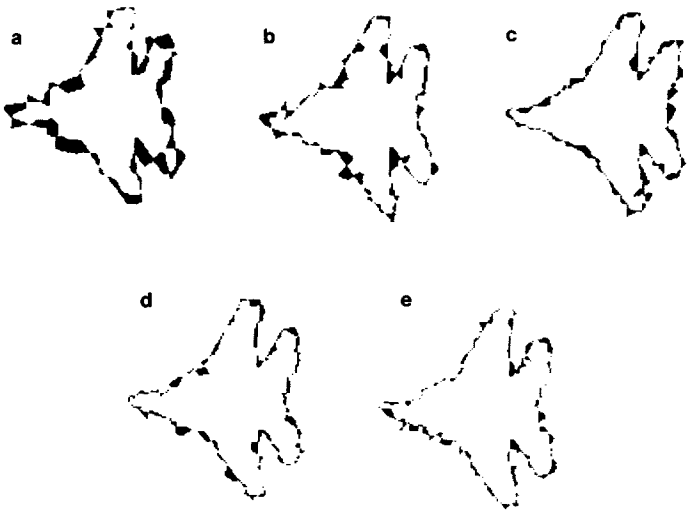


FIG. 3. Pixels having incorrect colors in the reconstruction of the image shown in Fig. 2a: (a) $\lambda = 0.01$ (b) $\lambda = 0.02$; (c) $\lambda = 0.03$; (d) $\lambda = 0.04$; (e) $\lambda = 0.05$.

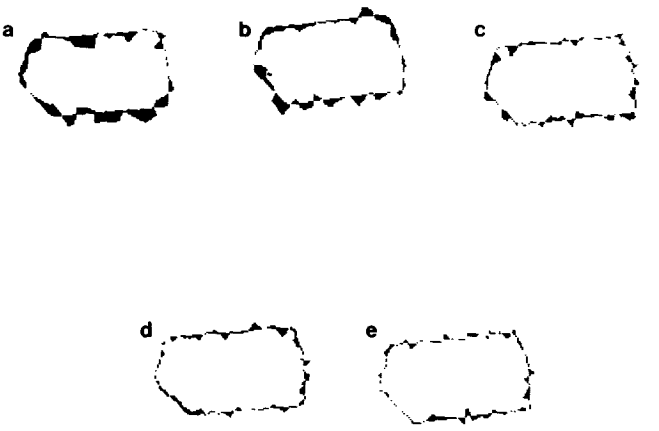


FIG. 4. Pixels having incorrect colors in the reconstruction of the image shown in Fig. 2b: (a) $\lambda = 0.01$; (b) $\lambda = 0.02$; (c) $\lambda = 0.03$; (d) $\lambda = 0.04$; (e) $\lambda = 0.05$.

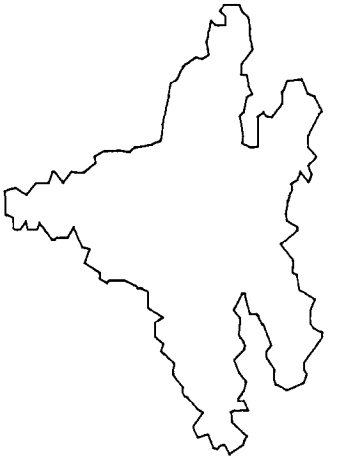

FIG. 5. Edges generated by the representation of the image in Fig. 2a for $\lambda = 0.05$.

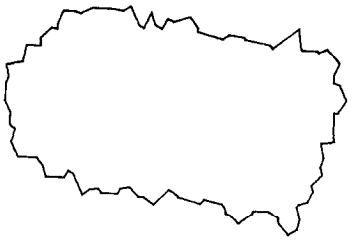

FIG. 6. Edges generated by the representation of the image of Fig. 2b for $\lambda = 0.05$.

*3.2.3. Secure Transmission*

The representation obtained above is particularly suitable for secure transmission. The locations of the nuclei need not be transmitted explicitly. As long as both the sender and the receiver use the same pseudo-random number generator, only the seed used to derive the set of nuclei providing an acceptable representation need be transmitted. The coordinates of the nuclei can be generated by the receiver in the same way as at the transmitting end, e.g., along a raster scan. Along with the seed, an ordered list of colors associated with the nuclei must also be sent to complete the representation.

Note that the error of representation as discussed in subsec. 3.1.1 could conceivably be reduced by manipulating the locations of the nuclei near region borders, such that the tessellation edges better approximate image edges. However, this would require including in the representation explicit information about the region shapes, which would make the transmission less secure.

## 4. ADAPTIVE VORONOI REPRESENTATION

The errors in the Voronoi representation result from the pixels in the vicinity of edges. The number of error pixels goes down with increasing intensity of the Poisson point process since the cell size is reduced. However, the total number $N$ of cells increases, thus decreasing the data compression factor. Note that to reduce the error we only need to reduce the size of the border cells. The reduction in size of the remaining (interior) cells offers no advantage. Thus, it would be useful to confine the effect of increased intensity values to border areas. Once again, we do not want to include any information regarding border locations in the representation.

