# Quadtree Region Representation in Cartography: Experimental Results

HANAN SAMET, MEMBER, IEEE, AZRIEL ROSENFELD, FELLOW, IEEE, CLIFFORD A. SHAFFER, STUDENT MEMBER, IEEE, AND ROBERT E. WEBBER, STUDENT MEMBER, IEEE

*Abstract*—**Results of a study are summarized in which quadtrees were used to encode the regions in three map overlays representing a small area in northern California. Programs were then written to perform various analysis and manipulation tasks on the quadtree-encoded regions. Data is provided on the compactness of the encodings and the efficiency of the programs.**

Manuscript received February 15, 1983. This work was supported in part by the U.S. Army Engineer Topographic Laboratory under Contract DAAK70-81-C-0059.

The authors are with the Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742.

## I. Introduction

This correspondence summarizes the results of an experimental study of the use of quadtree region representations in cartography. Quadtree encodings were constructed for the regions in three map overlays representing a small area of Northern California. Programs were then written to perform various analysis and manipulation tasks on the quadtree-encoded regions. This correspondence provides data on the compactness of the encodings and the efficiency of the programs. Further details about the study can be found in [1].

The purpose of the study was to evaluate quadtree encoding and processing techniques for cartographic data. In addition to providing empirical data on the sizes of the quadtrees involved, the study investigated the efficiency of the following specific processing tasks: connected component analysis; computation of area, perimeter, coordinates of centroid, and enclosing upright rectangle for components; determining whether a given point lies in a given region; constructing a quadtree containing a subsection of a region (i.e., a windowing operation); and constructing the quadtree of the union or intersection of two regions.

The quadtree representation of regions, first proposed by Klinger [2], has been the subject of numerous papers over the past few years [3]–[21]. Numerous algorithms have been developed for constructing compact quadtree representations, converting between them and other region representations, computing region properties from them, and computing the quadtree representations of Boolean combinations of regions from those of the given regions. For recent overviews of this literature see [22]–[23]. In particular, for most of the specific tasks mentioned in the previous paragraph, algorithms can be found in [6], [7], [14], [15]. In the study of connected component analysis, several different algorithms were compared, including methods using explicit links ("ropes") between quadtree nodes representing adjacent image blocks; see [6], [14], [20] for further discussion.

## II. Database

The database used in the study was supplied by the U.S. Army Engineer Topographic Laboratory, Ft. Belvoir, VA. It consisted of three map overlays (Figs. 1–3) representing land use classes, terrain elevation contours, and floodplain boundaries. In the case of the elevation contours, only those at multiples of 100 feet were used; and in all cases, only the portions of the overlays bounded by the fiducial marks were used. The data were hand-digitized to $400 \times 450$ pixels, and labels were associated with the pixels in each of the resulting regions, specifying the particular land use class or elevation range. The regions were then quadtree-encoded using a $512 \times 512$ grid.

Tables I–III show quadtree statistics for the 35 land use classes and 11 elevation ranges, as well as the three regions in the floodplain map. In each table, the column "nodes created" refers to the maximum number of nodes existing at any step in the process of building the quadtree. Building was done "bottom up," by scanning the image and merging nodes when all the nodes in a $2 \times 2$ block are found to be of the same type, as described in [9]. We see that the numbers of "nodes created" ranged from about 1700 to about 14 500 (still only a small fraction of the $2^{18}$ potential pixels in the $512 \times 512$ grid), and the numbers in the final quadtrees ranged from about 130 to about 13 000. "Black nodes" are leaf nodes representing blocks of pixels belonging to the given region; "white nodes" are leaf nodes representing blocks of background pixels; and "gray nodes" are nonleaf nodes.

These quadtrees were stored on disk in files that contained lists of the node types met in a preorder traversal of the quadtree. Some of the simpler algorithms can manipulate these files di-

