Piecewise Circular Description of Image Curves Using Constancy of Grey-level Curvature

John Dolan
George Reynolds
Les Kitchen

COINS Technical Report 86-33

July 1986

This work has been supported by the Defense Mapping Agency under contract 800-85-C-0012.

Piecewise Circular Description of Image Curves Using Constancy of Grey-level Curvature

John Dolan

George Reynolds

Les Kitchen

ABSTRACT

A method to derive local, piecewise-circular descriptors of image curves is presented. Edge extraction is used to localize processing; and edge contours are partitioned by constancy of quantized local grey-level curvature, rather than by extrema of curvature of the contours themselves. The use of a special segmentation technique, "overlapping partitions," overcomes many problems associated with quantization. Circular arcs are fitted independently to each of the extracted curved-edge segments. The parameters of these arcs, together with other measured attributes, provide a rich description of each curved-edge segment. The method is intended to provide reliable, local descriptive tokens for use by later grouping and interpretation processes.

1. INTRODUCTION

The task of early computer vision frequently involves the extraction and organization of image features into primitives better suited than the raw image to the requirements of subsequent symbolic processes. In this regard, much attention has been given to the problem of deriving descriptors for image curves [1,2,3].

The term *image curves*, is usually taken to mean linear/curvilinear events in the image plane, frequently associated with boundaries of objects and made manifest by some appropriate extraction process, such as edge detection. In this paper, image curves are in fact associated with pixels whose gradient magnitude (with respect to the intensity data) is locally maximum. Extraction normally results in the identification of sets of significant pixels, corresponding to the locations of such events. Description is a process of summarizing the planar distributions of the pixels comprising these sets. With respect to image curves, description entails two subproblems [4]: partitioning the underlying pixel sets into integral units for description, and fitting an appropriate curve to each such unit.

Description strategies may be broadly classed into two categories, according to their method of partitioning the underlying pixel sets. The more usual approach [1,2,3] involves chain-coding of pixels into contours and locating the extrema of curvature of these contours. These extrema determine breakpoints of the partition; and typically, description is a set of splines whose knots are these breakpoints. Such a method is largely viewpoint invariant [2], and normally results in continuous description of contours. However, it suffers at least one serious drawback: the location of breakpoints is based primarily on measurements at vertices—i.e., points of locally maximum or minimum curvature. But it is precisely these points for which the extraction process itself is least well defined.

Further, if one is concerned with interpretation of natural scenes (a problem with which any truly general vision system must contend), then one is faced with a profusion of objects in highly complex, often occluding, spatial arrangements. In this context, the essential problem cannot be reduced simply to a matter of tracking and describing the bounding contour of an object as would suffice with single-object laboratory images. Rather one must seek to develop systems that can deal with the inevitable fragmentation of contours with which they will be faced. That is, the requirement is for systems that can adequately describe the local *geometric character* of such fragments; and ideally, ones that can subsequently reintegrate those fragments which belong together into coherent wholes. The concern of the present work is this local description task.

A second, more recent approach to partitioning the underlying pixel sets of edge regions is the method of *overlapping sectors*, which was developed by Burns, Hanson, and Riseman [5] in the context of straight lines. The system described herein exploits a variant of this overlapping sectors approach but with respect to second order curves. Using this method, partitioning is determined by uniformity, rather than maxima/minima, of curvature; and in this sense it constitutes the dual of extrema methods. Moreover, the location of breakpoints is largely determined by measurements away from vertices—i.e.,in areas where the extraction process can be said to be well defined.

2. OVERLAPPING PARTITIONS SEGMENTATION

This novel approach to the problem of extracting straight lines was recently reported in Burns, et al. [5]. It involves simple local computation (not involving any histogram methods) and the computation of connected components. In this section we review the algorithm, and describe a simple generalization which makes it applicable to extracting curved lines. In the next section we will describe the details of the algorithm for deriving circular arc descriptors.

The central module of our algorithm we term the "Overlapping Partitions Algorithm," and in the context of extracting straight lines it may be summarized as follows. Processing begins by applying a gradient orientation measure to the image. Any number of methods are applicable; however, normally the row and column derivatives are computed and orientation is taken to be the direction of this gradient vector. The space of possible orientations is partitioned into a set of 8 non-overlapping sectors, each of 45 degrees arc. A second such partition is formed but rotated 22.5 degrees from the first. Thus, the sectors of one partition overlap the sectors of the other—hence the name *overlapping sectors.* For each partition, the image is labeled according to the sector into which the gradient orientation falls. Thus each pixel gets two labels. A connected components algorithm is then applied to the labeling derived from each partition. Finally, a selection procedure is employed to determine locally which partition is preferred (the components of this partition are termed *edge support regions*). Straight line descriptors are now obtained by fitting a straight line to each of the *edge support* regions.

By replacing gradient orientation with a curvature measure and by substituting an appropriate quantization of the measured curvature, the method can be adapted to deriving piecewise circular descriptors. Specifically we find ***curve support regions*** which are uniform with respect to quantized curvature and as such can be abstracted from the image data as parts of circles.

The curvature measure we employ is slight modification of the Kitchen-Rosenfeld [6] corner detector—i.e., the scaling by the gradient magnitude is dropped. However, this measure only makes sense when applied to areas of locally maximum gradient magnitude. Thus, in a manner similar to that suggested by Nagel [7], we use zero crossings of a second derivative operator to restrict the curvature operator to such areas. In this case, it provides an approximation to the local curvature of the underlying edge event; and this is precisely what we wish to describe. Thus, the general algorithm as it applies to image curves is as follows:

The processing begins by applying an operator to the image which extracts "significant edge" pixels. In our current implementation, we have been experimenting with zero crossings of both directional and non-directional second derivative operators. Both approximate areas of locally maximum gradient magnitude. In these areas of "significant" edge activity, apply the curvature operator described above. The range of this curvature operator is partitioned into two non-overlapping groups of intervals, with one partition shifted one half an interval length from the other. For each partitioning, the image is labeled according to the sector into which the curvature value falls. Thus, each pixel gets two labels. Finally, a connected components algorithm is applied to the labelings derived from each of the partitionings, and a selection procedure is applied to determine locally which partition is preferred (the components of this partition we term ***curve support regions***). Curved line descriptors are now obtained by fitting a circular arc to each of the curve support regions.

Associated with each curve is a set of curve attributes which includes: center-location, radius, arc-limits, strength, average-curvature, aggregate-error, mean-fit-error, and rho-kappa-correspondence. These attributes will be described in greater detail in a later section. Descriptors are intended to provide input to subsequent, higher level grouping operations. Thus, filtering may be performed with respect to any of these attributes.

3. GENERATING CIRCULAR DESCRIPTORS

As currently implemented, the system for deriving descriptors of image curves consists essentially of five components. The first entails the detection and localization of intensity edges in the image. The resultant set of edge pixels is used to restrict subsequent processing to edge neighborhoods. The second component analyzes these edge neighborhoods, specifically in terms of curvature in the underlying intensity data. The result is a segmentation of the edge regions based upon a quantization of this curvature. The third component entails a pre-filtering of these edge regions based upon their size and their support, resulting in so-called *curve support regions*. The fourth component derives a description of these support regions in terms of best-fit circular arcs. The final component performs post-filtering of the arc descriptors. In this section we describe each of these components in more detail.

3.1 Edge Detection

The edge detection component is decoupled from the remainder of the system. Thus, any edge operator may be used; it need only deliver a logical edge image (a bitmap) marking the locations of edge pixels. Most examples contained herein were produced using a version of the edge operator suggested by Haralick [8], which first applies the operator

$$\nabla I \cdot \nabla(||\nabla I||^2)$$

to the image. In order to reduce spurious effects of high frequency noise, the image is normally pre-smoothed with a Gaussian; thus, I may be considered a smoothed version of the image.

Note that I is a two-dimensional function in x and y; so too is $||\nabla I||^2$. As such, one may visualize their respective graphs, as surfaces in 3D space above the image plane itself. Considered from this point of view, the Haralick operator amounts to computing the directional derivative of the gradient magnitude surface in the direction of the gradient—i.e., across the idealized edge. A logical edge image is then derived by locating the zero-crossings of this "second derivative" surface.

3.2 Curve Segmentation

Once edges have been located, curvature is calculated at each edge pixel directly from the image itself. The idea is to obtain a set of segmentations of the edge contour based upon local curvatures of its projection in the image plane. The operator devised by Kitchen and Rosenfeld [6] provides a means of determining just such curvature; moreover, it does so locally and directly in two dimensions using convolution. This is in contrast to other systems dealing with curvature [1,2,3], which usually require reduction of the contour to a one-dimensional function, by chain-coding or some similar scheme. Moreover, it is our contention that these alternative methods, while attempting to describe the entire contour of an object, do not significantly abstract or summarize the curvilinear image event in a form which is suitable for subsequent grouping operations.

3.2.1 Curvature Operator

Kitchen and Rosenfeld [6] describe their operator in terms of the turning rate of the gradient, at each pixel, projected along the idealized edge. The form we use is defined by:

$$\kappa = \frac{I_{xx}I_x^2 + I_{yy}I_y^2 - 2I_{xy}I_xI_y}{(I_x^2 + I_y^2)^{3/2}}$$

where I is the smoothed image intensity data and I_x, I_y, I_{xx}, I_{xy} and I_{yy} are the partial derivatives in the row and column directions. In fact, the curvature measured by this operator is identical to the curvature of the iso-intensity contour line passing through the pixel. See Figure 1. By restating this contour line as some function of x, the equation of the curvature operator becomes the familiar equation for the curvature of a plane curve:

$$\kappa = \frac{\left(d^2y/dx^2\right)}{\left(1 + (dy/dx)^2\right)^{3/2}}$$

In other words, where the set of pixels belongs to an edge (as is the case here), the operator measures the curvature of the respective iso-intensity contour at each edge pixel. This amounts to a pixel-wise sampling of the curvature (in x and y) of the edge ramp itself. Where the zero-crossing contour follows a single iso-intensity contour, their respective curvatures will be identical. Where the edge contour wanders up or down the gradient, its curvature will be pointwise equivalent to curvatures of a succession of coincident iso-intensity contours. This has the effect of stabilizing the curvature measure, so that it better conforms to the behavior of the edge event itself. See Figure 2.

3.2.2 Quantization

Quantization is a mapping from ranges of input values to discrete representational values. In the present case, the mapping is from local grey-level curvatures measured at the edge contours to corresponding curvature labels—the particular label, in each case, depending clearly on the rules of the mapping. With respect to curvature, the optimal form of such a quantization rule is still very much an open question. In our experiments, we have generally employed a uniform linear quantization. This entailed multipying the measured curvature by a scaling factor and using evenly spaced buckets centered at the integers (Figure 3); but one can easily imagine other rules. For example, we are currently looking at quantizing by the log of the curvature, in order to better ensure scale-invariance.

3.2.3 Overlapping Partitions Segmentation

Regardless of the specific rule, quantization effects a partitioning of the input data, thereby introducing unintended artifacts at the partition boundaries. Where the measured curvature of adjacent points along the edge varies sufficiently to cross such a partition boundary, a break-point results. This sort of fragmentation has little to do with the inherent structure of the underlying contour itself and is therefore undesirable.

Performing a second quantization, with data shifted one half interval, induces a new and distinct set of breakpoints (Figure 3). Running a connected-components algorithm on the results of both quantizations produces two overlapping segmentations of the edge contours of the image. To overcome the cumulative fragmentation effects, what is needed is some strategy for integrating information across the two segmentations.

3.2.4 Voting Scheme

Adopting here the terminology and method of Burns et al.[5], the two segmentations are each comprised of *support regions* – collections of contiguous pixels with the same region label. The constituent pixels share an underlying, unifying property: for Burns it was gradient orientation; in the present case, it is quantized curvature. More to the point, each pixel belongs to two distinct support regions, one for each segmentation. It is thus possible, as Burns did, to devise a scheme whereby each pixel votes to belong to, that is to *support*, one region or the other.

The decision criteria applied will vary according to the property upon which the segmentations are constructed. The current implementation utilizes a crude approximation to arc-length, *size*, as determined by the cardinality of the support regions (since the regions are themselves comprised solely of pixels which follow the edge contour). A pixel thus votes to support that region which has longer arc-length by voting for the one with the greater number of pixels. See Figure 4. Votes are tallied for each region and its respective *support* is determined as the ratio of votes received to *size*. Regions are then usually pre-filtered for simple majority support of 0.5.

Since circle fitting is costly and requires a minimum number of points, it was thought that such a decision criterion would provide reasonably stable regions to which to fit the circular arcs. However, this tends to filter out possibly significant small regions which overlap larger regions. An alternative and possibly more useful choice might be to first fit arcs to all regions of at least some minimum size, and then to decide support based upon the mean fit error. In fact, this approach is currently under investigation. (With the current implementation, similar results are obtainable by setting the support-filter at 0.0 and post-filtering on the mean fit error.)

3.3 Region Filtering

Edge regions possess two important attributes which form the basis for pre-filtering—i.e., filtering done prior to the actual fit process. These are *size* and *support*. Filtering on simple majority support (*support* $\geq$ 0.5) serves to select those regions preferred according to the decision criterion of the voting scheme. This ensures that computational effort is expended on deriving only the preferred descriptor for each edge segment. In a similar manner, filtering on size ensures that only those regions with a sufficient number of data points (usually 4-5) to adequately support the fitting process are actually selected. The regions passed by these filters are termed ***curve support regions***. See Figure 5.

3.4 Circular Arc Fit

Associated with each support-region is an implicit curve, which best fits the constituent data points of the region; and which, since regions are based upon constancy of curvature, we take to be circular. The fit algorithm itself is adapted from Agin [8]. In his paper, Agin outlines a generalized eigenvalue method for fitting ellipses to data points. The parameters of the curve are given by:

$$f(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

It is a simple matter to tailor this algorithm to the case of the circle, as a degenerate form of ellipse. The rotation term disappears and the distinct parameters of the squared terms are constrained to be identical, yielding:

$$f(x,y) = A(x^2 + y^2) + Bx + Cy + D = 0$$

Agin points out that $f(x,y)$ may provide its own error function, in that point-to-curve distances for points along the curve are zero. As distance from the curve increases so does the magnitude of this error function. For a point within a small neighborhood of the curve, the function is proportional to its perpendicular distance to the curve; and the constant of proportionality is reciprocal to the magnitude of the gradient of the function. Moreover, it is possible to choose constraints on the coefficients of the function such that the average gradient is unity. The resulting error function is then directly related to the perpendicular distance form point to curve. Thus, pointwise error for each (x_i, y_i) is given by:

$$\xi_i = A(x_i^2 + y_i^2) + Bx_i + Cy_i + D$$

while the aggregate error is:

$$\Xi = \sum \xi_i^2 = \sum [A(x_i^2 + y_i^2) + Bx_i + Cy_i + D]^2$$

Adopting the generalized eigenvalue solution method described by Agin results in circles, each of whose sum of squared perpendicular distances to its respective data points is essentially minimal. Subsequent ordering of each set of data points by radial angle defines the arc-limits of its respective circular arc descriptor.

Note, the resultant circular arc descriptors are completely independent of each other. Each is treated as a satisfactory description of its local underlying support region alone.

In contrast to other approaches [1,2,3], no attempt is made to align (or smoothly join) a descriptor with its neighbors across breakpoints. This is both for simplicity of computation and because we view the current system more as the front-end to a comprehensive grouping system than as a stand-alone system.

3.5 Descriptor Filtering

As mentioned above, each descriptor has computed for it a set of attributes which includes: center-location, radius, arc-limits, strength, average-curvature, aggregate-error, mean-fit-error, and rho-kappa-correspondence. The definitions of these attributes along with plausible filterings are given as follows:

1. *center-location* (x_c, y_c)—center of fitted circular arc, may be used to select possibly concentric descriptors;

2. *radius* (ρ)—radius of fitted circular arc, provides selection of arcs of similar curvature;

3. *arc-limits* (θ_1, θ_2)—counterclockwise ordering of the beginning and end respectively of the fitted arc descriptor in terms of radial angle, may be used to select arcs based upon that portion of the complete circle they represent, or in concert with ***radius*** to select based upon arc-length;

4. *strength* $(\sum \|\nabla I\|/n)$—the average gradient magnitude of the underlying support region, may be used to filter out descriptors of weaker edges;

5. *average-curvature* $(\sum \kappa/n)$—average measured curvature in the underlying support region, may be used to select descriptors whose measured region curvature is centered about some value;

6. *aggregate-error* $(\sum(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - \rho)^2)$—the unnormalized *variance* of the data from the descriptor fit (this is equivalent to the function Ξ that was minimized),

reflects the overall unsigned goodness of fit, and thus may be used to filter descriptors accordingly;

7. *mean-fit-error* (Ξ/n)—the sum of squared errors normalized to the size of the region, another filter for goodness of fit;

8. *rho-kappa-correspondence* ($\sum(1-\rho\kappa)/n$)—the correspondence of the curvature of the descriptor fit to the average measured curvature in the underlying region, a signed goodness of fit, which goes to 0 as the distribution of measured curvatures centers on $1/\rho$, an indication of the reliability of the curvature measure over the particular region, and may be used to filter accordingly.

