Border Extraction Using Linked Edge Pyramids

TSAI-HONG HONG, MICHAEL SHNEIER, MEMBER, IEEE, AND
AZRIEL ROSENFELD, FELLOW, IEEE

***Abstract*—A method of extracting borders of objects in a noisy image is described. The method is based on constructing a "pyramid" of reduced-resolution versions of the image; applying edge detectors at each level of the pyramid; and creating links between edge pixels at successive levels. The object border is detected at a high level, where the effects of the noise are reduced; and its position is determined more accurately by following the links from the detected border pixels down to the lowest level.**

I. INTRODUCTION

Finding the boundaries of objects is one of the most basic operations in image processing. This correspondence presents a new hierarchical approach that uses edge pyramids to extract the boundaries. By linking edges at different levels of the pyramid and scoring the edges based on degree of compatibility with other edges, it becomes possible to extract clean boundaries for objects of arbitrary size in a noisy image.

Multiresolution ("pyramid") image representations have been studied by a number of investigators; several papers dealing with this approach can be found in [12]. The simplest such representation is the gray-level pyramid, where each level is constructed from the preceding one by block averaging, e.g. over 2 by 2 blocks, yielding an image half the size (one-quarter the area) of the image on the preceding level. This correspondence, however, deals with edge pyramids rather than gray-level pyramids. (For a different approach to defining edge pyramids see Levine [7].)

In contrast to earlier work on edge and line pyramids [10], [3], the edge pyramid is constructed by applying an edge detector to an existing gray-scale pyramid. The earlier work involves first producing an edge (or line) picture, and then constructing a pyramid from that. Either method of constructing the pyramids can produce good results. However the existence of a gray-scale pyramid in registration with the edge pyramid opens the way for a fruitful cooperation between the two image representations.

A key advantage of the edge pyramid is that a compatibility or compactness score between edges can be computed on a truly local basis. This is because, at some level in the pyramid, edges that were far apart will become close. This is especially true for opposite edges of a single object, because the pyramid process is unlikely to cause such edges to disappear until the reduction in resolution has brought them close together.

The process to be described involves a number of stages. First, the gray-scale pyramid is constructed and an edge operator is applied to it at each level. The edges on adjacent levels are then linked to give a tree or set of trees, representing parts of borders in the image. The third step is to compute a score for each edge point at each level in the pyramid. The score measures how well the edge obeys requirements of good continuation and closure. Based on the linked pyramid and the set of scores, the edges can be projected down the tree to produce the boundaries of objects.

Examples are given that illustrate the application of the method to a large noisy object, to a set of objects of different sizes, and to a set of elongated objects of different widths.

II. BUILDING THE PYRAMID

The edge pyramids used in these experiments were constructed by first building a gray-level pyramid (Fig. 1(a)) and then applying an edge operator at each level of the pyramid to produce a new structure.

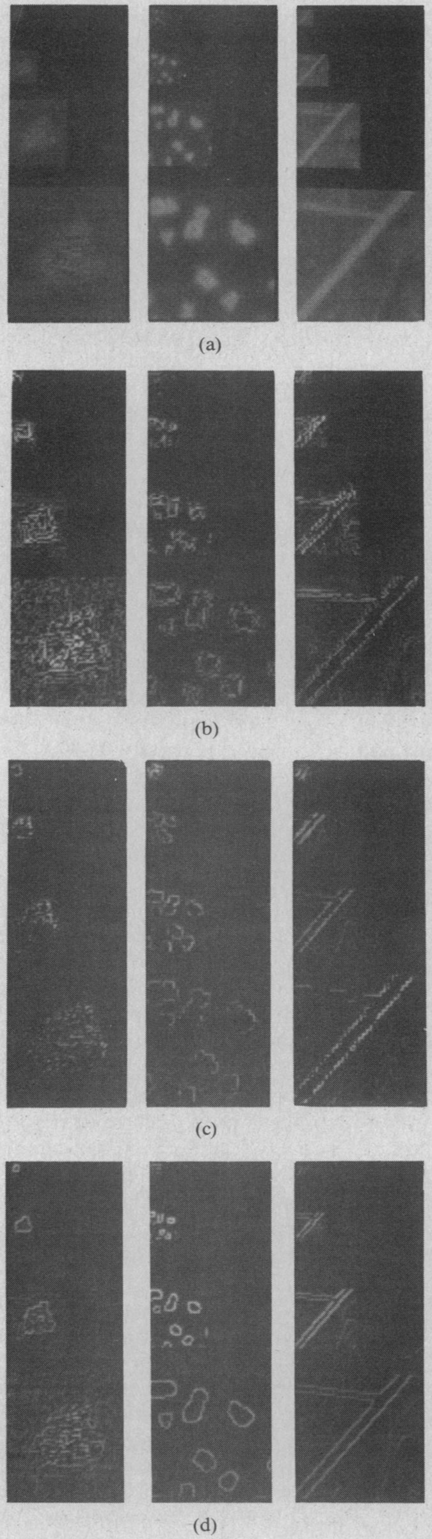


Fig. 1. (a) Gray-level pyramid. (b) Output of Mero-Vamos operator. (c) Output of zero-crossing detector. (d) Output of three-level mask operator.

The gray-scale pyramid was based on nonoverlapped 2 by 2 blocks. In order that the edges should not be blurred and the shapes distorted [11], the values at each level were defined as the medians, rather than the means, of the blocks at the level below.