0018-9472/83/1100-1148$01.00 

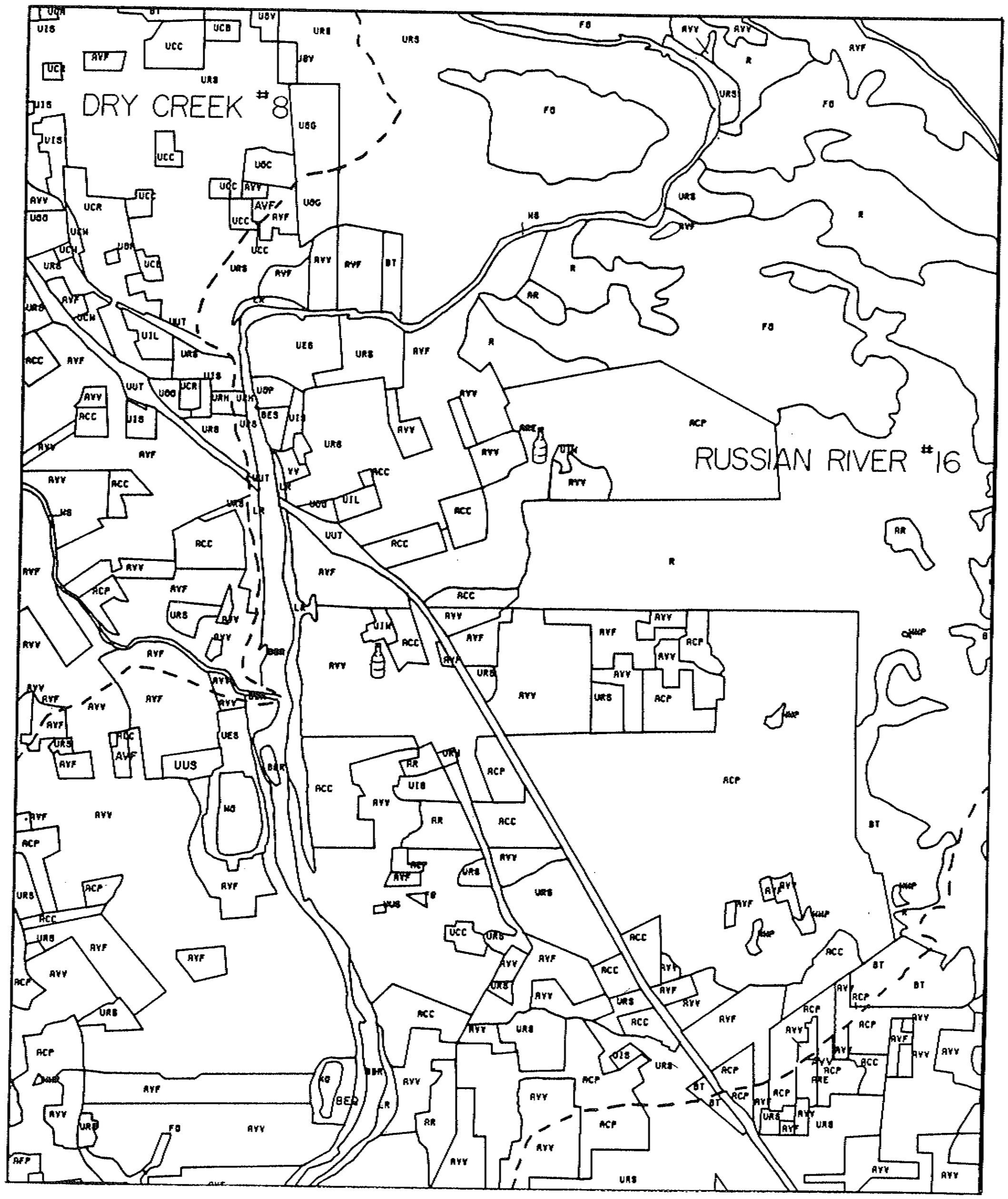


Fig. 1. Land use classes.

rectly. Others read the file into core where it is expanded to include pointers to father nodes and son nodes.

## III. Region Analysis and Manipulation

For each region type (land use class, elevation range), a connected component analysis was performed on the quadtree-encoded data, using the find-neighbor algorithm described in [14]. This algorithm was compared to an alternative algorithm that uses an augmented quadtree representation incorporating "ropes" (links between nodes of the same size that are spatially adjacent) [6]. A third algorithm, not previously described, was based on modifying the quadtree traversal algorithm to pass the neighbors of each subtree's root as parameters. This requires more time than using ropes (because of the need to pass the parameters), but requires 40 percent less memory. Tables IV–VI compare these three algorithms by giving the average number of nodes visited in order to find a neighbor. The results for find-neighbor are even better than predicted by theory [20].

Programs were also written to compute the area, perimeter, coordinates of centroid, and enclosing upright rectangle of each of the connected components [7], [15]. In addition, a program was written to determine, for a given quadtree-encoded region (not necessarily connected) and a given point (specified by its coordinates), whether or not the point lies in the region. This "point-in-polygon" task is especially efficient using quadtree encoding, since one need only move directly down the tree, as directed by the successive bits of the point's coordinates, until a leaf is reached; the point is in the region iff this leaf is "black."