We have noted in a previous section that filtering on size is normally done prior to fitting to ensure a minimum number of data points. Pre-filtering on support is also done, usually in order that computational effort be expended only on the preferred descriptor of each edge segment. Post-filtering, on the other hand, is available for all descriptor attributes and in any combination. In this way it is possible to select only descriptors satisfying a specified set of criteria. Since we view our system as a possible front-end to a symbolic/token based grouping system, such descriptor filtering should prove useful.

4. EXAMPLES

We now present examples to illustrate the performance of the system in three distinct image domains: low-relief, high contrast imagery (Example 1); aerial photography (Example 2); and natural scenes (Example 3).

4.1 Clock

The original 64×64 intensity image, shown in Figure 6(a), exhibits a number of distinct smoothly curving features associated with the numerals and the edge of the clock. It also contains some strictly linear features such as the flanges on which the numerals lie, as well as the ends of the numerals themselves. It is thus provides a demonstration of system performance with respect to both feature types. In addition, we will use our discussion of this image to trace the progression of processing steps that the system employs.

The first step is edge extraction; Figure 6(b) is the result. What is shown is the outline of the edge regions, those pixels through which the zero-crossing contour passes. The edges of the numeral three and its flange are quite good. The edges of the other flange and the clock itself seem acceptable. The regions in the lower left area of the image, correspond to edges of a clear circular face cover. Although it is difficult to see from the photograph, their general shape and location are also acceptable. There are problems however with the numeral two. The first of these has to do with the confounding of the upper part of the numeral with the top edge of the image. This is simply a function of its proximity to the image edge, and would disappear were the numeral positioned away from the edge. The second problem occurs on the diagonal stroke, near the base of the two. Here we see that the edge operator has smeared the right and left diagonal edges into one another. This results as a combination of the size of the edge masks (we used 3 × 3 Prewitt masks), and undersampling of the image (the diagonal stroke is barely four pixels wide at this point). Obviously, this second problem could avoided by using a smaller edge mask and/or higher sampling rate.

Figures 6(c) and 6(d) show two partitions of the edge regions based upon quantized curvature, measured in the intensity image by the Kitchen/Rosenfeld operator. They represent the unshifted and shifted versions, respectively, of a uniform quantization. Note the differences between the two figures, particularly at the corners of the flanges. As stated, the motivation behind *overlapping partitions* is to alleviate breakpoint artifacts induced by the partitioning process itself. We see that while the partition associated with 6(c) has placed breakpoints at what we might consider the intuitively correct positions—i.e., at the corners, the partition of 6(d) has also correctly performed its task, smoothing over these breakpoints in its version. In Figures 6(e) and 6(f), we see these same partitions, with the edge gradients overlaid. This gives a rough notion of the correspondence of the partition boundaries to the grey-level curvature, in so much as one may consider such curvature equivalent to the rate of change of the gradient direction.

After the determination of support, the partitions form the basis for the description process. Figures 6(g) and 6(h) show these same partitions again respectively, but with their corresponding arc descriptors fit to the *curve support regions*. In Figure 6(g), note that the descriptors for the flanges, for the inner curve at the bottom right of the numeral three, and for the edge of the clock all correspond quite well to the geometry of the underlying support regions, as well as to our intuitive notions about the shapes of the objects themselves. The same can be said for all descriptors associated with the numeral three in Figure 6(h). By contrast, the descriptor for the outer curve of the three in Figure 6(g) may in fact conform adequately to the geometry of the underlying region, given that our descriptor vocabulary is limited to circular arcs; but, in terms of our ideas regarding the shape of a three, the geometry of that region forces a description that can only be considered overly generalized. On the other hand, the problems with description of numeral two are associated with the edge extraction process and are discussed above in that connection.

Finally, in Figure 6(i) we see all arcs overlaid on the original edge regions, while Figure 6(j) shows these same arcs on the edge gradients.

4.2 Aerial Image

Figure 7(a) is a 127×127 intensity image that was reduced by averaging from a 256×256 section of a b/w aerial photograph. Figure 7(b) is the the outline of the corresponding edge image—i.e., the edge regions extracted. Figure 7(c) shows these same edge regions with all arc descriptors, having majority support and *curve support regions* with size ≥ 5 pixels, overlaid. Results are definitely mixed and difficult to summarize, except to say that the descriptive process fares only as well as the edge extraction process. Where the image structure is sufficiently well defined and at a scale which persists at the sampling rate, then edge regions are well placed and the descriptors tend to correspond well to the shapes of the actual underlying structures. This is the case in the upper regions of the image with both the triangular and circular approach ramps, and especially the upper boundaries of the road which runs vertically through the image. On the other hand, an area like the lower boundary of the horizontal road is confounded with the nearby median, because of their respective alignments with the sampling grid. This causes their edge regions to fuse at points; because of this the corresponding descriptor can only represent a compromise between the two structures.

Figures 7(d) and 7(e) represent the unshifted and shifted curvature partitions respectively, with their corresponding arc descriptors overlaid. Figure 7(f) is the result of prefiltering for regions whose size $\geq$ 15 pixels, and we see that it alleviates many of the problems. This makes sense when we consider that for a structure to persist over a large spatial extent with respect to our overlapping partitions means both that it is relatively stable in terms of its curvature and that the underlying structure must be reasonably well defined for the edge extraction process to have produced a coherent unit.

In Figure 7(g), we have overlaid the original set of arc descriptors on the edge gradients. This provides a clearer sense of the relative consistency of the description process with the underlying edge events, than does the overlay on the edge regions.

4.3 Road Scene

The order of figures for this example parallels closely that of the previous example. Figure 8(a) is a 127 × 127 subimage of the green spectrum of an original 256 × 256 color image a local road scene. Figure 8(b) is the corresponding edge image. Figure 8(c) is the entire set of arc descriptors, with majority support and *curve support regions* with size $\geq$ 5 pixels, overlaid on the edge image.

The unshifted and shifted curvature partitions are represented in Figures 8(d) and 8(e) respectively. It appears that most of the macro-structure of the image is captured in the unshifted partition, Figure 8(d). Worth noting are the descriptor along the right edge of the road and the long horizontal arc in the middle of the image just at the tree/road boundary. The latter fits its region extremely well; equally important is the fact that they both satisfy our notions about the character of the underlying structure. Also of interest is the large circular arc at the upper-right tree/sky boundary. While there is not a tight correspondence between this descriptor and the geometry of its support region, there is arguably a real sense in which it captures the overall tendency/shape of the underlying image structure. The shifted version 8(e), on the other hand, consist primarily of foliage detail in the interior regions of the trees. There are notable exceptions however in the upper-left and middle portions of the foliage/sky boundary, where medium sized arcs of this shifted partition serve to terminate long diagonal and quasi-vertical lines of the boundary.

Figure 8(f), like its counterpart Figure 7(f), demonstrates the effect of pre-filtering for larger regions, this time ones whose size $\geq$ 10 pixels. We see many of the interior foliage descriptors disappear, but the most of the important tree/sky and tree/road boundary descriptors remain. Finally, Figure 8(g) establishes the same descriptor/edge-gradient correspondence as Figure 7(g).

5. CONCLUSIONS

We have presented a method to derive circular arc descriptors for discrete segments of edge contours. Each descriptor is local and independent—summarizing the planar distribution only of that region which *supports* it. Descriptors possess attributes, and filtering mechanisms may be employed in order to facilitate their use as primitives by subsequent grouping processes. The advantages of the algorithm include its simple local character and the fact that it is robust in the face of moderate amounts of noise. This latter is due in part to the coarse partitioning of the output curvature measure .

ACKNOWLEDGEMENTS

All work was conducted at the Computer Vision Laboratory of the University of Massachusetts, Amherst. Thanks are due to Prof. Allen Hanson and Prof. Edward Riseman for their continued support and encouragement, and to the many researchers who developed and continue to develop the UMass VISIONS system. Special thanks to Brian Burns for his untiring explanations and perceptive suggestions, and to Michael Boldt for his many encouragements and insights.

REFERENCES

[1] Asada, H. and M. Brady, "The Curvature Primal Sketch," *Massachusetts Institute of Technology AI-Memo 758,* 1984.

[2] Marimont, D., "A Representation for Image Curves," *Proc. AAAI,* pp.237-242, 1984.

[3] Hoffman, D.D. and W.A. Richards, "Representing Smooth Plane Curves for Recognition: Implications for Figure-Ground Reversal," *Proc. AAAI,* pp.5-8, 1982.

[4] Duda, R. and P. Hart, *Pattern Classification and Scene Analysis,* Wiley Interscience, New York, 1973.

[5] Burns, J.B., A. Hanson and E. Riseman, "Extracting Linear Features," *Proc. of the Seventh International Conference on Pattern Recognition,* 1984 (to appear in IEEE-PAMI).

[6] Kitchen, L. and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters 1,* pp.95-102, 1982.

[7] Nagel, H.H., "Displacement vectors derived from second-order intensity variations in image sequences," *CVGIP*, 21, pp.85-117, 1983.

[8] Haralick, R., "Digital step edges from zero-crossings of second directional derivatives," *IEEE-PAMI*, 6, 1, pp.58-68, January 1984.

[9] Agin, G., "Fitting Ellipses and General Second-Order Curves," *Carnegie-Mellon University , Robotics Institute Memo,* 1981.

FIGURES

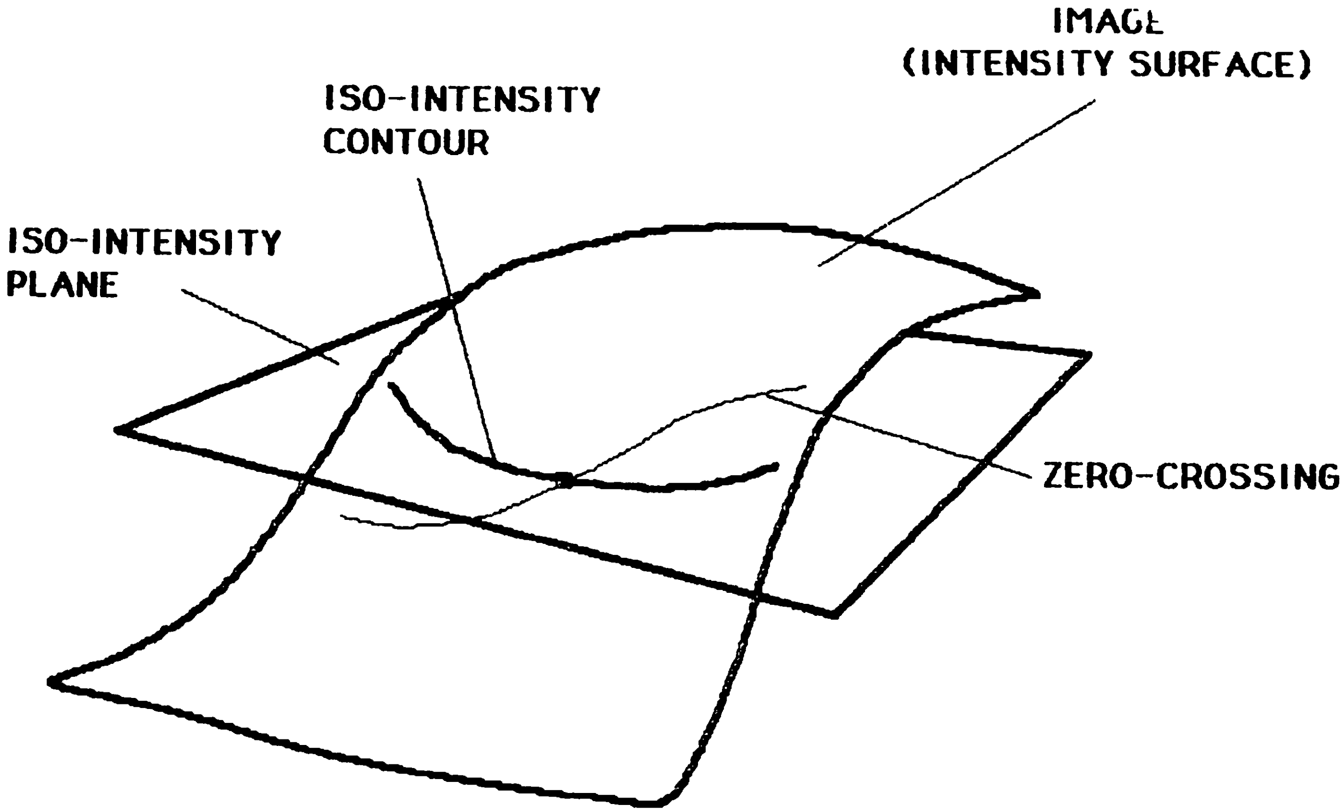

Figure 1 Kitchen/Rosenfeld Curvature Operator. Gives the turning rate of the gradient, about a particular pixel, projected along the idealized edge. This can be shown to be equivalent to the curvature of the iso-intensity contour passing through the pixel.

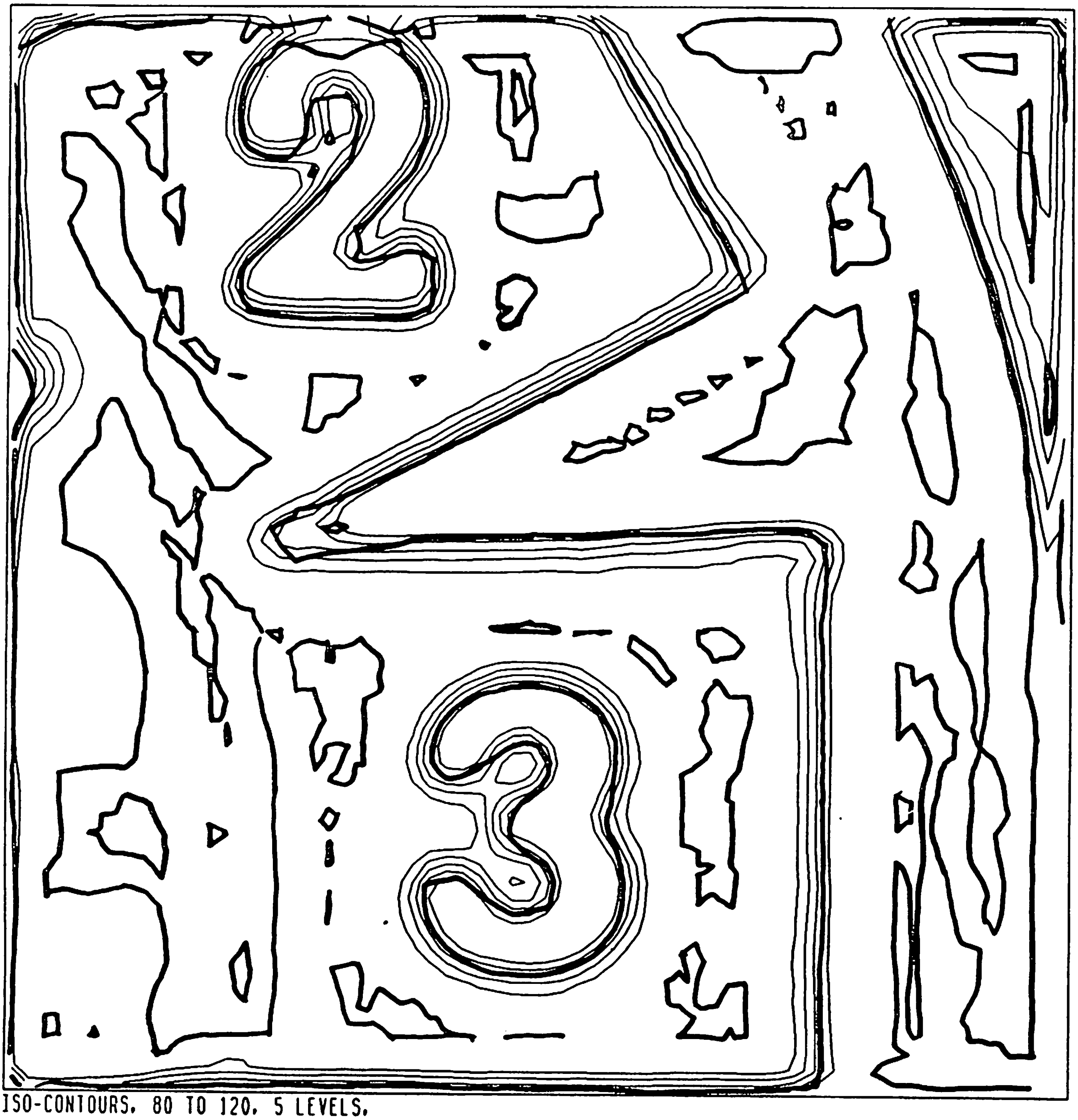

Figure 2 Zero-crossing contour on iso-intensity contours.