Manuscript received December 2, 1981; revised April 2, 1982. This work was supported by the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206).

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

0018-9472/82/0900-0660$00.75 ©1982 IEEE

Three edge operators were then used to produce edge pyramids. The main criteria for an operator were that it should not shift the edge too much from one level to the next in the pyramid and that it should maintain direction of the edge across levels where possible. Because of the resampling involved in constructing the gray-level pyramid, there is no guarantee that an edge detector will find an edge in the same relative position at successive levels. The position of the edge is partly a function of the size of the window used in the edge operator. An edge operator discussed by Mero and Vamos [8] allows the calculation of edge position within a window, as well as its magnitude and direction. As Fig. 1(b) shows, although the positions of the edges were maintained well, the operator was discarded because the edges it produced were not smoothly connected; this is a common problem with Hueckel-like operators.

A 5 by 5 zero-crossing algorithm was also tested (Fig. 1(c)) but failed to maintain the positions of the edges accurately enough, perhaps because of the size of the window. The operator that was finally used (Fig. 1(d)) is one that scored highest in the edge evaluation tests of Kitchen and Rosenfeld [6]. This is the three-level template operator [1] which used eight directional masks, e.g.,

$$\begin{array}{rrr} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array} \quad \text{and} \quad \begin{array}{rrr} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{array}.$$

The edge detection was followed by a nonmaximum suppression stage. A 3 by 3 window was placed around each edge point. The direction of the edge was used to find the two edge points to use for nonmaximum suppression. If the edge point had a magnitude greater than both points, it survived; otherwise, it was deleted.

III. EDGE LINKING

For the purposes of linking edges together between the levels, each edge at each level of the pyramid can be assumed to arise from either a 2 by 2 or a 4 by 4 region on the level below. (In effect, when we use a 4 by 4 region, we are treating the pyramid as though it were an overlapped structure.) Note that these mappings are imposed on the structure, rather than resulting from the way the pyramid was built (as they did in [10]).

For an edge detection to be acceptable, the edge at one level in the pyramid must map to a similar edge or edges in the corresponding region of the level below. By restriction of the mapping region to a 2 by 2 neighborhood, a top-down linking process can be defined to verify that the edges are compatible. This process is similar to the "projection" procedure defined by Hanson and Riseman [3] and Uhr [14].

Each edge point at level n has four sons on level $n - 1$. By comparing the edge directions of the four sons with their father, those that are compatible (i.e., differ by a small enough amount) can be linked to the father. Doing this for each adjacent pair of levels results in a tree structure with roots of trees at various levels in the pyramid. The roots of trees at each level can be projected down through their links to the bottom level. The resulting images indicate how well the edge detector maintains information across levels. In each part of Fig. 2, the lower left picture shows all the edges in the image that have nonzero magnitude; the lower right, upper left, and upper right show those edges belonging to trees whose roots are on levels ≥ 1, ≥ 2, and ≥ 3, of the pyramid, respectively.

The linking method that was employed for the experiments, however, used a bottom up technique based on 4 by 4 overlapping neighborhoods [2]. Each son has four potential fathers, and each father has 16 sons. In the bottom-up approach to edge linking, each son compares his direction with those of his four fathers and chooses the father whose direction is most compatible. If the difference in directions is less that some threshold (here: 23°), the son is linked to the father. Otherwise, the son becomes the root of a tree. (Ties are broken by choosing the first father that satisfies the criteria.)

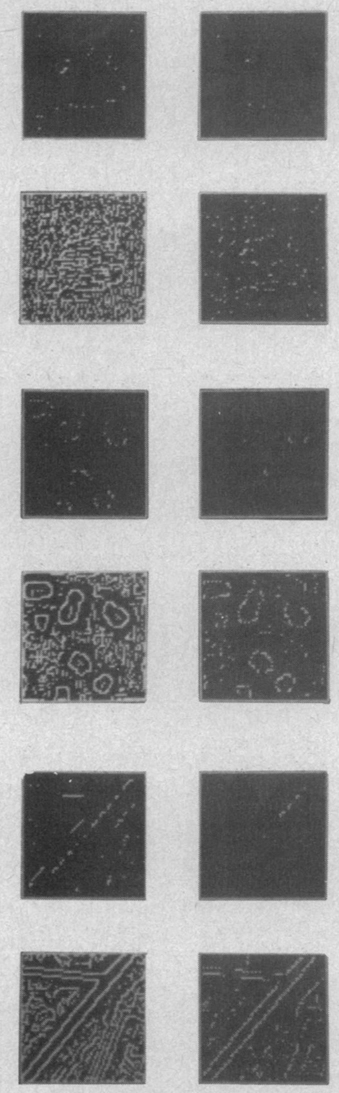


Fig. 2. Results of nonoverlapped pyramid linking. Edge points belonging to trees whose roots are on levels: top left ≥ 2, top right ≥ 3, bottom left ≥ 0, and bottom right ≥ 1.

When the roots at each level are projected down through their trees (Fig. 3), a significant feature of the pyramid becomes apparent: the height in the pyramid at which an object disappears is proportional to the size of the object. This is especially apparent in the chromosome example, and results from the smoothing effect of the pyramid construction process. Thus if objects of a particular known size are sought, the appropriate level in the pyramid can be projected down [5].

It is possible to employ still another linking scheme for constructing edge pyramids. If the 4 by 4 overlapped mapping is used, and the linking is performed top-down, then a father will link to as many of his 16 sons as are compatible with him. The difference between this scheme and the bottom-up scheme is that now a son can be linked to more than one father. The potential of the resulting structure has not been explored.