Finally, two types of programs for manipulating quadtree-encoded regions were written. The first constructs the quadtree of the union or intersection of two regions from the quadtrees of the regions [6], [7]. This is done by traversing the two trees in parallel

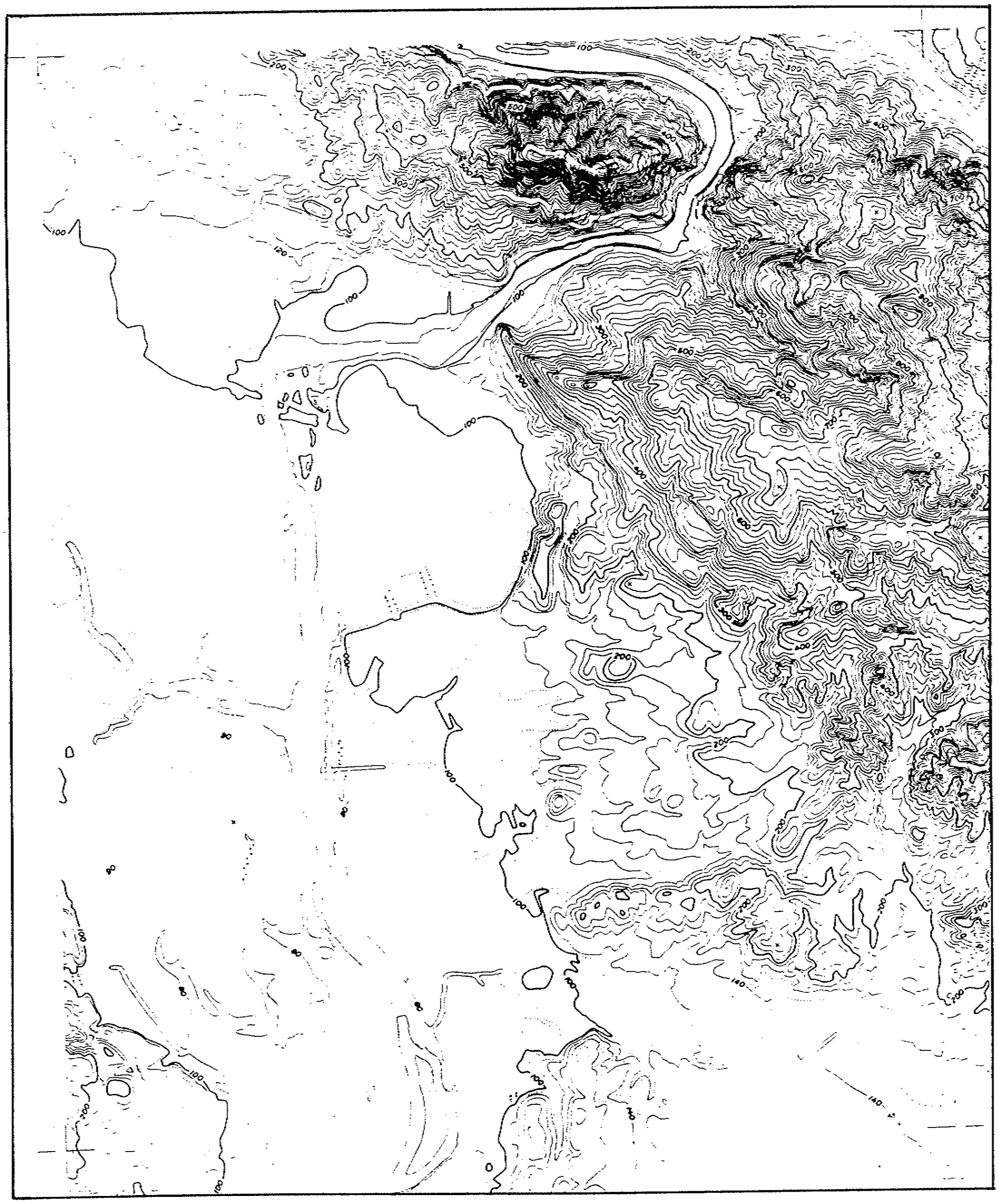

Fig. 2. Elevation contours.

and building the output tree in accordance with the following rules:

| if current nodes are | union | intersection |
|---|---|---|
| both black | return black | return black |
| both white | return white | return white |
| both gray | recurse | recurse |
| one black | return other subtree | return black |
| one white | return white | return other subtree |

The second program constructs the quadtree of the intersection between a given region (defined by a quadtree) and a given square window (of size a power of two). This is done by finding the smallest subtree of the given quadtree that contains the window. If it coincides with the window, return the subtree; if not, split the window into quadrants and process each of them analogously with respect to the current subtree. In both programs, on returning from each recursive call, it is necessary to check if the four leaves in a $2 \times 2$ block are all of the same color, and if so, to replace their father by a leaf of that color.

When a region is displayed using its quadtree representation [11], we can generate approximations to the region by truncating the tree at a given level. Fig. 4 shows the results of displaying the central region of the floodplain map using 10, 9, 8, 7, and 6 levels of its quadtree, respectively. Table VII shows the number of nodes at each level (level 1 is the root) for each of the three map overlays when their regions are represented by quadtrees. These approximations show the utility of constructing a breadth-first traversal of a quadtree for purposes of skimming a database.