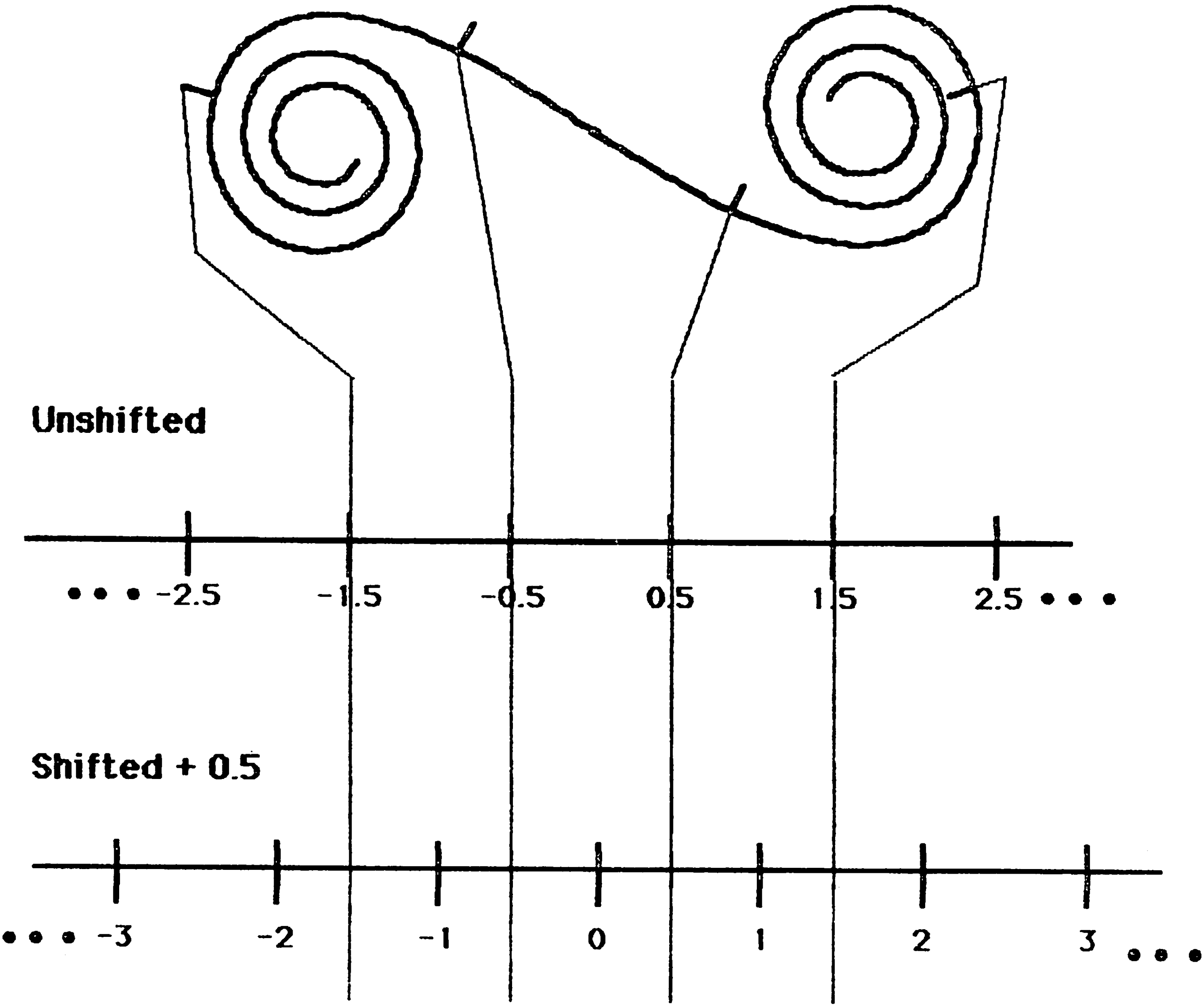

Figure 3 Example of default uniform linear quantization of grey-level curvature.

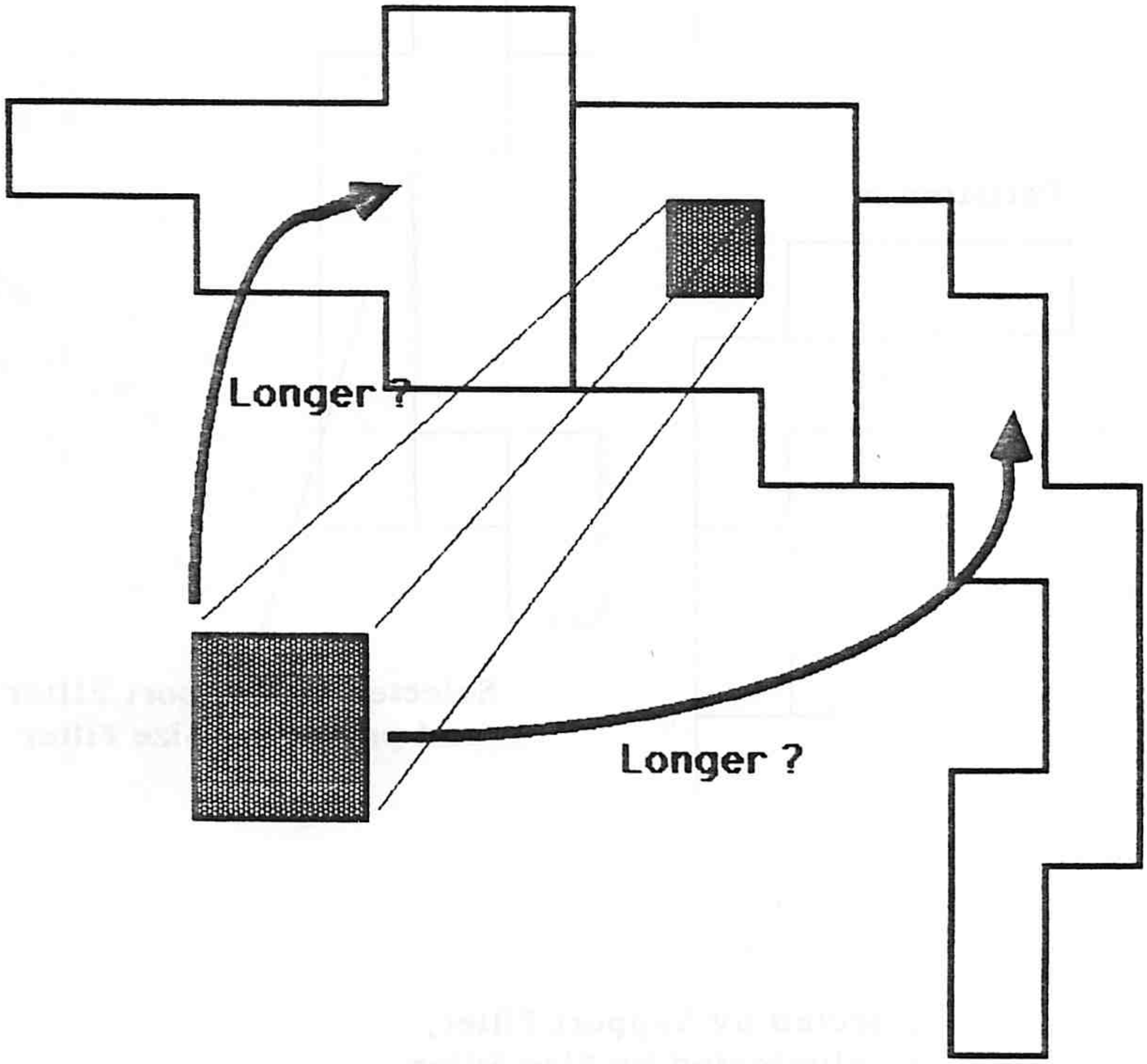

Figure 4 Voting scheme. Each pixel votes to support whichever region is longer of those having it as a member.

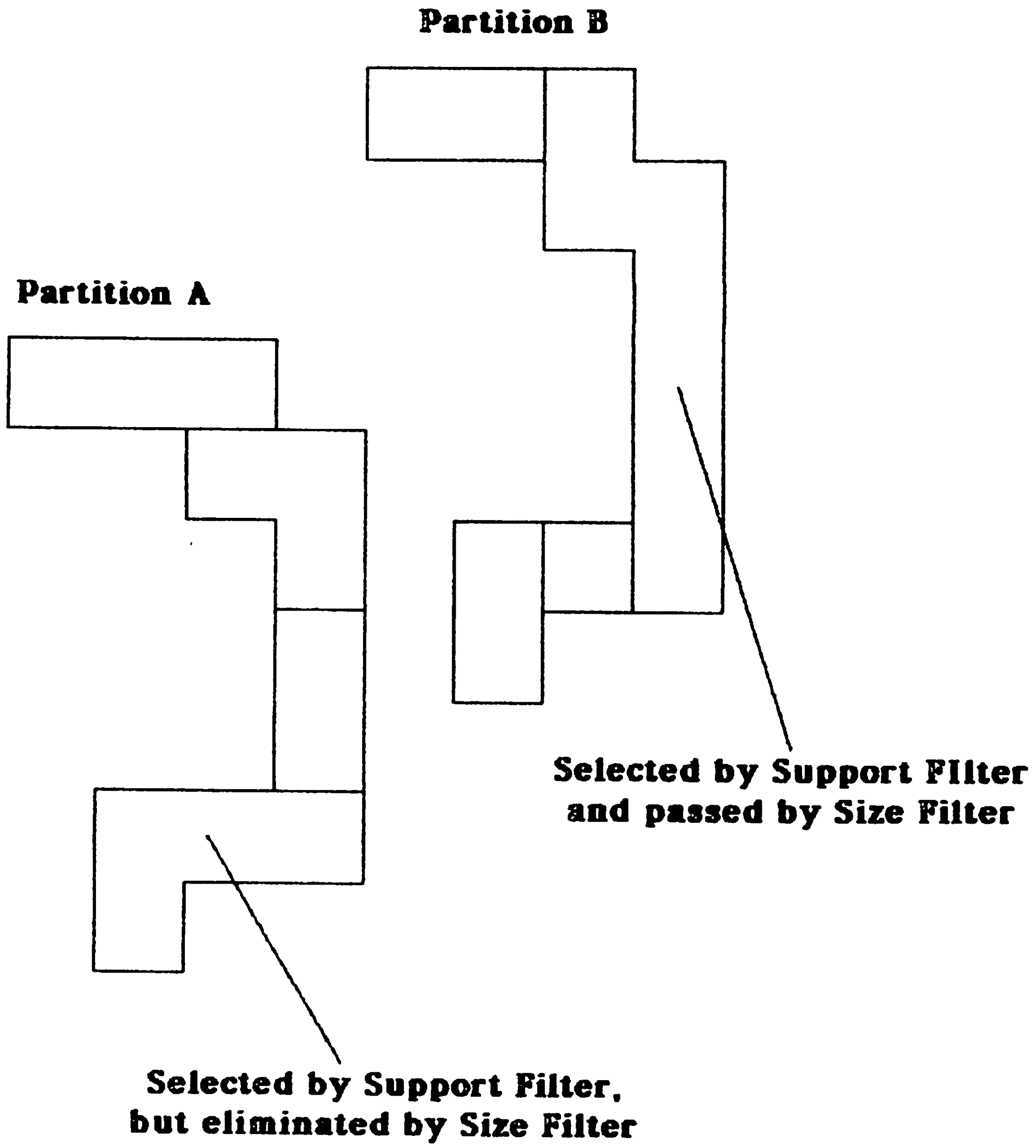
Partition B
Partition A
Selected by Support FIlter
and passed by Size Filter
Selected by Support Filter,
but eliminated by Size Filter

Figure 5 Region filtering.

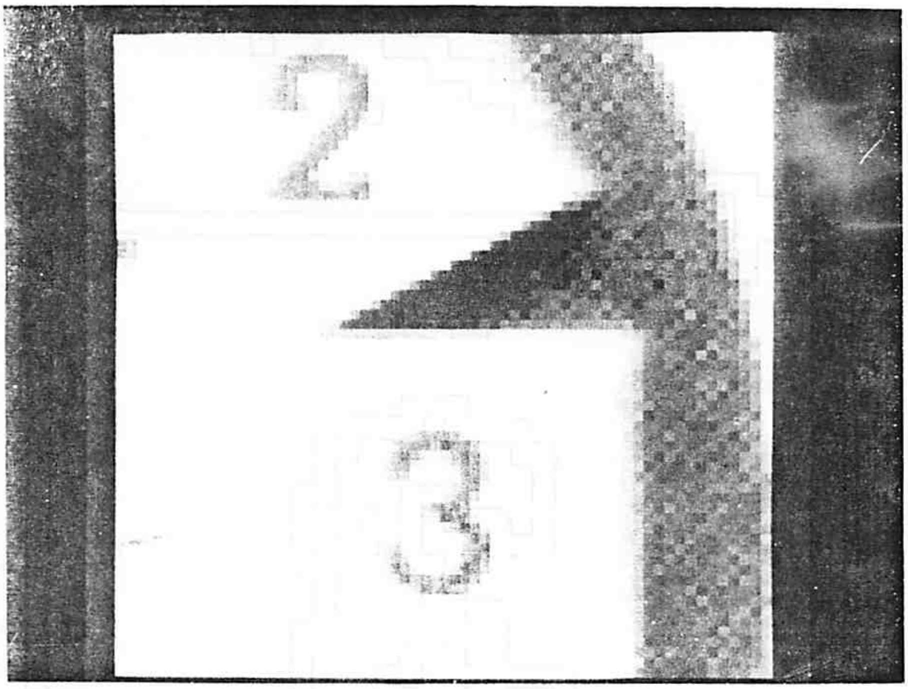

Figure 6(a) Clock image: original intensity data.

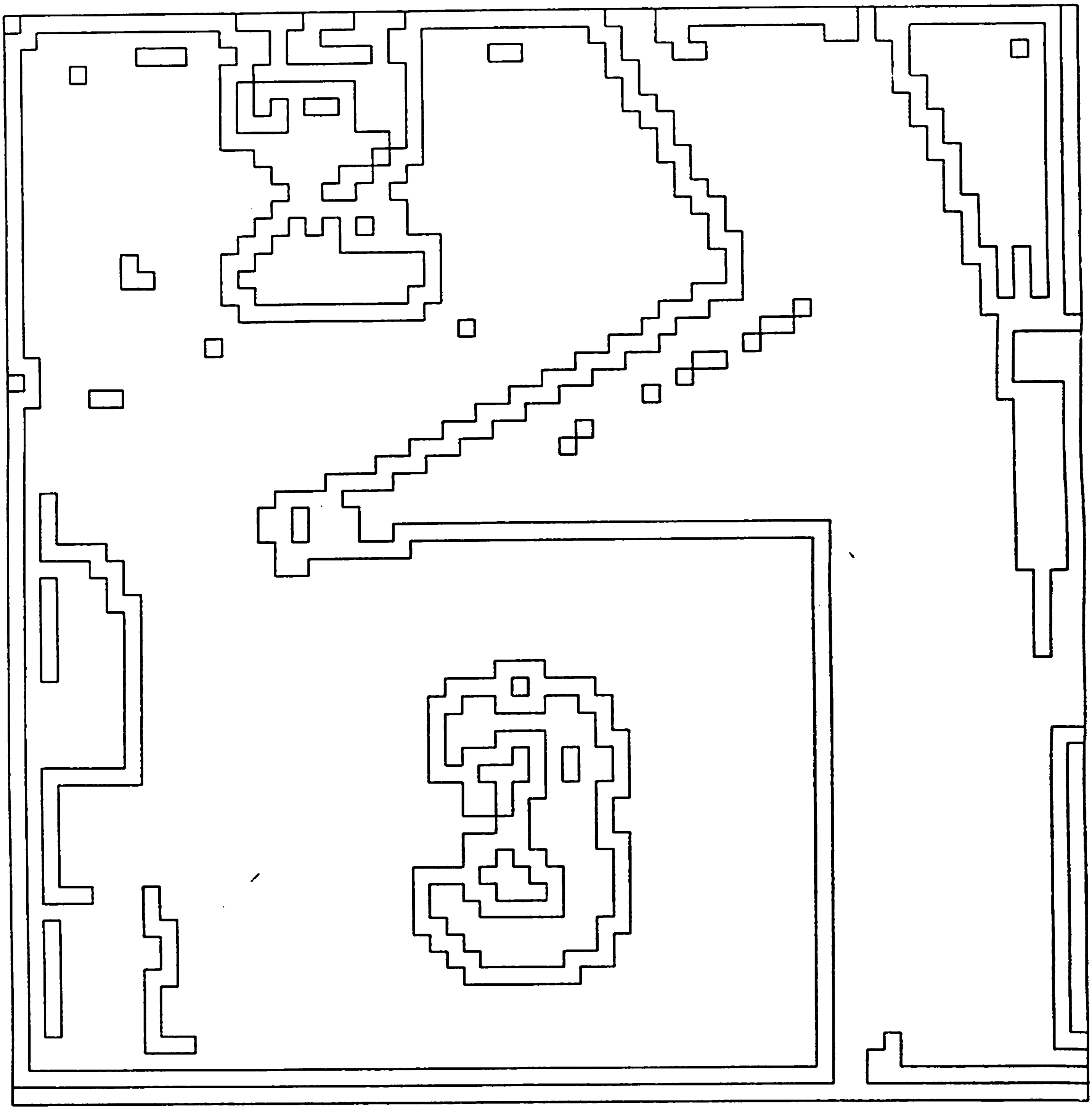

Figure 6(b) Clock image: edge regions.