We accomplish selective increase in intensity by first dropping points at high intensity throughout the image. Now, those cells which are homogeneous and are surrounded by homogeneous cells occur deep in the interiors of regions, and hence can be deleted. The image area covered by such a cell can then be occupied by its neighboring cells which grow larger. To identify the deleted nuclei, a special marker is transmitted as the color of the nucleus. At the receiver, such nuclei are deleted from those generated by the random number generator, before the Voronoi tessellation is obtained, Note that the number of bits assigned to the marker can be minimal, perhaps one. Thus the deletion of the large number of interior cells results in direct savings in the total number of transmitted bits, without reducing the spatial resolution in the border areas. The reduction in the required data rate is proportional to the gray level resolution, i.e., the number of bits used to represent color. Let $N_i$ denote the number of the deleted, interior cells, and let $B$ denote the number of color bits per pixel. Then, to transmit an interior (noninterior) nucleus requires 1 ($B + 1$) bits. Thus, the total number of bits required to transmit the $n \times n$ image is $N_i + (N - N_i)(B + 1)$, giving a data compression rate of

$$\frac{n^2B}{N_i + (N - N_i)(B + 1)} = \frac{n^2B}{(N - N_i)B + N}.$$

Thus, there is improvement in data compression over the nonadaptive case whenever $N_iB > N$. This will often be the case for blocky images as there would be a large number of interior cells. If $(N - N_i)B \ll N$ then the data compression rate is $n^2B/N$. Table 2 lists for the adaptive representation, the number of noninterior cells

TABLE 2

Averages of Experimental Results for the Adaptive Representation of the Images in (a) Fig. 2a and (b) Fig. 2b

| | Intensity ($\lambda$) | Average Total Number of Cells, From Table 1 ($\bar{N}$) | Average Number of Noninterior Cells ($\bar{N} - \bar{N}_i$) | Average Percentage Number of Error Pixels | Average Data Compression $\frac{n^2 B}{(\bar{N} - \bar{N}_i)B + \bar{N}}$ |
|---|---|---|---|---|---|
| (a) | 0.01 | 157.40 | 125.60 | 7.20 | 107.90 |
| | 0.02 | 303.20 | 275.60 | 5.08 | 50.23 |
| | 0.03 | 476.20 | 416.20 | 4.50 | 33.06 |
| | 0.04 | 627.40 | 529.40 | 4.02 | 25.84 |
| | 0.05 | 777.80 | 663.40 | 3.58 | 20.65 |
| (b) | 0.01 | 161.60 | 96.80 | 4.17 | 132.41 |
| | 0.02 | 305.20 | 183.00 | 2.95 | 70.05 |
| | 0.03 | 466.00 | 247.20 | 2.52 | 50.43 |
| | 0.04 | 641.60 | 334.80 | 2.07 | 37.09 |
| | 0.05 | 776.00 | 388.20 | 1.95 | 31.65 |

*Note.* Here $\bar{A}$ denotes the average value of $A$.

and percentage number of error pixels, and the value of the data compression rate for $B = 6$. The numbers of error pixels here should have been the same as in Table 1. The small differences present result from the random manner in which the pixels along a tessellation edge separating an interior cell and a border cell are assigned to either of the two cells. These pixels may assume correct or incorrect color and thus cause a small, random change in the total number of error pixels compared to the nonadaptive case. The image in Fig. 2a is not sufficiently compact to benefit from the extra overhead involved in the adaptive representation, as can be seen from the average percentage numbers of border cells and the data compression rates in Tables 1a and 2a, respectively. For Fig. 2b, on the other hand, the adaptive representation provides higher compression rates for the same mean squared errors.

## 5. CONCLUDING REMARKS

We have described a method of image representation based on the Voronoi tessellation of the image defined by a randomly distributed set of points. The representation is particularly useful for secure transmission of images. The Voronoi polygons are used as uniformly colored, randomly shaped blocks which fit together as in a jigsaw puzzle, to provide a mosaic approximation to the given piecewise constant image. This scheme resembles the MAT representation in that it uses certain primitive shapes to approximate the image. The nuclei and thus the corresponding cells can fall anywhere in the image. Unlike the classical MAT, the locations of the individual cells cannot be tuned to minimize the representation error. The error can be reduced by increasing the density of points, and for a given density, by trying different sets of statistically similar point patterns and selecting

one that gives the minimum error. The lack of freedom to control individual point locations is the price paid for achieving the security in transmission. The data compression achieved by the representation is further improved by marking for deletion those cells that are deep in the interior of a region. Such an adaptive representation achieves high spatial resolution in the border areas, where it is necessary, and coarse resolution in the interiors. The experimental results given in the paper used binary images extracted from gray level images by thresholding. However, gray level images can be processed directly by carrying out neighborhood cell homogeneity tests in the manner described in [3].

## REFERENCES

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vol. 2, Academic Press, New York, 1982.
2. H. Blum and R. Nagel, Shape description using weighted symmetric axis features, *Pattern Recognition* **10**, 1978, 167–180.
3. N. Ahuja, L. S. Davis, D. L. Milgram, and A. Rosenfeld, Piecewise approximation of pictures using maximal neighborhoods, *IEEE Trans. Comput.* **C-27**, No. 4, 1978, 375–379.
4. G. Voronoi, Nouvelles applications des parameters continus a la theorie des formes quadratiques. Duexieme Memoire: Recherches sur les parallelloedres primities, *J. Reine Angew. Math.* **134**, 1908, 198–287.
5. M. I. Shamos and D. Hoey, Closest-point problem, in *Proc. 16th Ann. Symp. Found. Comput. Sci.* October 1975, pp. 131–162.
6. D. T. Lee, *Proximity and Reachability in the Plane*, Ph.D. dissertation, Coordinated Science Laboratory Report ACT-12, University of Illinois, Urbana, Illinois, 1978.