In general, it should be noted that displaying quadtree files is much faster than pixel-based files on many common CRT devices

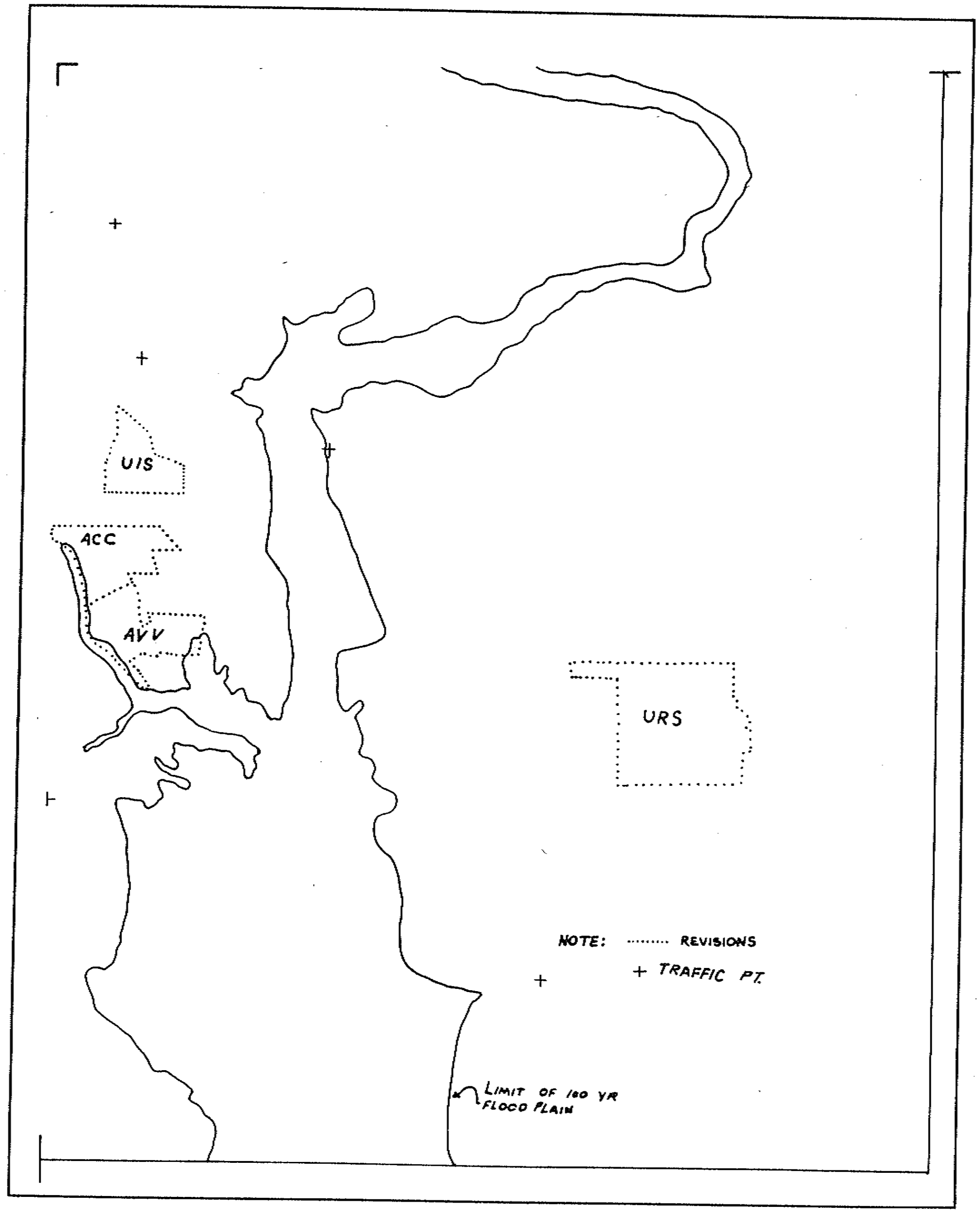


Fig. 3. Floodplain boundaries.

that allow specification of an entire block to be of one color. The leaves of the quadtree indicate a convenient partitioning of a map into monochrome blocks. Since it is not necessary to store the quadtree in core in order to display it, simple hardware could be designed that displayed quadtree files directly.