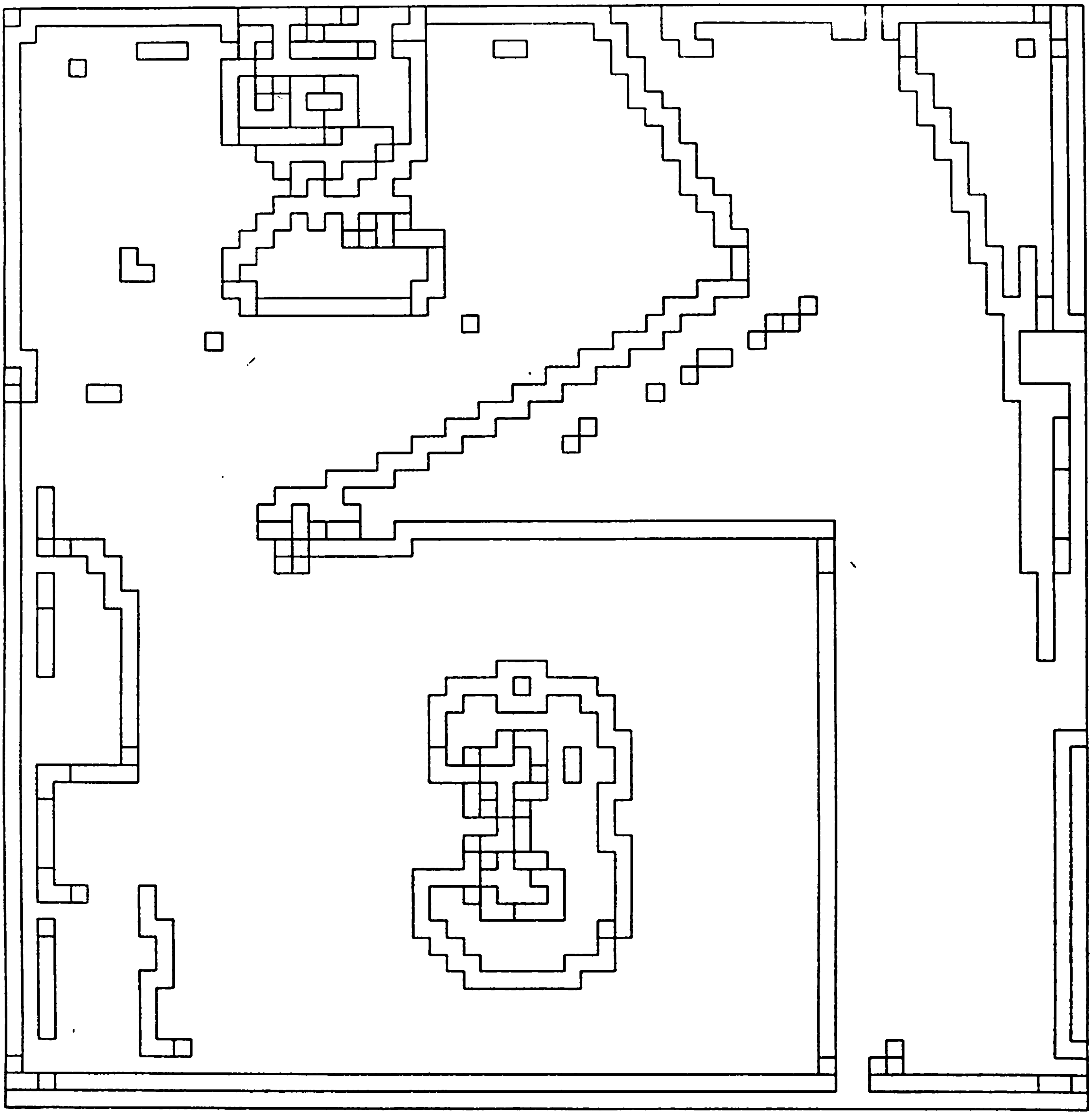

Figure 6(c) Clock image: unshifted support regions.

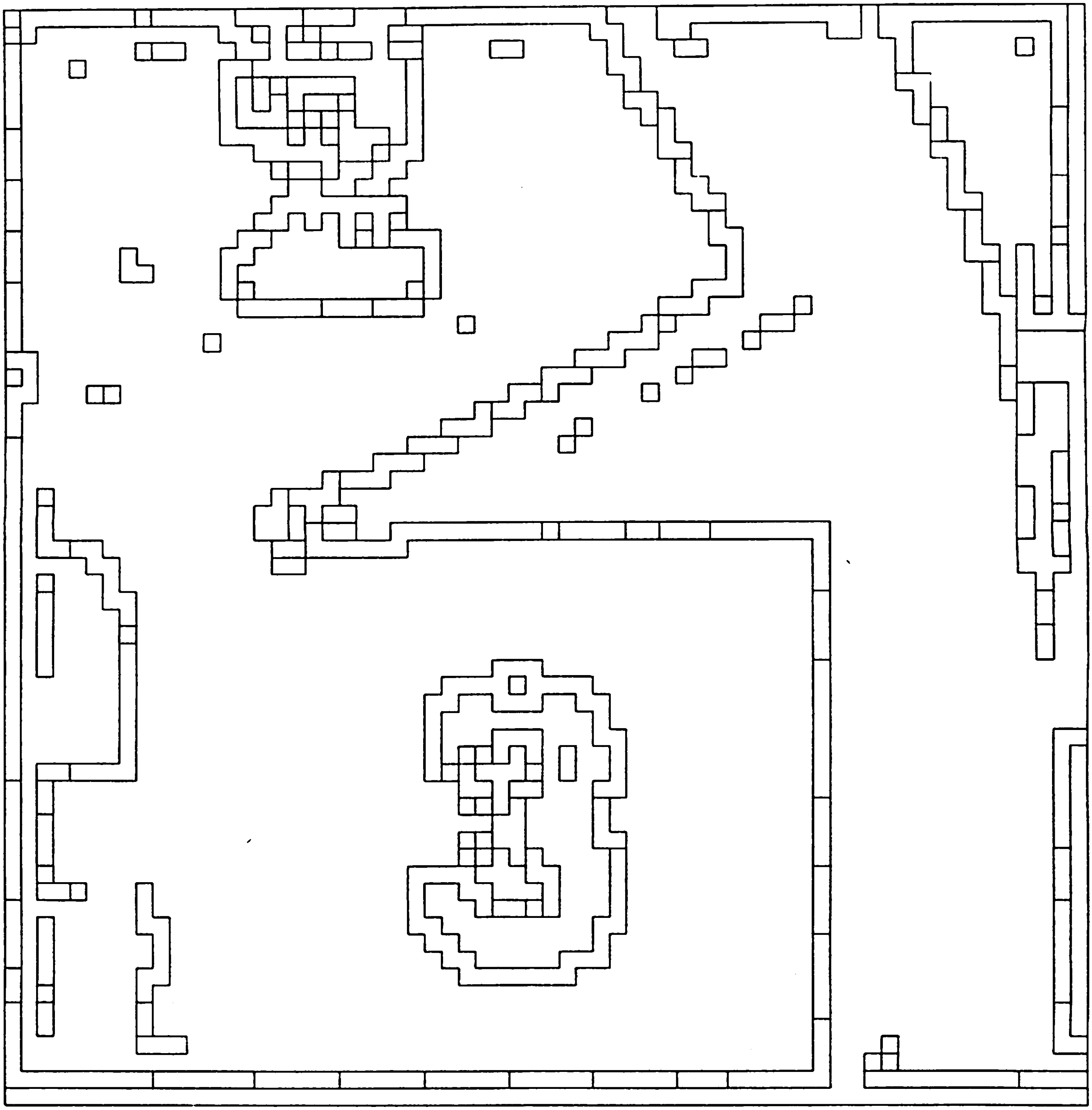

Figure 6(d) Clock image: support regions, shifted 0.5.

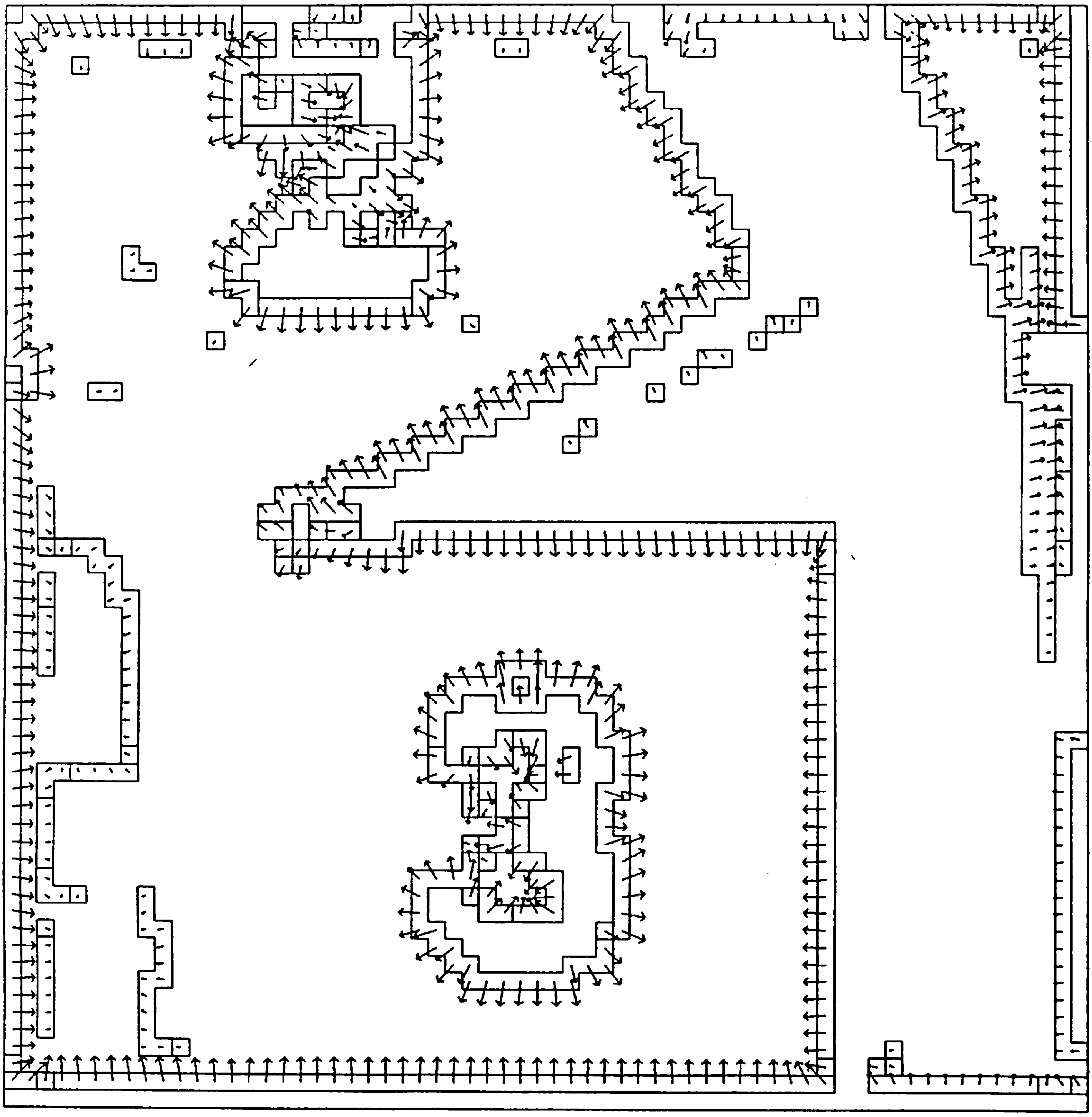

Figure 6(e) Clock image: edge gradients on unshifted regions.

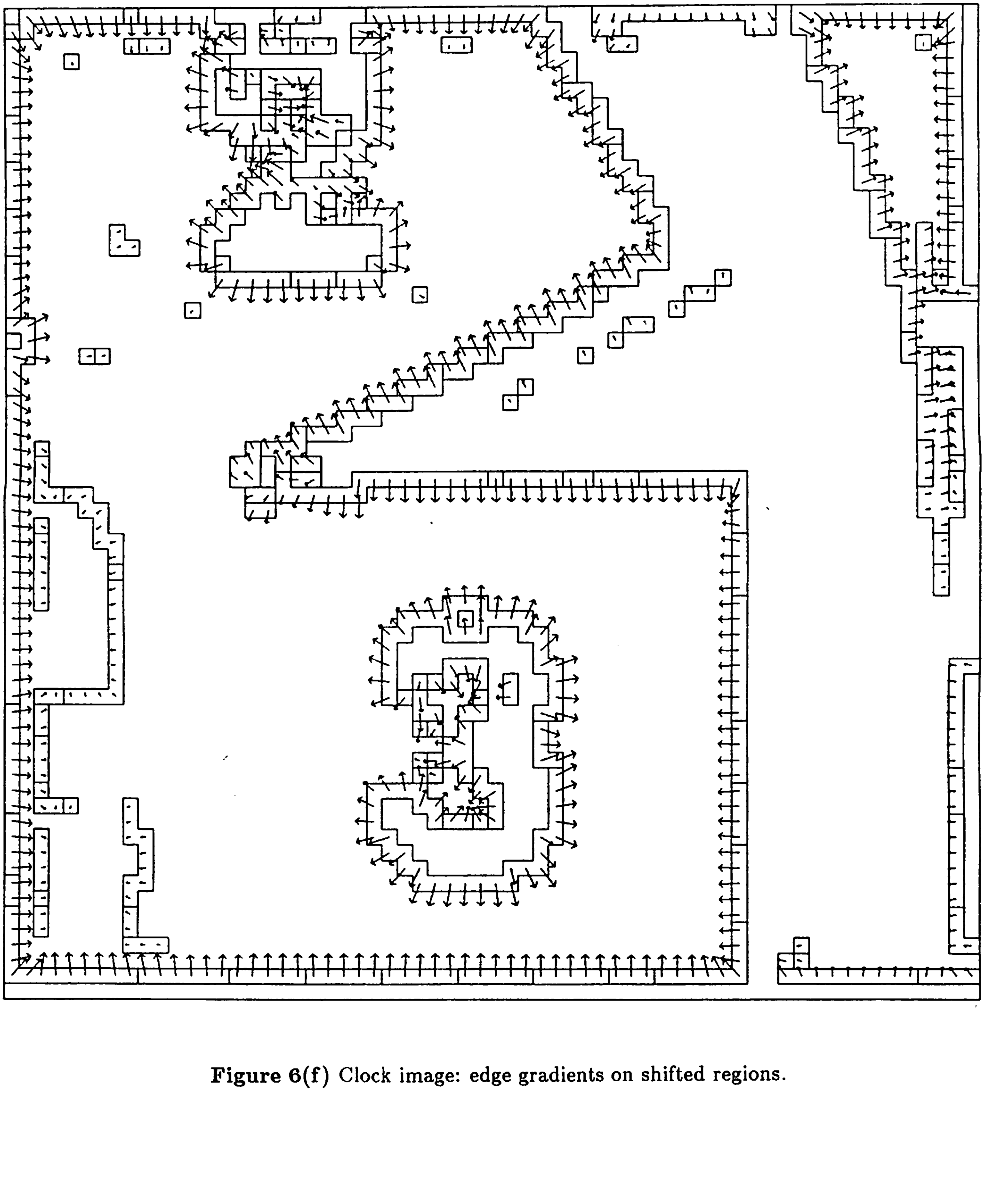

Figure 6(f) Clock image: edge gradients on shifted regions.