IV. EDGE COMPATIBILITY

Merely linking the edges in the pyramid and projecting the resulting trees down does not guarantee that the objects that are produced will be closed or compact. Clearly, edges that survive to a level high in the pyramid must correspond to fairly long isolated boundaries. Adjacent edge nodes, however, are not nec-

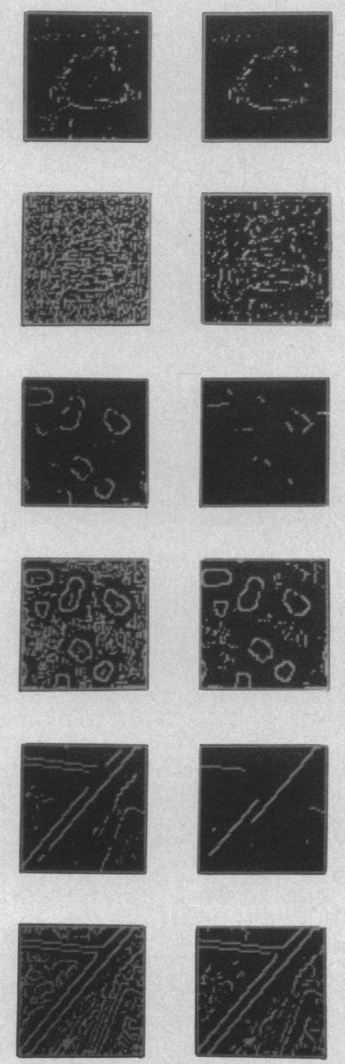


Fig. 3. Results of overlapped pyramid linking. Edge points belonging to trees whose roots are on levels: top left ≥ 2, top right ≥ 3, bottom left ≥ 0, bottom right ≥ 1.

```
                    4
        *       4   4
2               3   4
2   2   1   3   3
2   1   1   1   3
```

Fig. 4. Computation of scores for parallels: position in which the presence of parallels causes the score of the asterisk to be incremented. Directions are measured counterclockwise from positive x-axis.

essarily part of the same object. It would be useful if edges that belonged to the same object could be grouped together into a single tree representing the boundary of the object.

Unless the object has smooth boundaries and is compact, it is unlikely that the whole boundary will be represented at a single level in the pyramid. This is because corners and short edge segments are not as likely to link to their fathers as are long straight segments. As a result an object boundary will be represented by a number of trees, rooted at various levels in the pyramid.

Thus two steps are involved in extracting clean object boundaries. The first involves finding those edges that belong to real objects, and the second involves collecting together the roots that form the complete boundary.

Finding the edges that belong to objects is a two-part process. An edge is a candidate for an object boundary if it has predecessors and successors that are compatible with it and if there is an antiparallel edge across the object from it. A score is calculated for each edge point based on measuring these compatibilities.

Finding antiparallel pairs in the pyramid is a truly bounded operation, in contrast with the situation at a single level [9], [4]. If an antiparallel edge is not found within four pixels at the given level of the pyramid, there is no need to search farther out, because the edges will be closer together at the next higher level. This bound on the search distance is one of the satisfying advantages of the pyramid method.

The first pass of the compactness algorithm computes a score for antiparallel edges and is performed at every level of the pyramid. The process is simplified because the edges have had nonmaxima suppressed. Let x be the direction of the edge at the starred position in Fig. 4. Depending on the direction, some subset of the points in the 5 by 6 neighborhood of the asterisk is examined as follows.

If $202° \leq x < 248°$, then edges in positions marked 2 in Fig. 4 are examined. If any 2 is antiparallel to the asterisk (i.e., differs in direction by $180° \pm 23°$), the asterisk and 2 are incremented by 1. Similarly, if $248° \leq x < 293°$, then edges in position 1 in the figure are used in the same way. If $293° \leq x < 338°$, the edges at position 3 are used, and if $338° \leq x < 359°$ or $0° \leq x < 23°$, then the edges labelled 4 are used.

This process involves a single pass through the image and is followed by a second pass to score edge continuity. A 3 by 3 neighborhood is placed about each edge point, and a set of points

```
*   *           * *        * *          *
* ↑ *         * ↖            ←          * ↙
              *            * *            * *



                    *      * *          * *
* ↓ *           ↘ *          →            ↗ *
*   *         * *          * *              *
```

Fig. 5. Computation of scores for continuations: asterisks indicate neighbors examined when center point has slope indicated by arrow.