## IV. Concluding Remarks

The experiments described in this paper provide a quantitative assessment of the efficiency of quadtrees as a means of representing regions in a cartographic database. The quadtree representation is significantly more compact than the binary array representation. Efficient algorithms exist for region property computation and manipulation using the quadtree representation; the expected time required by these algorithms is proportional to the sizes of quadtrees involved as established on theoretical grounds in [9], [11], [14], [15], [20]. The point-in-polygon and set-theoretic operations on regions are especially efficient. Truncation of quadtrees can be used to generate approximations to regions that are even more compact. We have not given timing results in this correspondence (see [1]) because of their dependence on the operating system and on the available memory. The quadtree algorithms are easy to implement using structured programming languages; in our experiments, the *C* language was used. In conclusion, our experiments confirm that quadtrees constitute a viable method of region representation which is quite suitable for use in geographic information systems.

## Acknowledgment

The authors gratefully acknowledge Janet Salzman for her help in preparing this correspondence.

TABLE I
QUADTREE BUILDING STATISTICS FOR LANDUSE MAP

| Class | Nodes in Tree | Nodes Created | % Used in Tree | Gray Nodes | White Nodes | Black Nodes |
|---|---|---|---|---|---|---|
| acc | 4337 | 5925 | 73.2 | 1084 | 1847 | 1406 |
| acp | 7725 | 8981 | 86.0 | 1931 | 3048 | 2746 |
| ar | 1145 | 2697 | 42.5 | 286 | 499 | 360 |
| are | 129 | 1725 | 7.5 | 32 | 71 | 26 |
| avf | 11937 | 13341 | 89.5 | 2984 | 4776 | 4177 |
| avv | 13193 | 14445 | 91.3 | 3298 | 5359 | 4536 |
| bbr | 537 | 2109 | 25.5 | 134 | 250 | 153 |
| beq | 353 | 1873 | 18.8 | 88 | 168 | 97 |
| bes | 193 | 1825 | 10.6 | 48 | 94 | 51 |
| bt | 2293 | 3841 | 59.7 | 573 | 951 | 769 |
| fo | 5485 | 7109 | 77.2 | 1371 | 2121 | 1993 |
| lr | 1481 | 3045 | 48.6 | 370 | 670 | 441 |
| r | 7001 | 8609 | 81.3 | 1750 | 2792 | 2459 |
| ucb | 249 | 1881 | 13.2 | 62 | 118 | 69 |
| ucc | 817 | 2433 | 33.6 | 204 | 381 | 232 |
| ucr | 1069 | 2701 | 39.6 | 267 | 457 | 345 |
| ucw | 449 | 2081 | 21.6 | 112 | 197 | 140 |
| ues | 1113 | 2737 | 40.7 | 278 | 506 | 329 |
| uil | 345 | 1977 | 17.5 | 86 | 158 | 101 |
| uis | 1037 | 2649 | 39.1 | 259 | 453 | 325 |
| uiw | 293 | 1917 | 15.3 | 73 | 139 | 81 |
| unk | 1121 | 2681 | 41.8 | 280 | 540 | 301 |
| uoc | 173 | 1805 | 9.6 | 43 | 79 | 51 |
| uog | 377 | 2009 | 18.8 | 94 | 149 | 134 |
| uoo | 429 | 2061 | 20.8 | 107 | 201 | 121 |
| uop | 269 | 1901 | 14.2 | 67 | 186 | 66 |
| uov | 229 | 1861 | 12.3 | 57 | 99 | 73 |
| urn | 237 | 1861 | 12.7 | 59 | 125 | 53 |
| urs | 9921 | 11313 | 87.7 | 2480 | 3993 | 3448 |
| uus | 297 | 1921 | 15.5 | 74 | 142 | 81 |
| uut | 3069 | 4621 | 66.4 | 767 | 1379 | 923 |
| vv | 153 | 1785 | 8.6 | 38 | 76 | 39 |
| wo | 485 | 2029 | 23.9 | 121 | 225 | 139 |
| ws | 4677 | 6245 | 74.9 | 1169 | 2025 | 1483 |
| wwp | 457 | 2049 | 22.3 | 114 | 101 | 242 |

TABLE II
QUADTREE BUILDING STATISTICS FOR TOPOGRAPHY MAP

| Elevation | Nodes in Tree | Nodes Created | % Used in Tree | Gray Nodes | White Nodes | Black Nodes |
|---|---|---|---|---|---|---|
| 0– 100 | 6809 | 8161 | 83.4 | 1702 | 2577 | 2530 |
| 100– 200 | 13853 | 14913 | 92.9 | 3463 | 5295 | 5095 |
| 200– 300 | 11813 | 13381 | 88.3 | 2953 | 4713 | 4147 |
| 300– 400 | 8845 | 10469 | 84.5 | 2211 | 3596 | 3038 |
| 400– 500 | 7121 | 8745 | 81.4 | 1780 | 2917 | 2424 |
| 500– 600 | 6005 | 7629 | 78.7 | 1501 | 2534 | 1970 |
| 600– 700 | 5341 | 6973 | 76.6 | 1335 | 2140 | 1866 |
| 700– 800 | 4725 | 6357 | 74.3 | 1181 | 1955 | 1589 |
| 800– 900 | 3121 | 4753 | 65.7 | 780 | 1292 | 1049 |
| 900–1000 | 1277 | 2909 | 43.9 | 319 | 516 | 442 |
| 1000–1100 | 161 | 1793 | 9.0 | 40 | 88 | 33 |