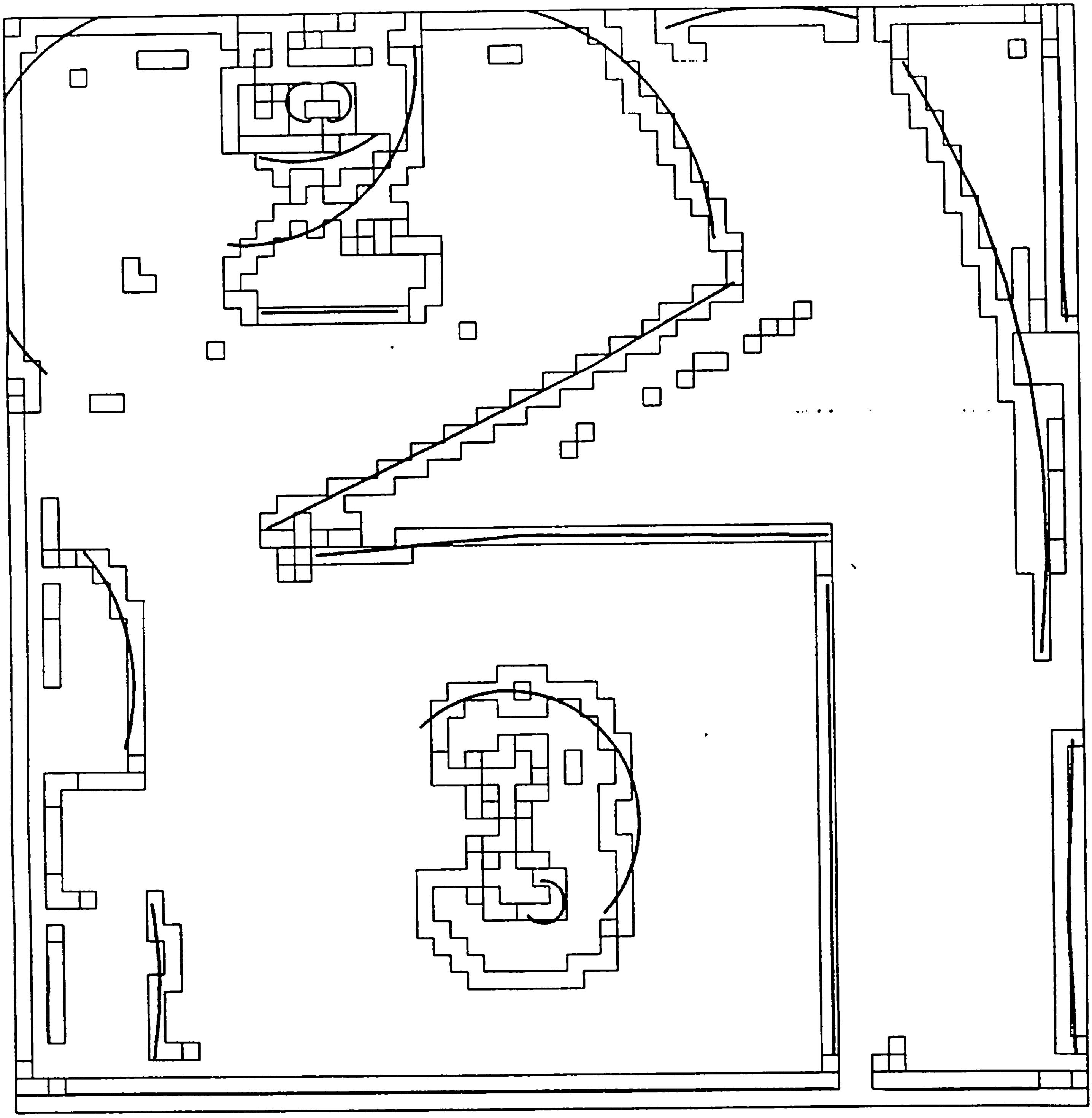

Figure 6(g) Clock image: arc descriptors on unshifted support regions.

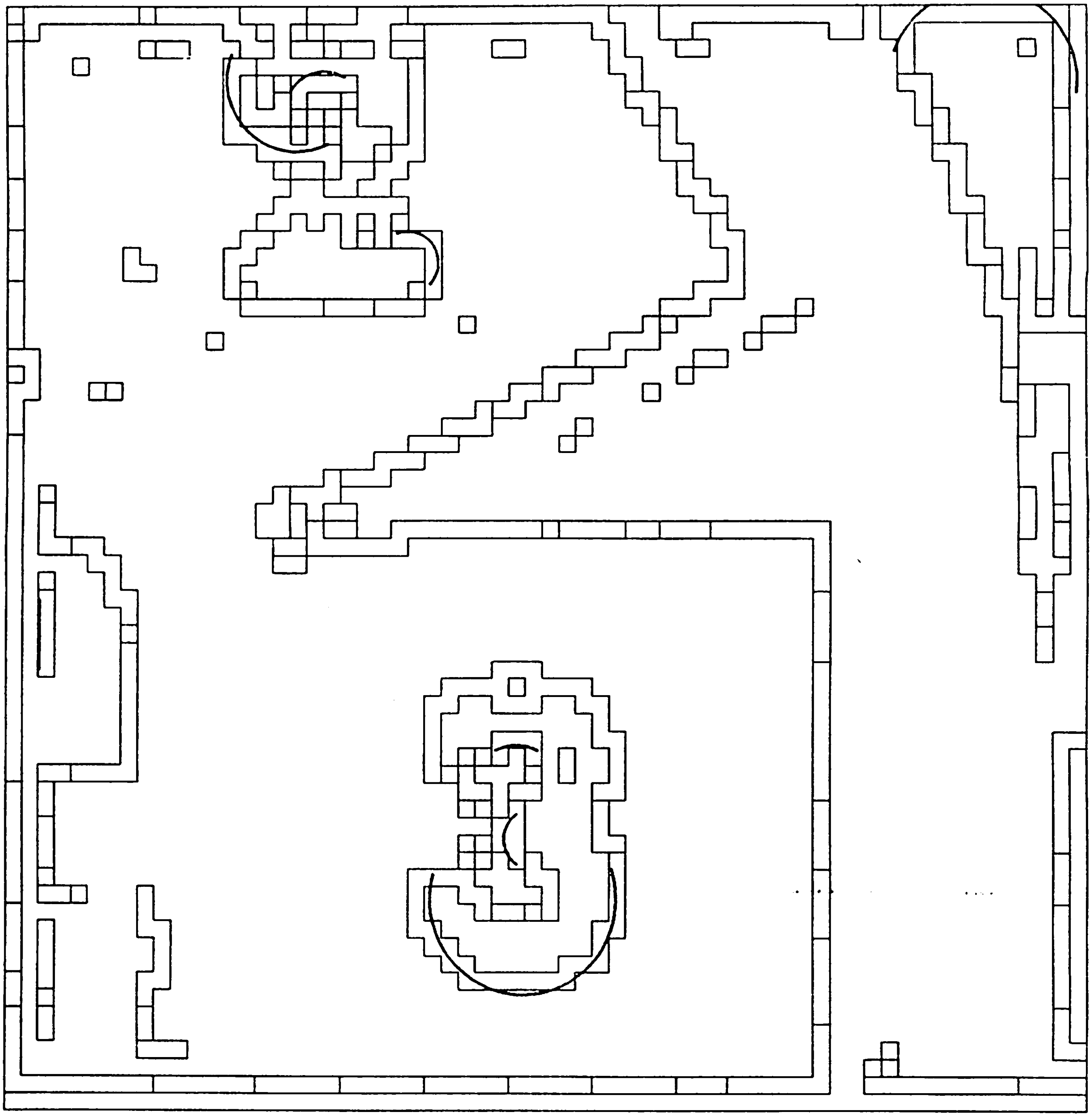

Figure 6(h) Clock image: arc descriptors on shifted support regions.

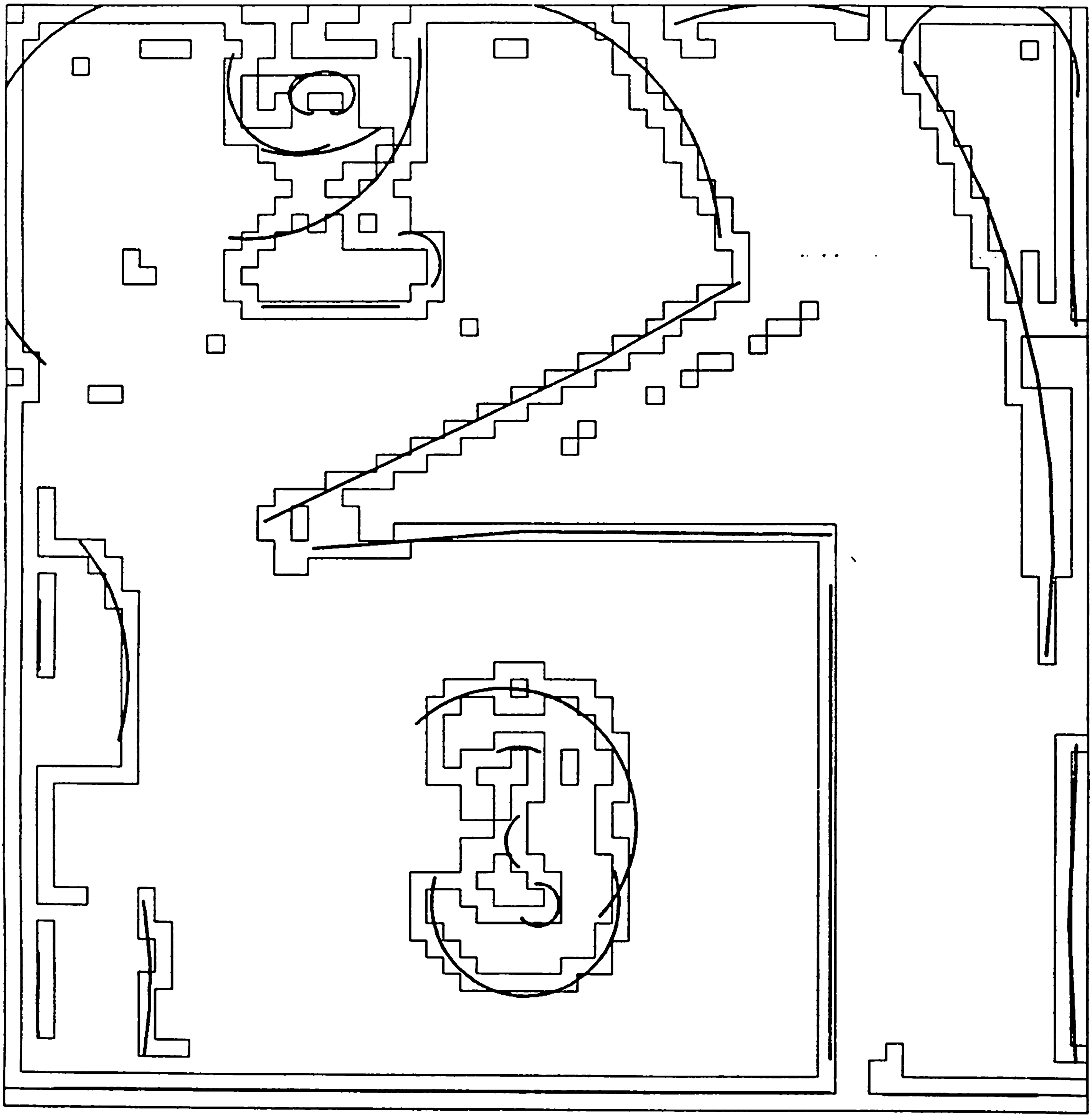

Figure 6(i) Clock image: all arcs on edge pixels.

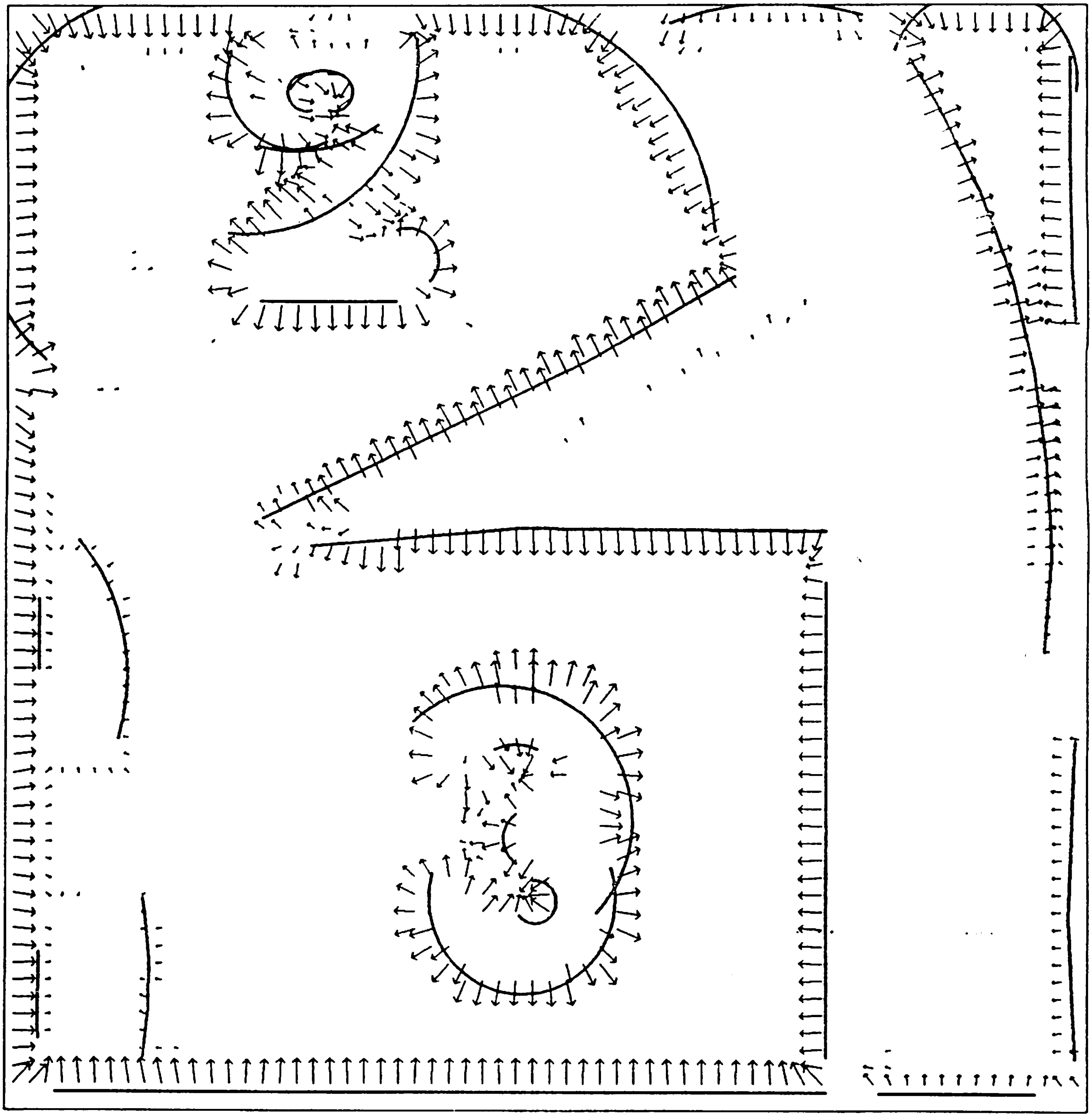

Figure 6(j) Clock image: all arcs on edge gradients.