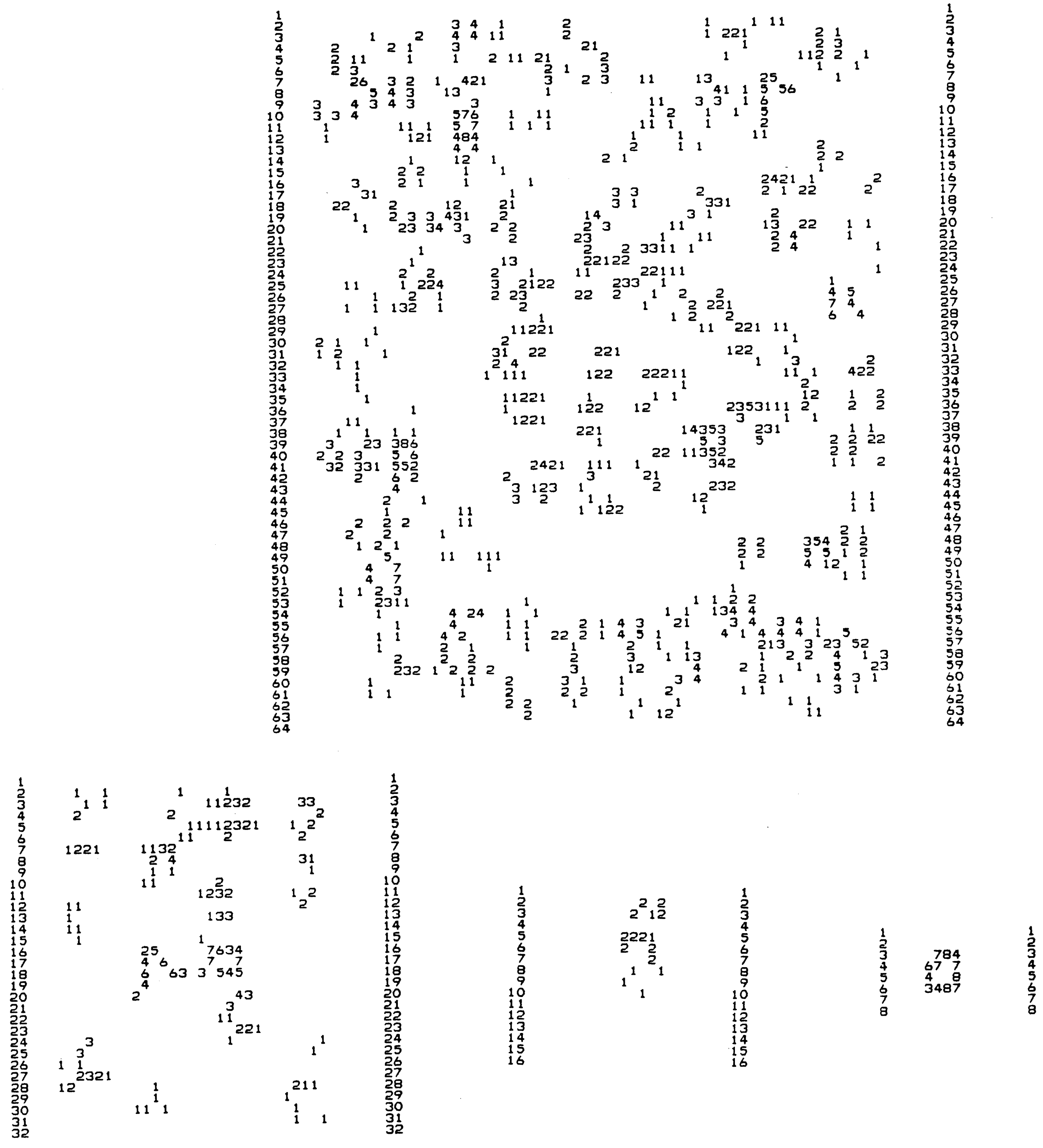


Fig. 6. Compatibility score calculated for each edge point at each level of pyramid for tank image.

within that neighborhood is examined for compatibility. The set depends on the direction of the central edge, and a neighbor is considered consistent if its direction does not differ from that of the central edge by more than 90°. The set of points that is examined for each direction is shown in Fig. 5. For each neighbor that is consistent with the central edge's direction, the score of the central point is increased by one. The scores calculated at each level of the pyramid are shown in Fig. 6 for the picture of the tank.

The scoring is somewhat arbitrary but does not seem to be crucial to the subsequent processing. Other scoring methods for compatibility have been defined by Hong *et al.* [4], Kitchen and Rosenfeld [6], Tavakoli [13], and Scher *et al.* [9]. Whichever is used, the result after the compatibility process is an estimate of how well each edge at each level of the pyramid belongs to a compact object on a smooth boundary. This estimate is used in the next stage of processing, which produces trees corresponding to consistent edge segments.

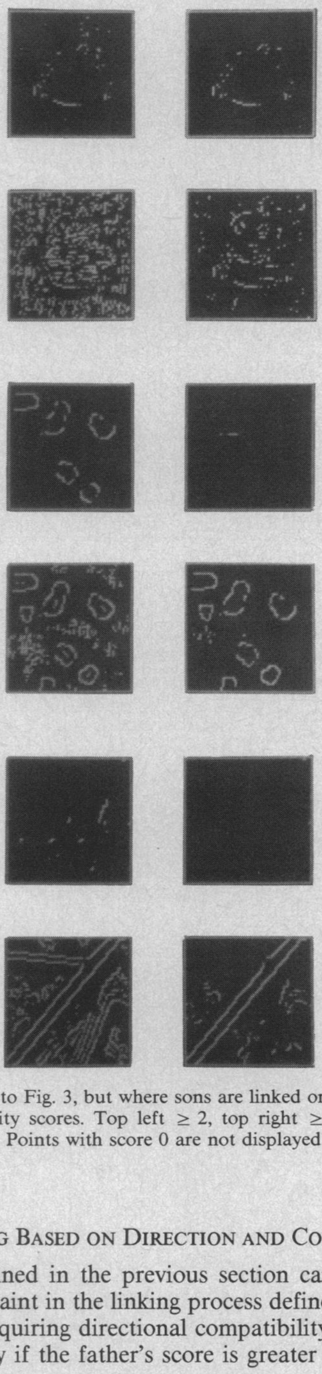


Fig. 7. Analogous to Fig. 3, but where sons are linked only to fathers having higher compatibility scores. Top left ≥ 2, top right ≥ 3, bottom left ≥ 0, bottom right ≥ 1. Points with score 0 are not displayed.