TABLE III
QUADTREE BUILDING STATISTICS FOR FLOODPLAIN MAP

| Area | Nodes in Tree | Nodes Created | % Used in Tree | Gray Nodes | White Nodes | Black Nodes |
|---|---|---|---|---|---|---|
| left bank | 4021 | 5473 | 73.5 | 1005 | 1491 | 1525 |
| floodplain | 6257 | 7645 | 81.8 | 1564 | 2485 | 2208 |
| right bank | 2885 | 4009 | 72.0 | 721 | 1133 | 1031 |

TABLE IV
LANDUSE CONNECTED COMPONENT RESULTS

| Class | Number of Neighbors Sought | Findnbr Avg Cost | Ropes Avg Cost | Args Avg Cost |
|---|---|---|---|---|
| acc | 2812 | 3.55 | 1.40 | 3.08 |
| acp | 5492 | 3.58 | 1.40 | 2.81 |
| ar | 720 | 3.48 | 1.33 | 3.18 |
| are | 52 | 5.63 | 1.98 | 4.96 |
| avf | 8354 | 3.53 | 1.40 | 2.86 |
| avv | 9072 | 3.55 | 1.39 | 2.91 |
| bbr | 306 | 3.59 | 1.35 | 3.51 |
| beq | 194 | 3.82 | 1.31 | 3.64 |
| bes | 102 | 3.30 | 1.38 | 2.80 |
| bt | 1538 | 3.53 | 1.35 | 2.98 |
| fo | 3986 | 3.54 | 1.45 | 2.75 |
| lr | 882 | 3.71 | 1.25 | 3.36 |
| r | 4918 | 3.63 | 1.46 | 2.85 |
| ucb | 138 | 3.31 | 1.20 | 3.61 |
| ucc | 464 | 3.64 | 1.37 | 3.52 |
| ucr | 690 | 3.60 | 1.42 | 3.10 |
| ucw | 280 | 3.58 | 1.36 | 3.21 |
| ues | 658 | 3.81 | 1.38 | 3.38 |
| uil | 202 | 3.75 | 1.55 | 3.42 |
| uis | 650 | 3.53 | 1.39 | 3.19 |
| uiw | 162 | 3.84 | 1.35 | 3.62 |
| unk | 602 | 3.45 | 1.35 | 3.72 |
| uoc | 102 | 3.59 | 1.49 | 3.39 |
| uog | 268 | 3.66 | 1.40 | 2.81 |
| uoo | 242 | 3.22 | 1.35 | 3.55 |
| uop | 132 | 3.64 | 1.42 | 4.08 |
| uov | 146 | 3.42 | 1.29 | 3.14 |
| urn | 106 | 3.89 | 1.28 | 4.47 |
| urs | 6896 | 3.54 | 1.40 | 2.88 |
| uus | 162 | 3.55 | 1.35 | 3.67 |
| uut | 1846 | 3.62 | 1.38 | 3.33 |
| vv | 78 | 3.33 | 1.28 | 3.92 |
| wo | 278 | 3.97 | 1.38 | 3.49 |
| ws | 2966 | 3.70 | 1.28 | 3.15 |
| wwp | 202 | 3.62 | 1.38 | 4.52 |

TABLE V
TOPOGRAPHY CONNECTED COMPONENT RESULTS

| Elevation | Number of Neighbors Sought | Findnbr Avg Cost | Ropes Avg Cost | Args Avg Cost |
|---|---|---|---|---|
| 0– 100 | 5060 | 3.48 | 1.41 | 2.69 |
| 100– 200 | 10190 | 3.51 | 1.41 | 2.72 |
| 200– 300 | 8294 | 3.53 | 1.41 | 2.85 |
| 300– 400 | 6076 | 3.57 | 1.36 | 2.91 |
| 400– 500 | 4848 | 3.62 | 1.36 | 2.94 |
| 500– 600 | 3940 | 3.64 | 1.36 | 3.05 |
| 600– 700 | 3732 | 3.62 | 1.36 | 2.86 |
| 700– 800 | 3178 | 3.69 | 1.38 | 2.97 |
| 800– 900 | 2098 | 3.57 | 1.37 | 2.98 |
| 900–1000 | 884 | 3.54 | 1.41 | 2.89 |
| 1000–1100 | 66 | 3.56 | 1.41 | 4.88 |