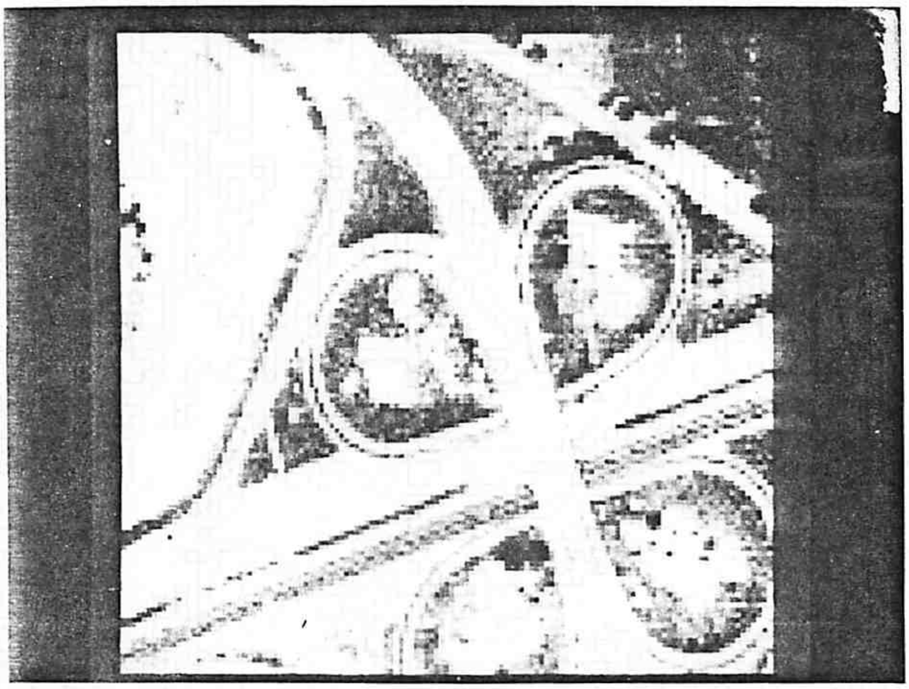

Figure 7(a) Aerial image: original intensity data.

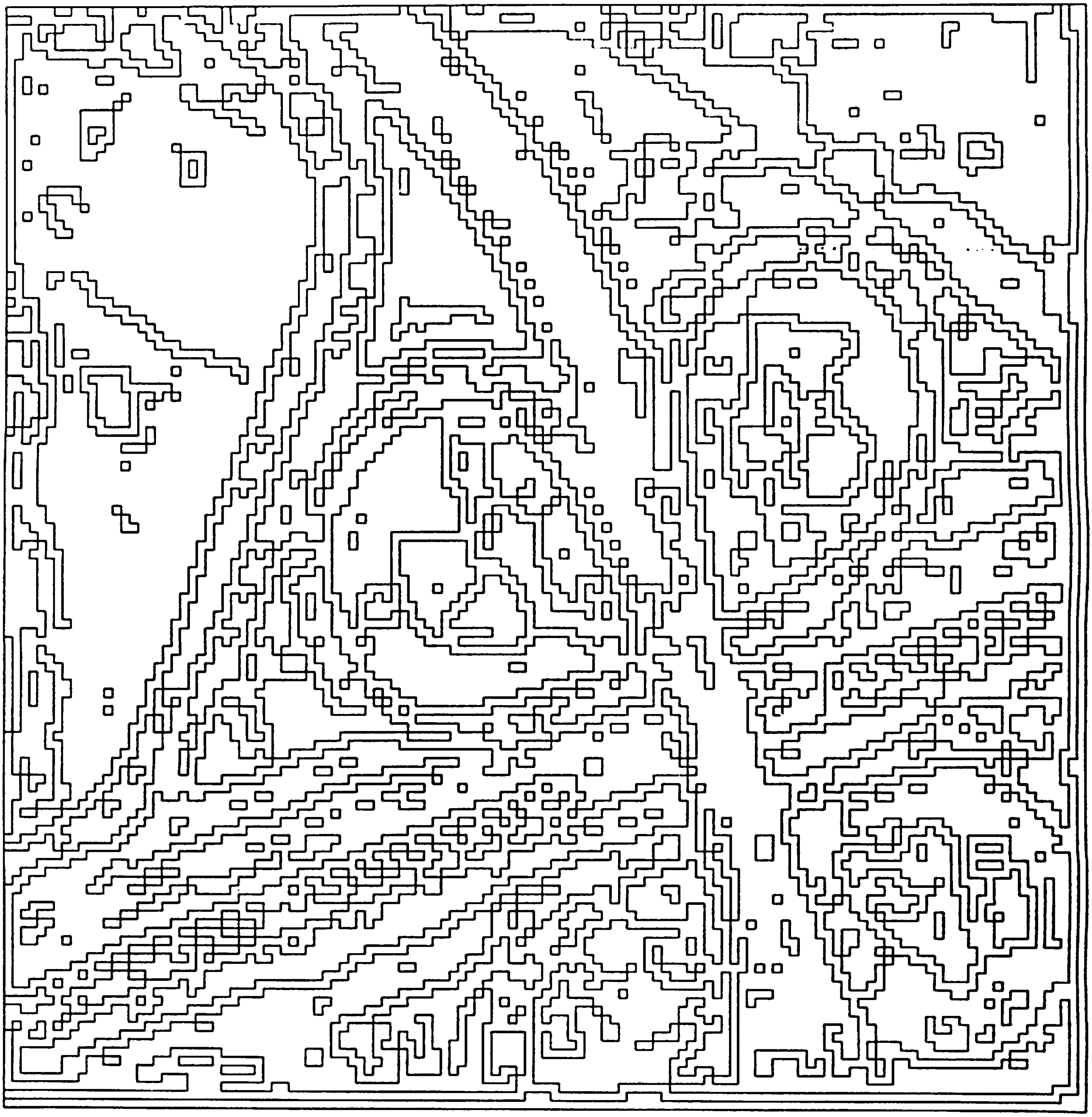

Figure 7(b) Aerial image: edge regions.

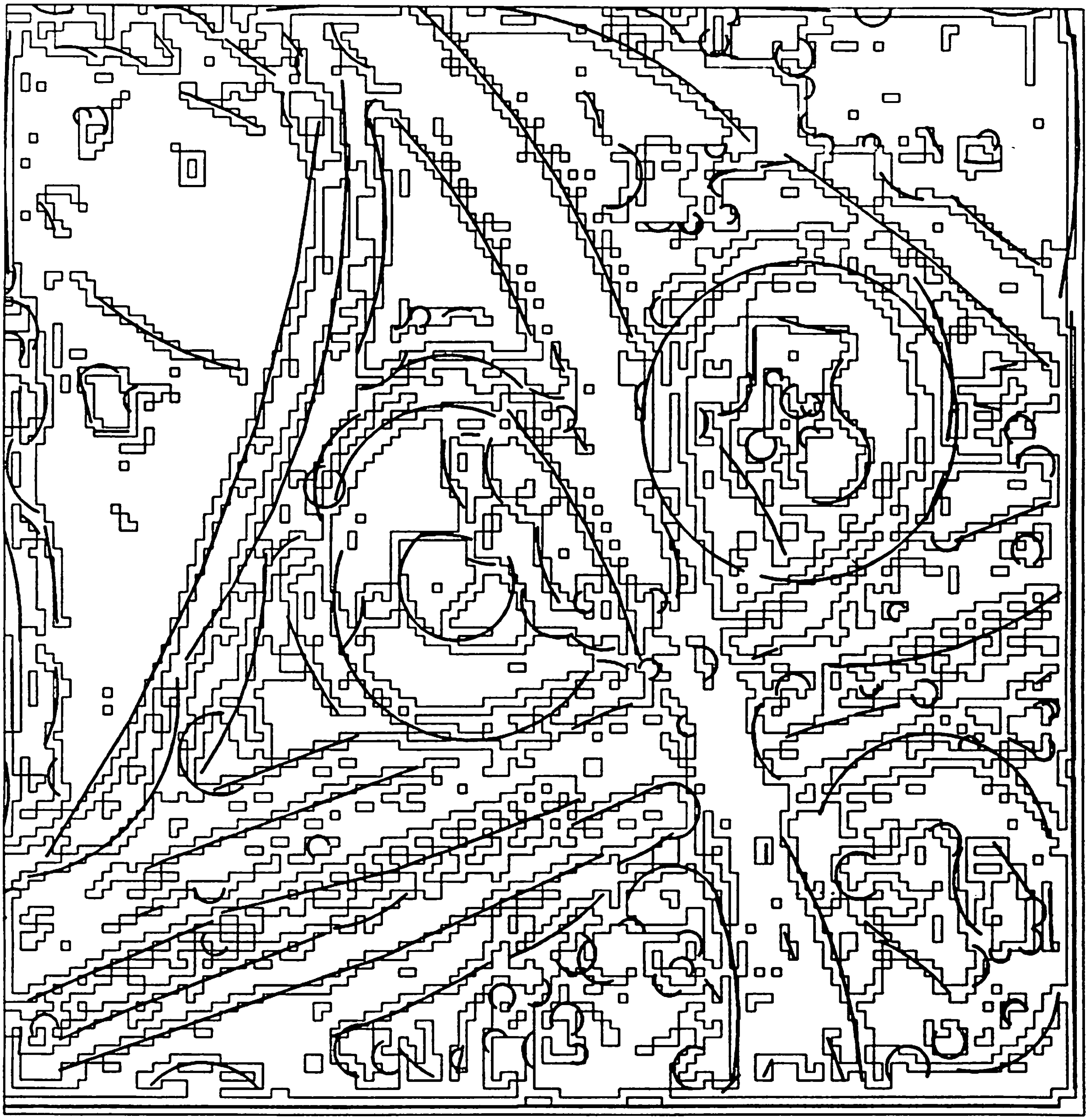

Figure 7(c) Aerial image: arc descriptors whose support regions have more than 5 pixels, overlaid on edge pixels.

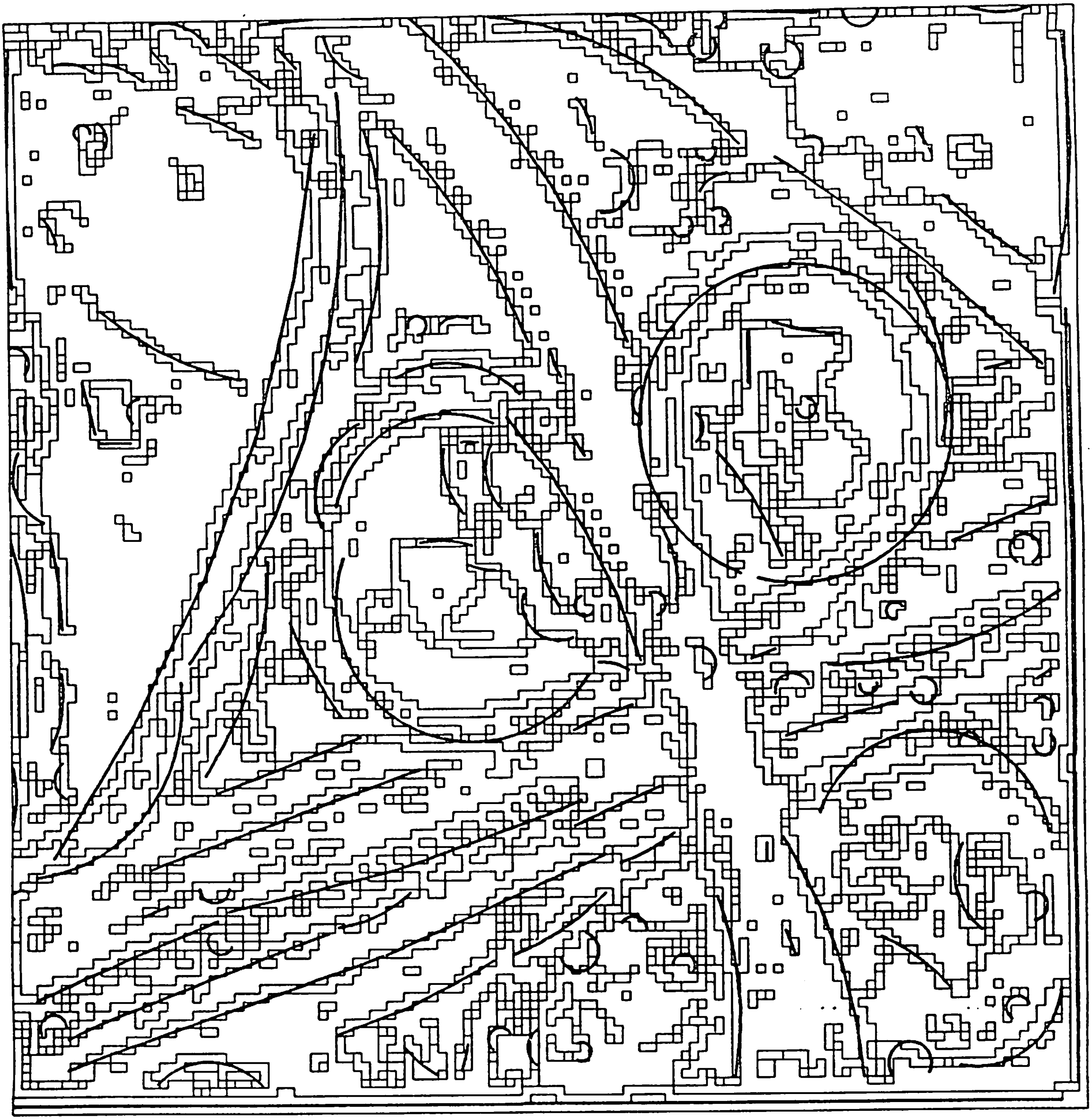

Figure 7(d) Aerial image: arcs on unshifted support regions.

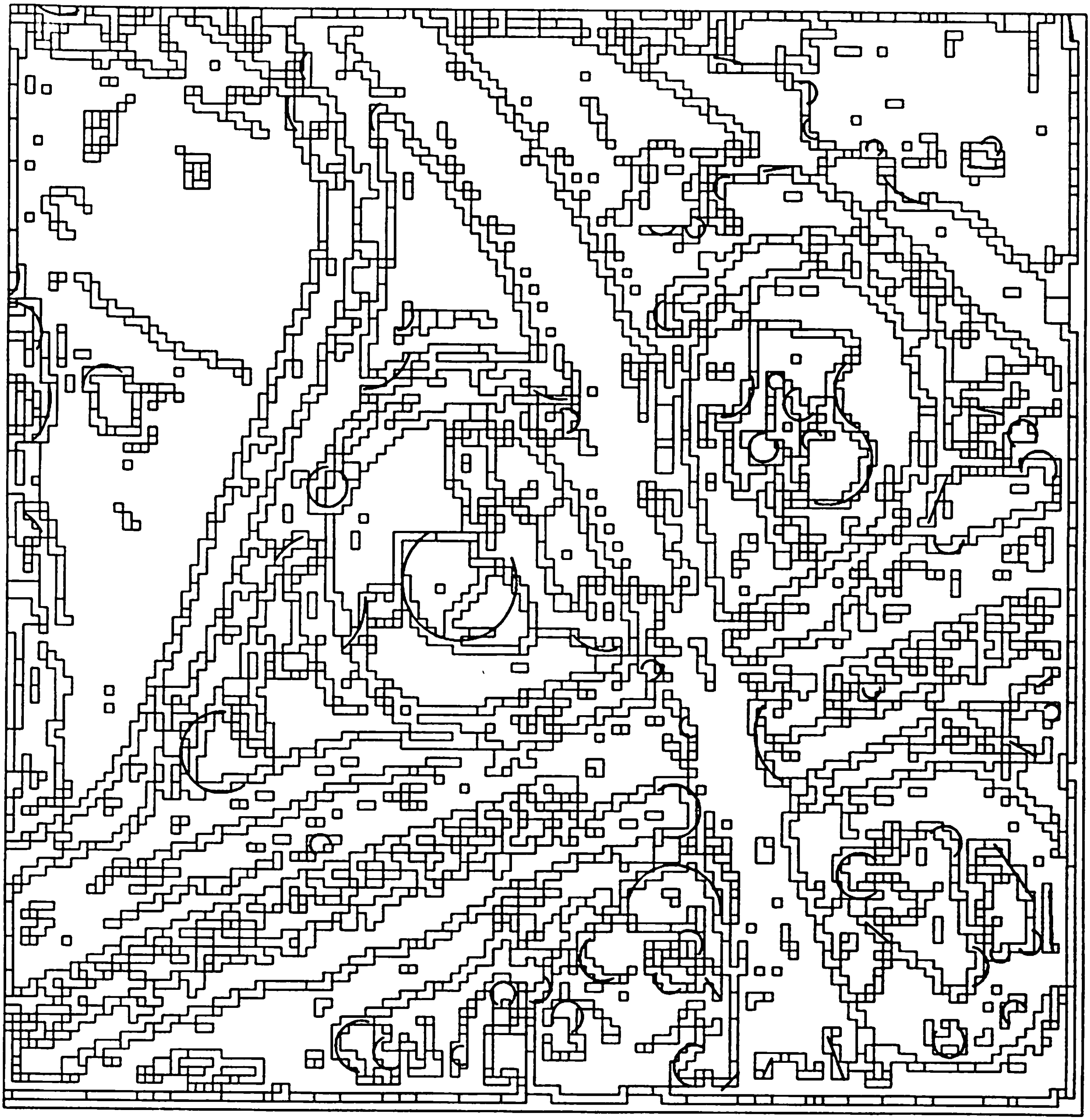

Figure 7(e) Aerial image: arcs on shifted support regions (shifted by 0.5).