V. LINKING BASED ON DIRECTION AND COMPATIBILITY

The score defined in the previous section can be used as an additional constraint in the linking process defined in Section III. In addition to requiring directional compatibility, a son is linked to his father only if the father's score is greater than or equal to the son's score.

The score is a measure of the compatibility of an edge with other edges at the given pyramid level. Different objects in the scene may be of different sizes, and the size is reflected by the height in the pyramid at which the edges in the object attain their highest score. Consider what happens to a fairly extended closed object. Low down in the pyramid, the distance between antiparallel pairs of edge points is quite large, so that edges only receive compatibility scores based on the good continuation of their neighbors. As the level of the pyramid increases, however, the opposite edges of the object become closer and closer, until they are within the range of the scoring process. At this stage the score for edge points should rise because the correct level has been reached. At higher levels the entire object will merge into a spot or set of spots, and the compatibility will again decrease. Thus by terminating linking when the score attains its maximum, the roots of the resulting trees are directly related to the borders of compact objects.

Note that there is no guarantee that the entire boundary of the object will be extracted at a single level. If there are sharp protuberances or concavities, these will receive high scores lower down in the pyramid and are also less likely to be compatible with the directions of their fathers. An example can be seen in the image of the tank of Fig. 7, where the border is spread over several levels. Fig. 8 shows only the scores associated with nodes that are rooted at each level in the pyramid. When compared with the scores shown in Fig. 6, it can be seen that the roots usually correspond to parts of object boundaries and that most of the noise disappears at low levels in the pyramid.

A variant of this linking scheme involves a son linking to his father if any ancestor (based on direction compatibility) has a score higher than the son's score. This allows edges that score highly because of noise to be linked correctly as part of a larger object. The results of applying this linking method are shown in Fig. 9.

VI. RECONSTRUCTING THE REGION BOUNDARIES

Reconstructing the object boundaries from the linked pyramid involves projecting down certain of the roots in the tree, by expanding all their descendants. The crucial problem lies in deciding which such trees to expand to complete the boundary of an individual object.

If all objects are known to lie within a fixed range of sizes, the corresponding levels in the pyramid can be examined, and all trees rooted at those levels can be expanded. The results of applying this method to the three example images were shown in Figs. 2, 3, 7, and 9.

A more informal method that does not require knowledge of the object sizes is based on the same scoring procedure that was used to define the compatibilities of edges. Recall that an edge received a high score if there were several close neighbors that either continued the edge or were antiparallel to it. Instead of simply counting the number of such neighbors, each edge point can maintain a set of links to other compatible edges. To project the boundary of a single compact object involves projecting all the roots that are linked together, plus, perhaps, some roots lower down in the tree. There are a number of problems to be considered.

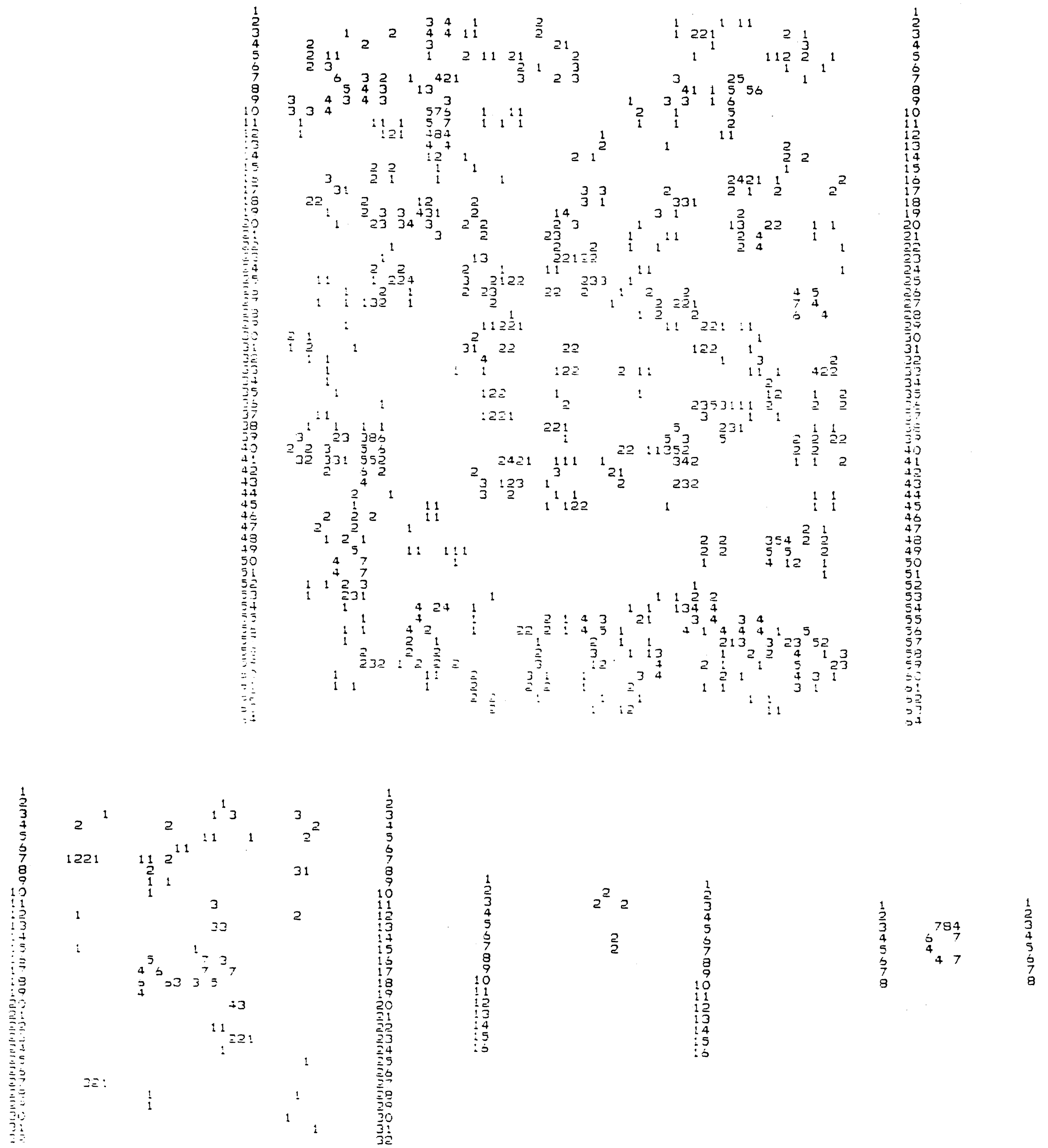
First, consider the top level of the pyramid of the chromosome image in Fig. 8. The only edge consists of a single short straight line with a very low compatibility score. This edge provides no indication of how the complete object to which it belongs might be oriented in the image and does not have links to other roots that are parts of the root of the boundary of the object. Thus it is not clear how to proceed with the edge expansion. The solution is to force the sons of this node (one level down) to be considered roots instead, and to link them, if possible, to consistent neighbors.