TABLE VI
FLOODPLAIN CONNECTED COMPONENT RESULTS

| Region | Number of Neighbors Sought | Findnbr Avg Cost | Ropes Avg Cost | Args Avg Cost |
|---|---|---|---|---|
| left bank | 3050 | 3.25 | 1.35 | 2.64 |
| floodplain | 4416 | 3.50 | 1.46 | 2.83 |
| right bank | 2062 | 3.62 | 1.66 | 2.80 |

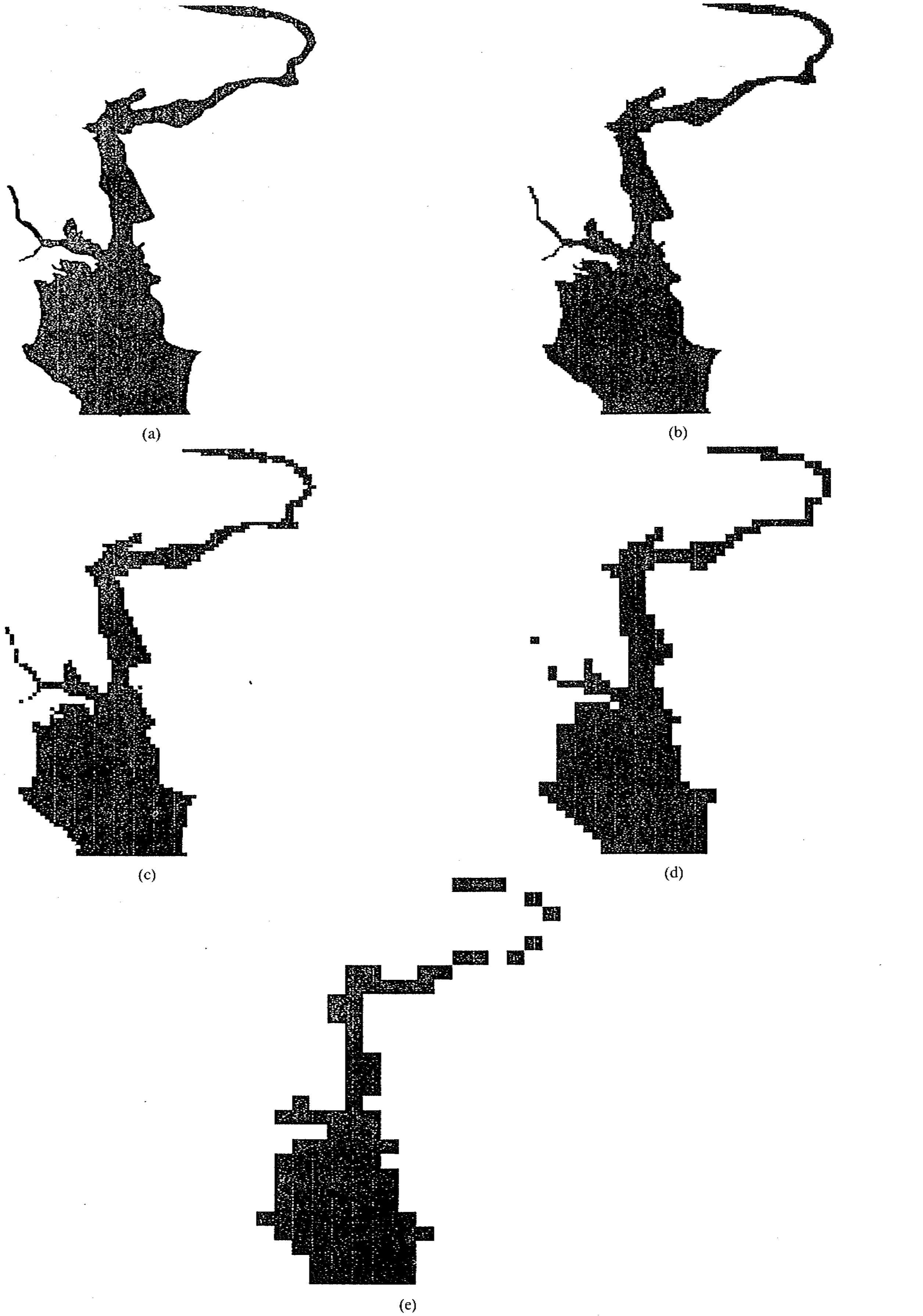


Fig. 4. (a) Result of executing QDISPLAY on flood center of the floodplain map using 10 levels. (b) Result of executing QDISPLAY on flood center of the floodplain map using 9 levels. (c) Result of executing QDISPLAY on flood center of the floodplain map using 8 levels. (d) Result of executing QDISPLAY on flood center of the floodplain map using 7 levels. (e) Result of executing QDISPLAY on flood center of the floodplain map using 6 levels.