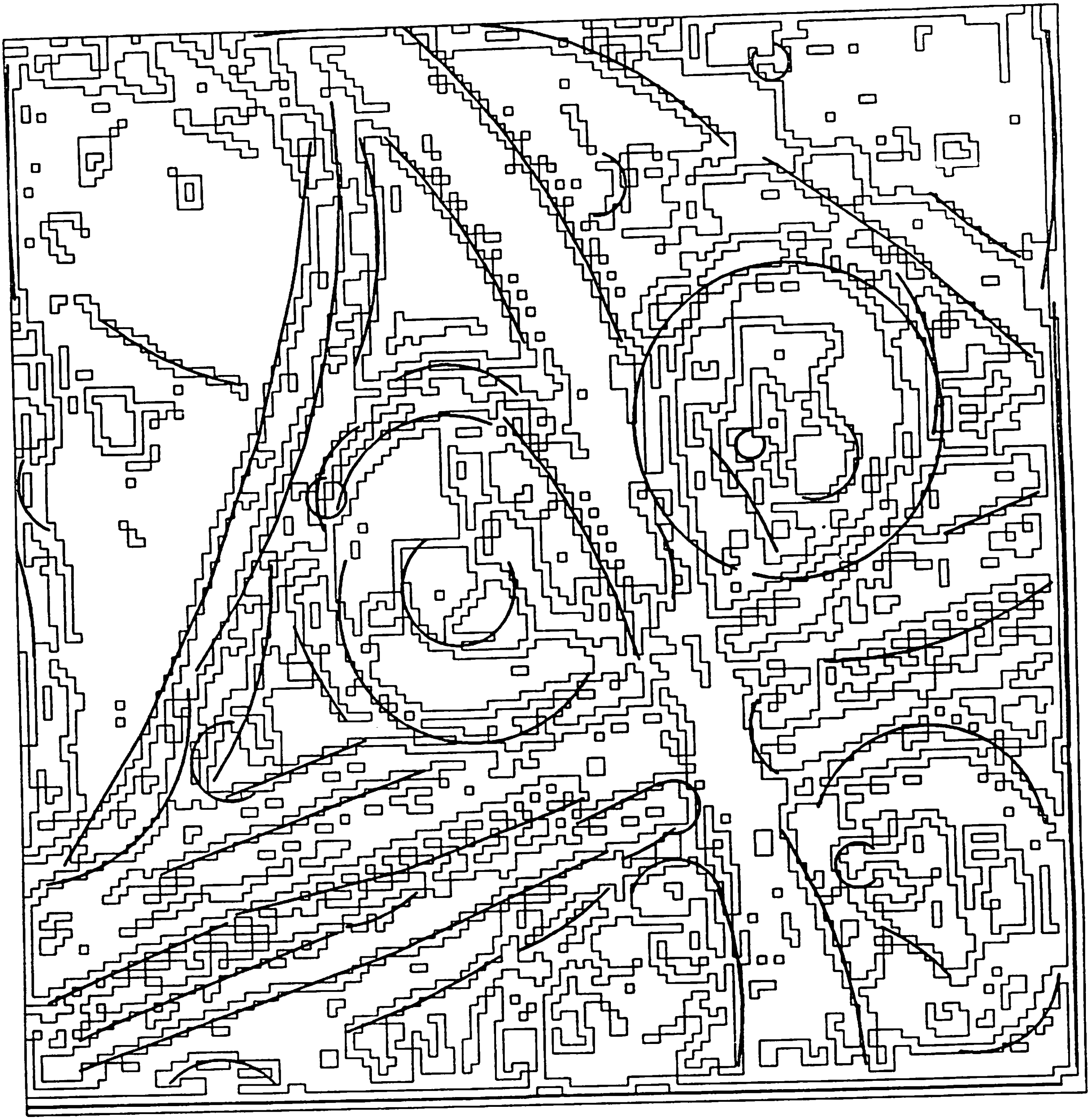

Figure 7(f) Aerial image: arcs whose support regions have more than 15 pixels, overlaid on edge pixels.

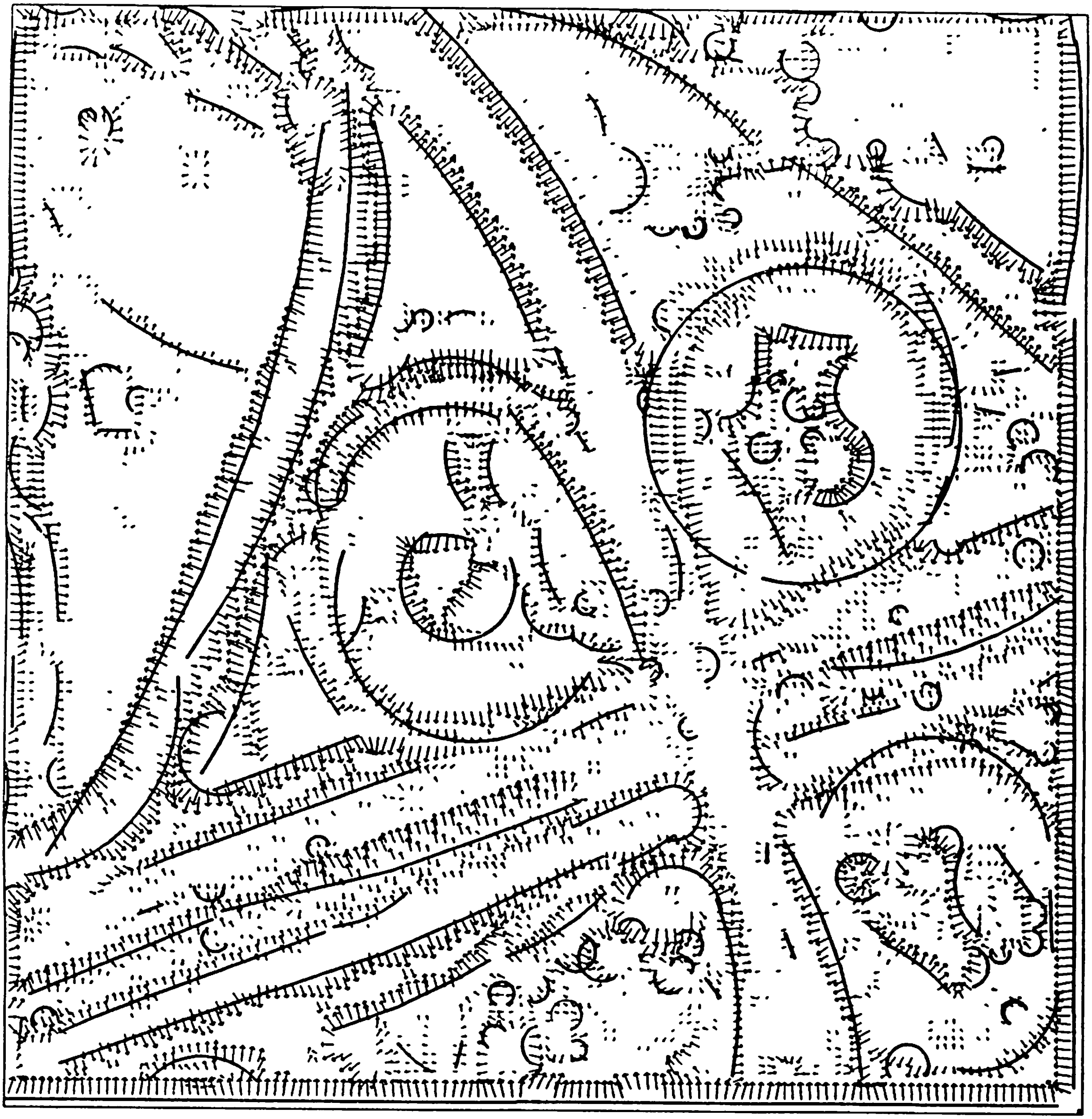

Figure 7(g) Aerial image: arcs on edge gradients.

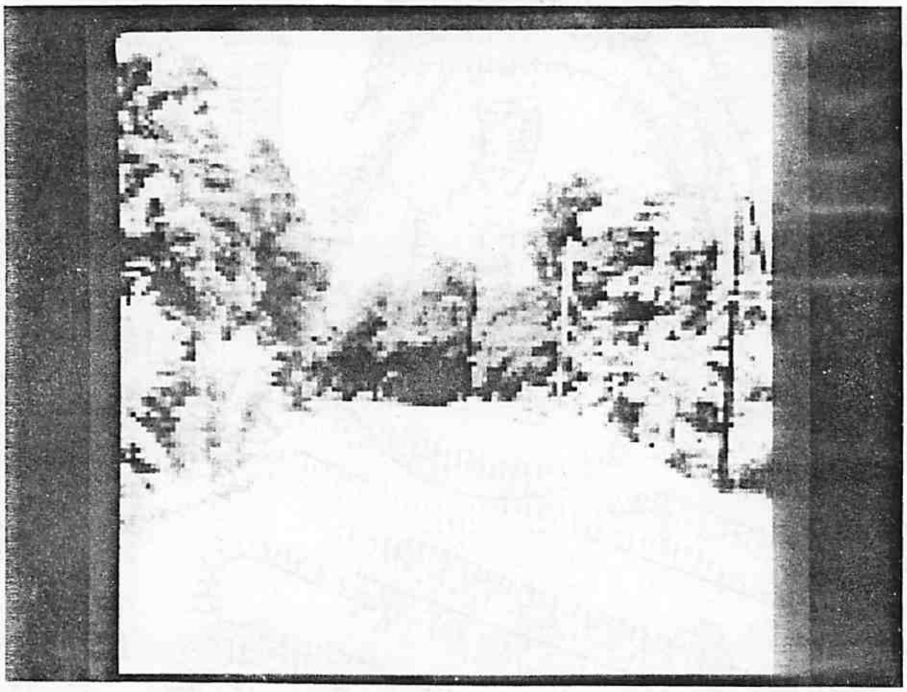

Figure 8(a) Road scene: original intensity data.

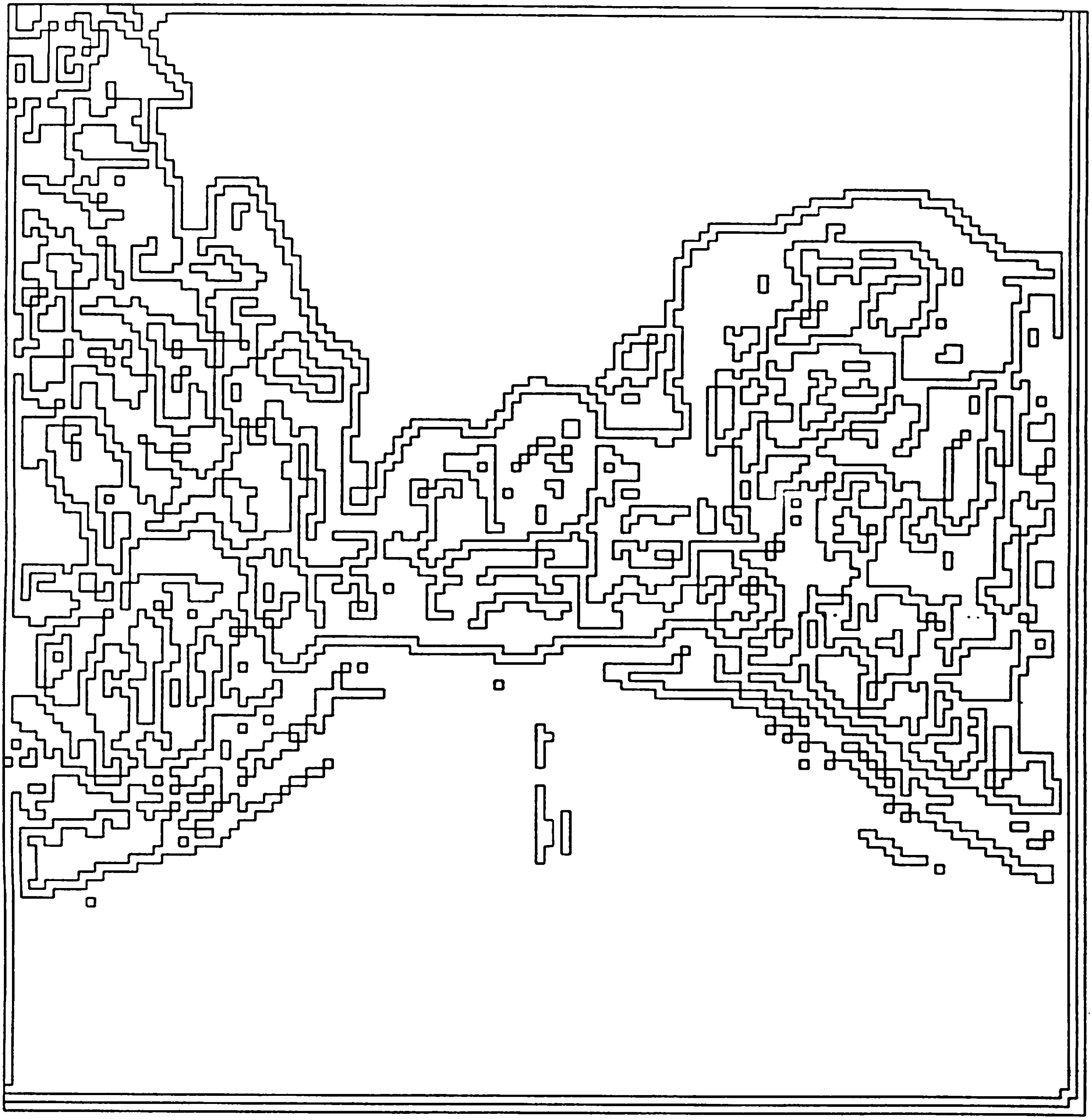

Figure 8(b) Road scene: edge regions.

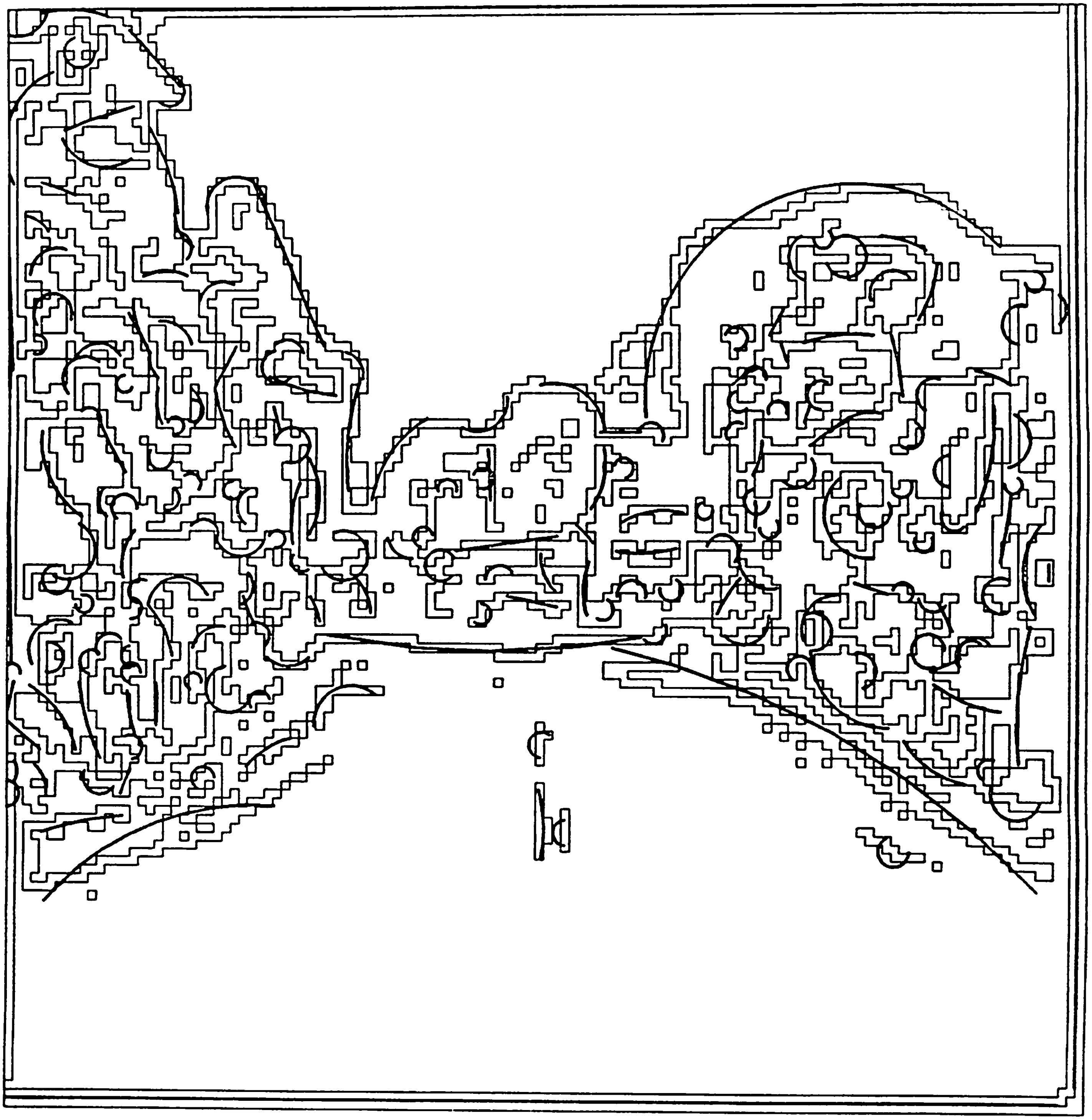

Figure 8(c) Road scene: arcs on edge pixels.