The algorithm for expansion, given a sufficiency of compact tree roots, is as follows.

1) Pick any root and flag it to be expanded.
2) Follow the links from the flagged root, and flag the neighbors as well. Repeat this step from each neighbor until no unflagged nodes are reachable. (This can be done in a manner similar to connected component labeling.)
3) Expand all the flagged nodes by tracing their links to the bottom level.

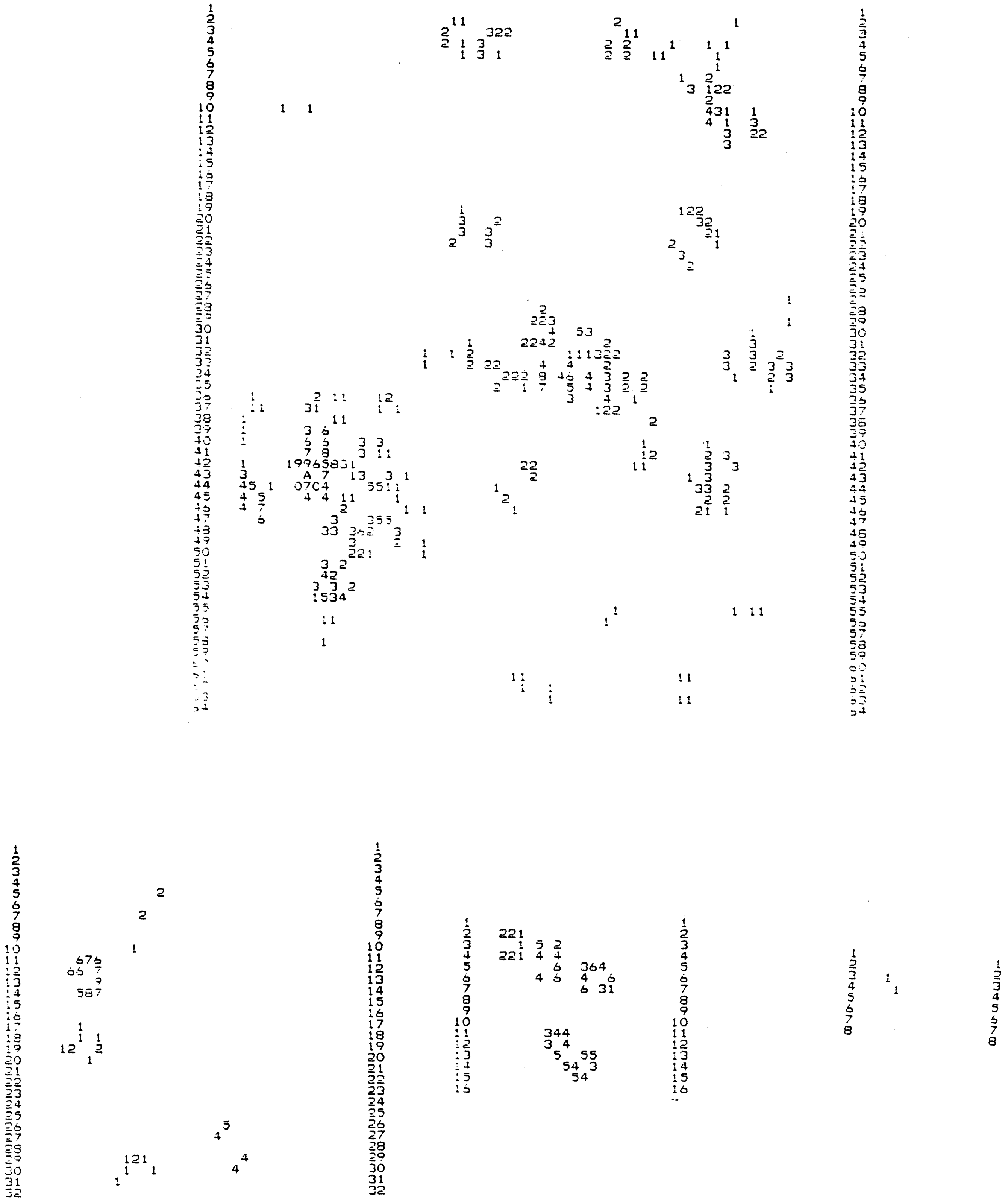
The results of this process are object borders but reflect only the parts of the border that were rooted at a particular level in the tree. Notice, though, that the edges comprising the border are known to belong together. This is in contrast with the simplistic method of taking a union of the roots at each level in the tree.