TABLE VII
QUADTREE TRUNCATION STATISTICS FOR EACH MAP

| Depth | Land use | Map | Topography | Map | Floodplain | Map |
|---|---|---|---|---|---|---|
| of Tree | # of Nodes | % Reduced | # of Nodes | % Reduced | # of Nodes | % Reduced |
| 10 | 38233 | 00.00 | 33349 | 00.00 | 6941 | 00.00 |
| 9 | 22089 | 44.22 | 18517 | 44.47 | 4473 | 35.55 |
| 8 | 9489 | 75.18 | 7473 | 77.59 | 2297 | 66.91 |
| 7 | 3341 | 91.26 | 2537 | 92.39 | 1093 | 84.26 |
| 6 | 1057 | 97.23 | 833 | 97.50 | 529 | 92.38 |
| 5 | 309 | 99.19 | 296 | 99.19 | 213 | 96.94 |
| 4 | 85 | 99.78 | 77 | 99.77 | 77 | 98.89 |
| 3 | 21 | 99.95 | 21 | 99.94 | 21 | 99.70 |
| 2 | 5 | 99.99 | 5 | 99.99 | 5 | 99.93 |
| 1 | 1 | 99.99 | 1 | 99.99 | 1 | 99.99 |

## REFERENCES


[1] A. Rosenfeld, H. Samet, C. Shaffer, and R. E. Webber, "Application of hierarchical data structures to geographical information systems," Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Rep. TR-1197, June 1982.

[2] A. Klinger, "Data structures and pattern recognition," in *Proc. 1st Int. Joint Conf. Pattern Recognition*, 1973, pp. 497–498.

[3] A. Klinger and C. R. Dyer, "Experiments in picture representation using regular decomposition," *Comput. Graphics Image Proc.*, vol. 5, pp. 68–105, 1976.

[4] A. Klinger and M. L. Rhodes, "Organization and access of image data by areas," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 50–60, 1979.

[5] N. Alexandridis and A. Klinger, "Picture decomposition, tree data-structures, and identifying directional symmetries as node combinations," *Comput. Graphics Image Proc.*, vol. 8, pp. 43–77, 1978.

[6] G. M. Hunter and K. Steiglitz, "Operations on images using quadtrees," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 145–153, 1979.

[7] M. Shneier, "Calculations of geometric properties using quad-trees," *Comput. Graphics Image Proc.*, vol. 16, pp. 296–302, 1981.

[8] G. M. Hunter and K. Steiglitz, "Linear transformation of pictures represented by quadtrees," *Comput. Graphics Image Proc.*, vol. 10, pp. 289–296, 1979.

[9] H. Samet, "Region representation: Quadtrees from binary arrays, *Comput. Graphics Image Proc.*, vol. 13, pp. 88–93, 1980.

[10] ——, "An algorithm for converting rasters to quadtrees," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 93–95, 1981.

[11] ——, "Algorithms for the conversion of quadtrees to rasters," Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Rep. TR-979, Nov. 1980.

[12] ——, "Region representation: Quadtrees from boundary codes," *Commun. Ass. Comput. Mach.*, vol. 23, pp. 163–170, 1980.

[13] C. R. Dyer, A. Rosenfeld, and H. Samet, "Region representation: Boundary codes from quadtrees," *Commun. Ass. Comput. Mach.*, vol. 23, pp. 171–179, 1980.

[14] H. Samet, "Connected component labeling using quadtrees," *J. Ass. Comput. Mach.*, vol. 28, pp. 487–501, 1981.

[15] ——, "Computing perimeters of images represented by quadtrees," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 683–687, 1981.

[16] C. R. Dyer, "Computing the Euler number of an image from its quadtree," *Comput. Graphics Image Proc.*, vol. 13, pp. 270–276, 1980.

[17] H. Samet, "Distance transform for images represented by quadtrees," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 298–303, 1982.

[18] M. Shneier, "Path-length distances for quadtrees," *Inform. Sci.*, vol. 23, pp. 49–67, 1981.

[19] S. Ranade, A. Rosenfeld, and H. Samet, "Shape approximation using quadtrees," *Pattern Recogn.*, vol. 15, pp. 31–40, 1982.

[20] H. Samet, "Neighbor finding techniques for images represented by quadtrees," *Comput. Graphics Image Proc.*, vol. 18, pp. 37–57, 1982.

[21] ——, "A quadtree medial axis transform," *Commun. Ass. Comp. Mach.*, vol. 26, pp. 680–693, 1983.

[22] A. Rosenfeld, "Quadtrees and pyramids: hierarchical representation of images," Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Rep. TR-1171, May 1982.

[23] H. Samet, "Hierarchical data structures for representing geographical information," Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Rep. TR-1208, Aug. 1982.