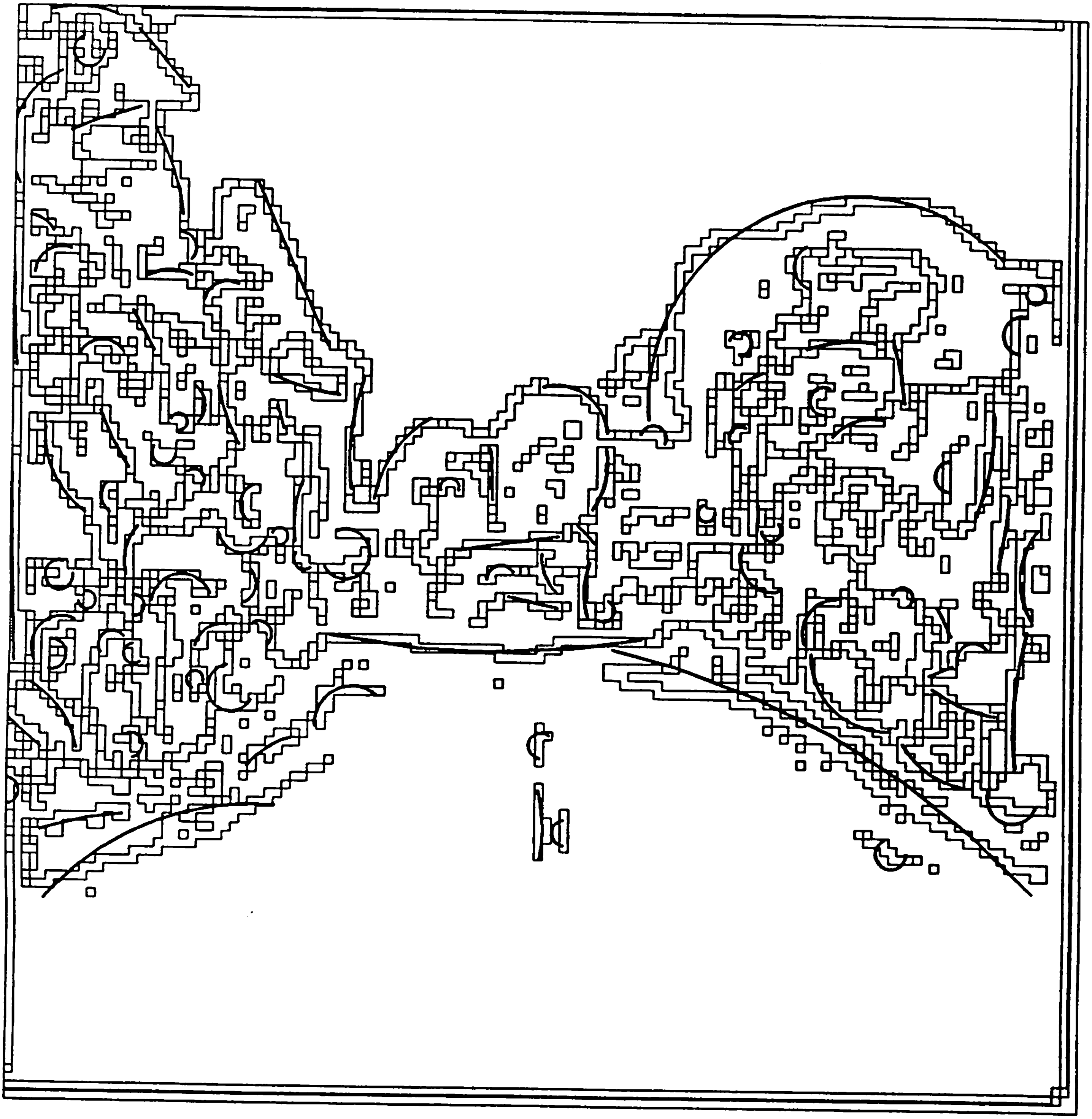

Figure 8(d) Road scene: arcs on unshifted support regions.

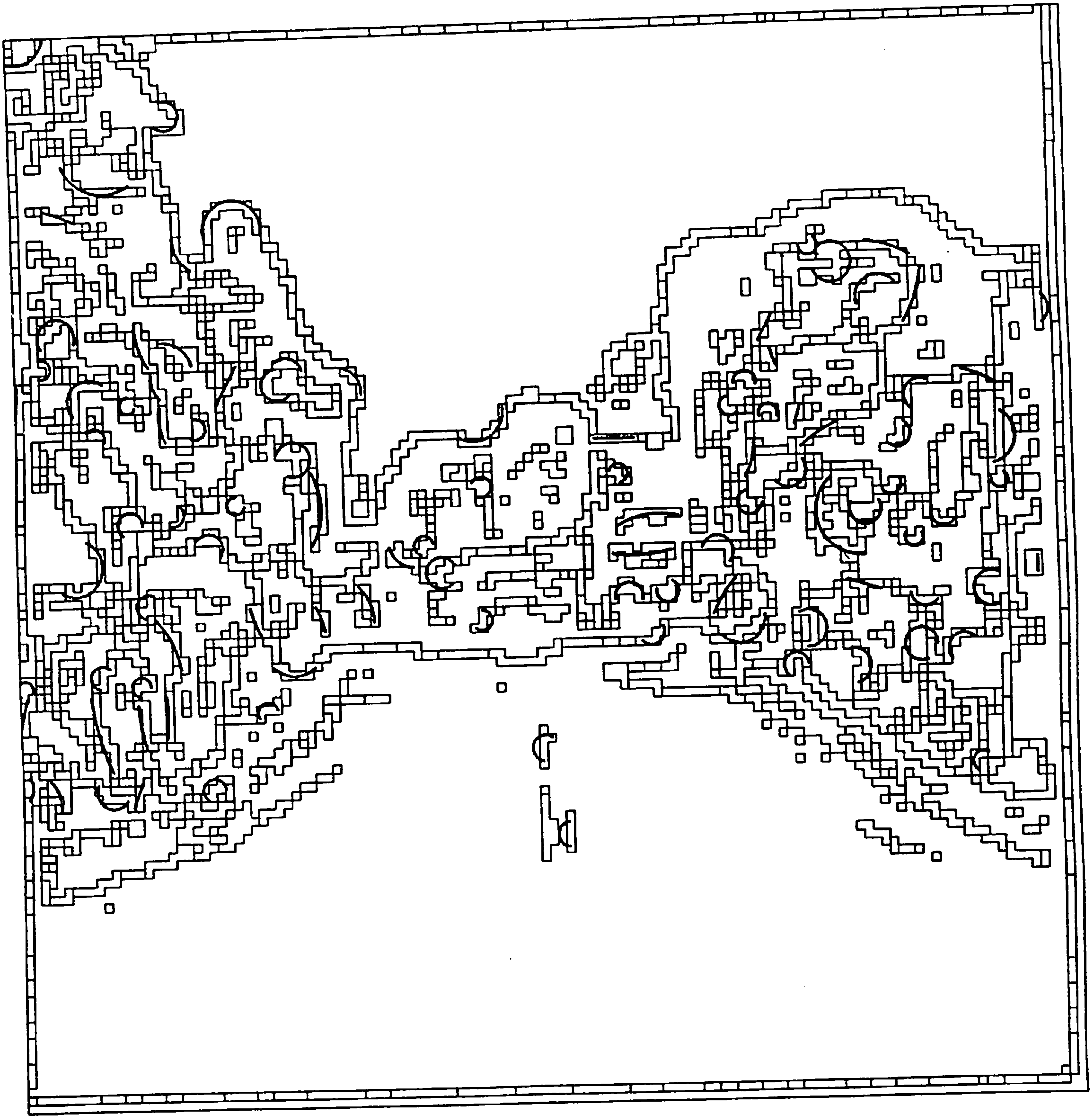

Figure 8(e) Road scene: arcs on shifted support regions (shifted by 0.5).

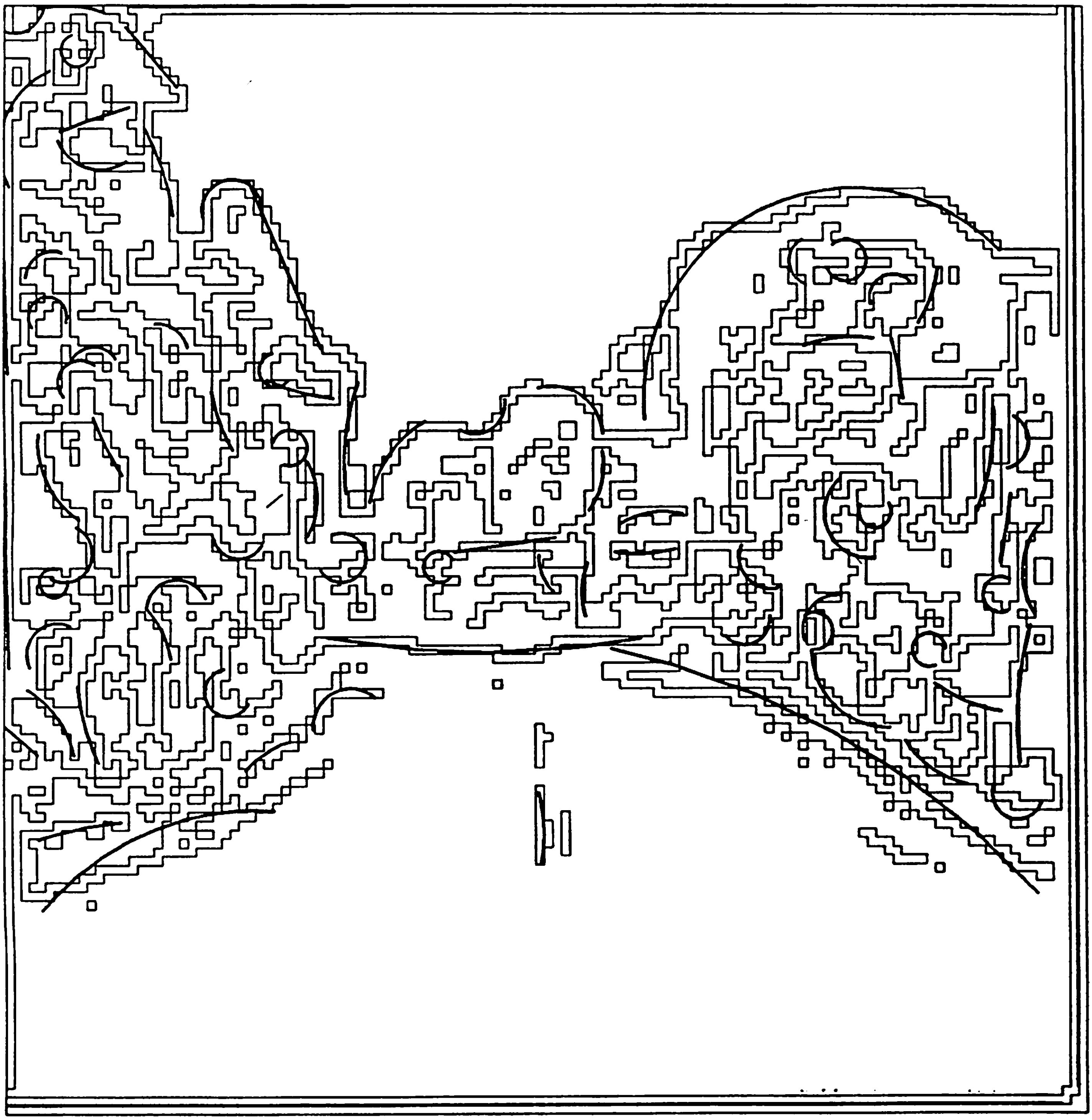

Figure 8(f) Road scene: arcs whose support regions have more than 10 pixels, overlaid on edge pixels.

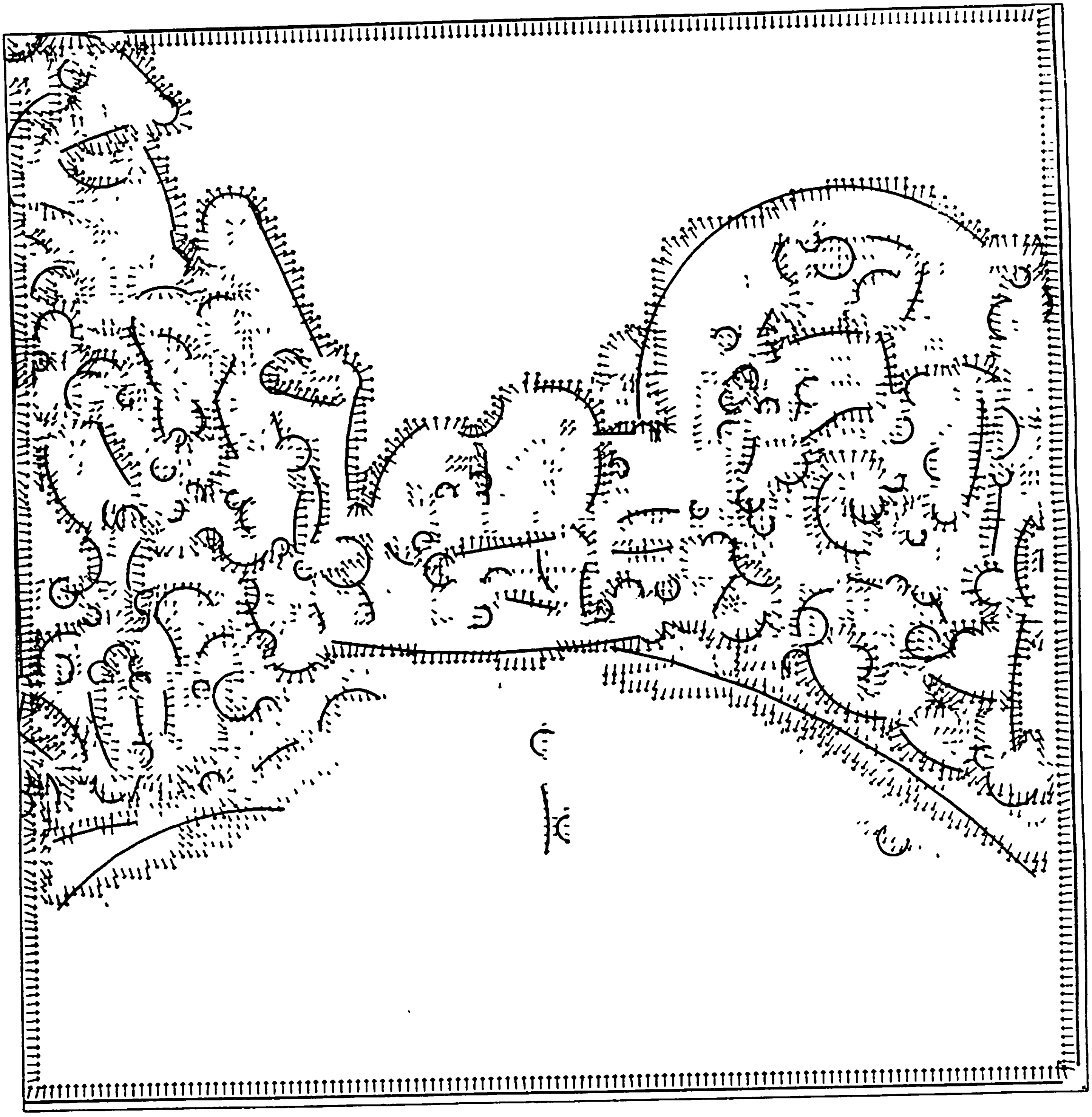

Figure 8(g) Road scene: arcs on edge gradients.