It still remains to fill in the gaps left either by roots lower down in the tree, or by the failure of the edge detector to find any edges in parts of the image.



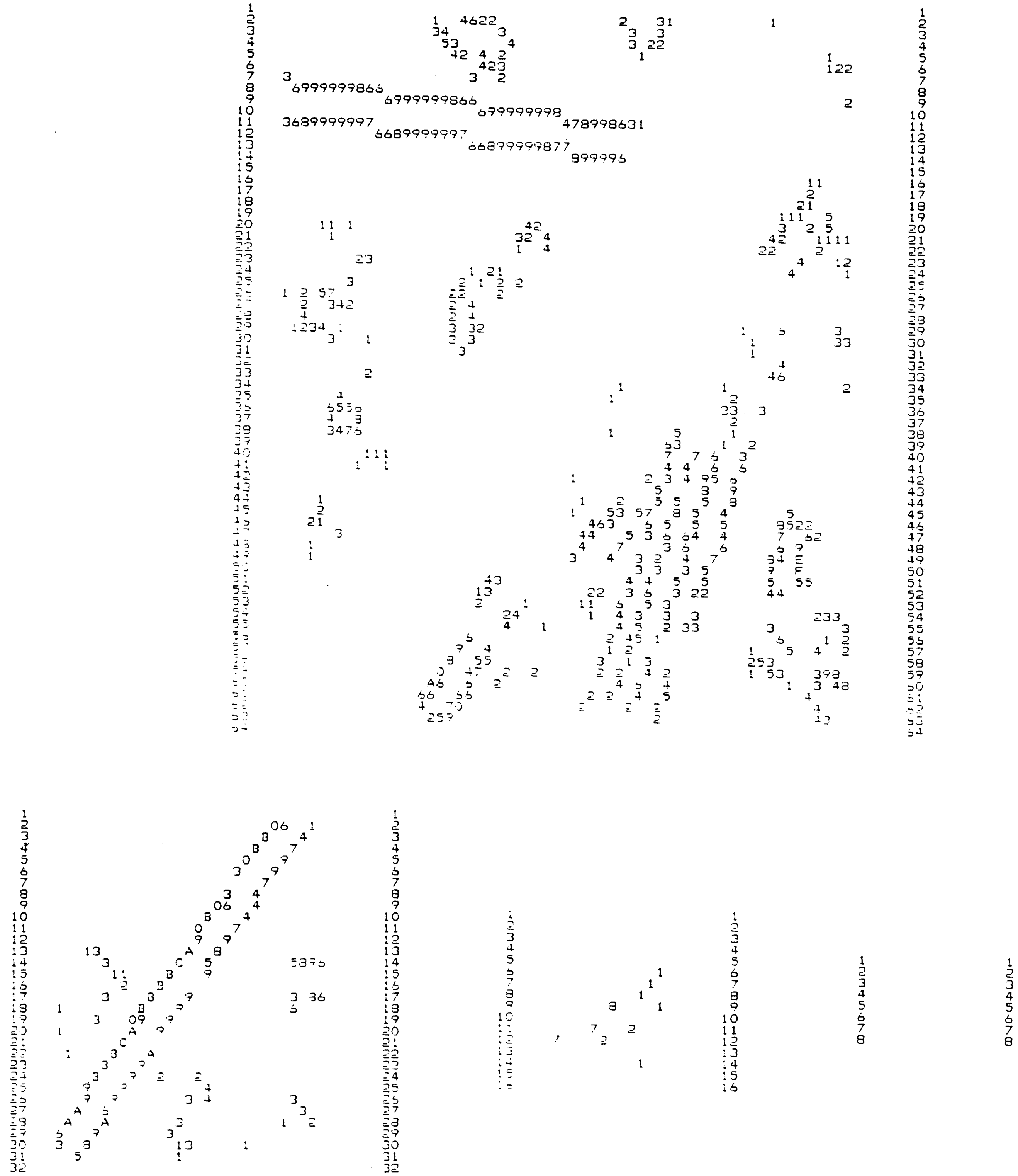
(a)

Fig. 8. Roots of trees at each level of pyramid, using both the direction and compatibility scoring criteria to terminate linking. (a) Tank pyramid. (b) Chromosome pyramid. (c) Runway pyramid.



(b)

Fig. 8. *Continued.*



(c)

Fig. 8. *Continued.*

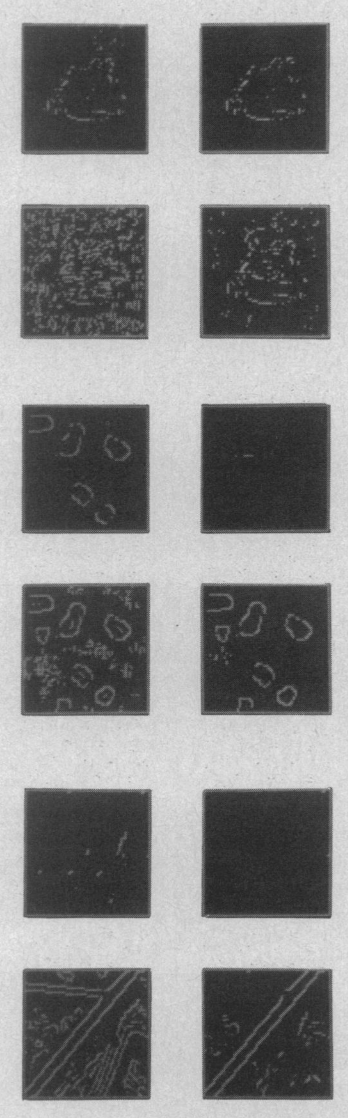


Fig. 9. Analogous to Fig. 3, but where sons are linked to fathers only when some ancestor has a higher compatibility score. Top left ≥ 2, top right ≥ 3, bottom left ≥ 0, bottom right ≥ 1. Points with score 0 are not displayed.

Gaps in the border usually become smaller and smaller as the level in the pyramid increases. Given two roots of subtrees that are not adjacent, there are usually only a few neighboring points that are compatible with the directions of the roots—e.g., the starred points in

* ↙
* *
* *
↑

By expanding these regions down the pyramid, adding in roots that are compatible with existing roots, a constrained search can discover the parts of the boundary that are represented in the pyramid at any level.

At the bottom level there might still be gaps in the boundary. These can be filled in by interpolation based on the positions and directions at the boundaries of the gaps.

VII. CONCLUSION

An edge pyramid has been defined based on applying an edge detector to each level of a gray-scale pyramid. A means of linking edges between levels has been described that takes into account both the compatibility of edge directions in adjacent pyramid levels and a measure of the compactness or compatibility of neighboring edge points in each level.

Reconstructions of edge data from the pyramid are less noisy than applying the edge operator to the original image and retain only the important edges. It is also possible to extract objects of known sizes by noting that those objects are constrained to be rooted at particular levels in the pyramid. More important, it is possible to extract compact object borders based on a local search for compatible points.

ACKNOWLEDGMENT

The help of Janet Salzman in preparing this paper is gratefully acknowledged.

REFERENCES

[1] I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proc. IEEE*, vol. 67, pp. 753–763, May 1979.

[2] P. J. Burt, T. H. Hong, and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, pp. 802–809, Dec. 1981.

[3] A. R. Hanson and E. M. Riseman, "Segmentation of natural scenes," in *Computer Vision Systems*, Hanson and Riseman, Eds. New York: Academic, 1978, pp. 129–163.

[4] T. H. Hong, C. Dyer, and A. Rosenfeld, "Texture primitive extraction using an edge-based approach," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp. 659–675, Oct. 1980.

[5] M. D. Kelly, "Edge detection in pictures by computer using planning," in *Machine Intelligence 6*, Michie and Meltzer, Eds. Edinburgh: Edinburgh Univ., 1971, pp. 379–409.

[6] L. Kitchen and A. Rosenfeld, "Edge evaluation using local edge coherence," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, pp. 597–605, Sept. 1981.

[7] M. D. Levine, "Region analysis using a pyramid data structure," in *Structured Computer Vision*, Tanimoto and Klinger, Eds. New York: Academic, 1980, pp. 57–100.

[8] L. Mero and T. Vamos, "Real-time edge-detection using local operators," *Proc. IJCPR*, Coronado, CA, 1976, pp. 31–34.

[9] A. Scher, M. Shneier, and A. Rosenfeld, "A method for finding pairs of anti-parallel straight lines," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-14, pp. 316–323, 1982.

[10] M. Shneier, "Two hierarchical linear feature representations: edge pyramids and edge quadtrees," *Computer Graphics and Image Processing*, vol. 17, pp. 211–224, 1981.

[11] S. L. Tanimoto, "Pictorial feature distortion in a pyramid," *Computer Graphics and Image Processing*, vol. 5, pp. 330–352, 1976.

[12] S. Tanimoto and A. Klinger, Eds., *Structured Computer Vision: Machine Perception through Hierarchical Computation Structures*. New York: Academic, 1980.

[13] M. Tavakoli, "Toward the recognition of cultural features," in *Proc. DARPA Image Understanding Workshop*, College Park, MD, April 1980, pp. 33–57.

[14] L. Uhr, "'Recognition cones' and some test results: the imminent arrival of well-structured parallel-serial computers; positions, and positions on positions," in *Computer Vision Systems*, Hanson and Riseman, Eds. New York: Academic, 1978, pp. 363–378.

Determination of the Structure of a Canonical Model for the Identification of Linear Multivariable Systems

HOSSNY EL-SHERIEF AND N. K. SINHA

***Abstract*—An algorithm is presented for determining the structural parameters of a canonical state space model from noisy measurements.**

Manuscript received December 10, 1981; revised December 11, 1981 and March 22, 1982.

H. El-Sherief is with the Exxon Production Research Company, P.O. Box 2189, Houston, TX 77001.

N. K. Sinha is with the Electrical Engineering Department, McMaster University, Hamilton, ON, Canada.

0018-9472/82/0900-0668$00.75 ©1982 IEEE