Dynamically Quantized Spaces for Focusing the Hough Transform

Joseph O'Rourke

Electrical Engineering Department
Johns Hopkins University
Baltimore, MD 21218

ABSTRACT

A dynamic data structure is suggested for representing the accumulator array used with a Hough Transform. The structure, called a *Dynamically Quantlsd Space,* allocates resources where they are most needed to best discriminate the peak in the parameter space. The effect is to focus the Hough Transform. Such attention control is especially needed when the parameter space has a large number of dimensions.

1. THE HOUGH TRANSFORM

The Hough Transform [7] has developed into a useful image processing technique. It is applicable whenever some set of image patterns *P* can be described by a finite number of parameters *u1u2, .. . ,uk.* Its purpose is to determine the parameters for the patterns in the image. Each point (u1, . . *,uk)* in parameter space determines a particular pattern *p€P* via a function / : f(u1,, ...uk)— *p.* The Hough Transform consists of making measurements in the image that give evidence for a subset of the patterns *P9CP,* and mapping these patterns into the parameter space with the inverse of / : *f~x{P0*)-

If this process of mapping an image measurement into a set of points in parameter space is repeated, then those points that actually correspond to a pattern present in the image will be mapped to (or "hit" or "voted for* [13)) most frequently. This property forms the basis of implementations of the technique. The parameter space is represented as a k-dimensional accumulator array, each cell of which counts the number of times it has been hit by a mapping from the image. The parameter coordinates of the cell that attains the highest count then corresponds to the pattern that has the most evidential support from the image measurements. Thus the Hough Transform converts a global pattern detection problem into a local detection problem: finding a local maximum in the parameter space [13].

This technique has been successfully applied towards finding straight lines [7]I5I, circles [9], parabolas [17], and arbitrary 2D shapes [13][1] in images, as well as for image registration [17]; many other applications are possible.

2. DYNAMIC QUANTIZATION

The quantization of the parameter space represented by the accumulator array is usually chosen to be uniform: each ceil has the same size. Static non-uniform quantization schemes have been used to ensure that each cell is equally likely to be hit (4][8]. Rut all quantization schemes run into a serious problem: the large amount of storage required for the cells of the accumulator array. This problem becomes especially acute for high-dimensional parameter spaces, * where even a coarse quantization leads to immense storage requirements. It is to help alleviate this problem that *Dynamically Quantized (DQ) Spaces* have been proposed [10][11][12].

Tn a DQ Space, cells may split or merge in response to the local characteristics of the data. Each cell represents a *k*-dimensional rectangular box in the parameter space, and contains a counter that accumulates the number of hits within the bounds of the cell. A cell may be modified in three ways:

M1 : Us counter may be *incremented* by being hit by an inverse mapping from the image.

M2 : It may *spilt* in half across one dimension, creating two new cells.

*M*3: It may *merge* with one or more of its immediate neighbors.

The splitting and merging are guided by two goals:

G : The count in each cell should represent an *equal* portion of the total count over ail cells in the Space.

G2 ■ The hits within each cell should be *uniformly* distributed throughout the volume of the cell.

This last goal requires that each cell have some knowledge of the distribution of hits within its boundaries. This is accomplished by associating a *k*-dimensional *Imbalance* vector (essentially an approximation to the gradient) with each cell, and updating it as points fall in the cell.

The above goals lead to the following split and merge rules.

SPLIT RULE: A cell should be split if its count is too large (relative to the total count), *or* if its internal imbalance becomes too large. Each of the two new cells resulting from the split receive exactly half of the count of the original cell.

MERGE RULE: A cell should be merged with its neighbor if together their counts are not too large, *and* the merge will not produce a highly unbalanced cell. The new cell resulting from the merge is assigned as a count the sum of the counts in the merging participants.

Note that the merge rule is essentially the complement of the split rule. The actual rules in use are considerably more complicated than the above statements indicate; see [10] for details.

3. HIERARCHICAL ORGANIZATION

It is clear from the above description that the cells must be stored in a data structure that (a) allows any cell to be accessed quickly for incrementing its count, and (b) permits the neighbors of any cell (used in the merge rule) to be found easily.

A hierarchical organization immediately suggests itself, and in particular, the k-dimensional generalization of the quad and oct tree [16] seems an obvious candidate for the structure. However, their rigid organization and large number of sons per node (2*) have led me to use instead the *k-d* trees of Bentley [3][6][2]. Sloan is currently investigating the quad/oct tree alternative, which he calls a Dynamically Quantized (DQ) Pyramid [15]

* The highest dimensional space I hive seen in the literature is the 4-dtmensiorml space used in [13][1].

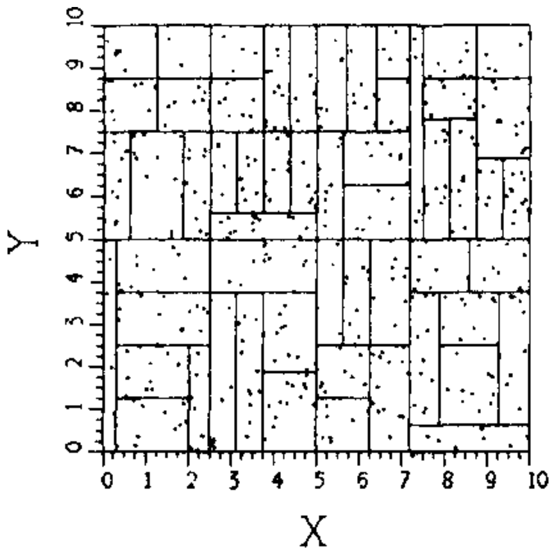

Figure 1. 500 points with little central tendency.

Each node of a *k-d* tree represent! a k-dimensional rectangular box. Every node has exactly two sons, which partition it into two smaller pieces. The index of the dimension across which the box is split is stored at the node, along with an indication of how far along the splitting dimension the cut occurs (it does not have to be at the midpoint).

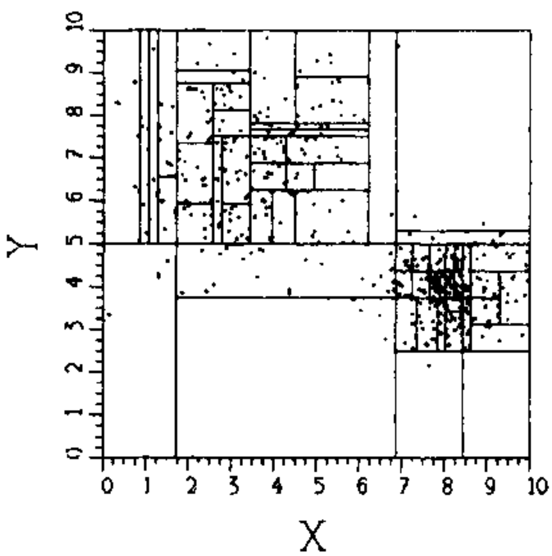

Figure 2. 500 points drawn from two distributions: one centered at (3,7), and the other (with a smaller variance) centered at (8,4).

4. PROPERTIES OF DQ SPACES

In the current implementation, points are "plotted* in parameter space by (1) accessing the appropriate cell via a path from the root, (2) incrementing the cell's counter, and (3) firing the split or merge rule for the cell if applicable, and checking the rules for any neighboring cells modified by a rule firing, and so on recursively. The effect is that a DQ Space will dynamically reorganize itself to match the local characteristics of the data. This scheme possesses the following properties:

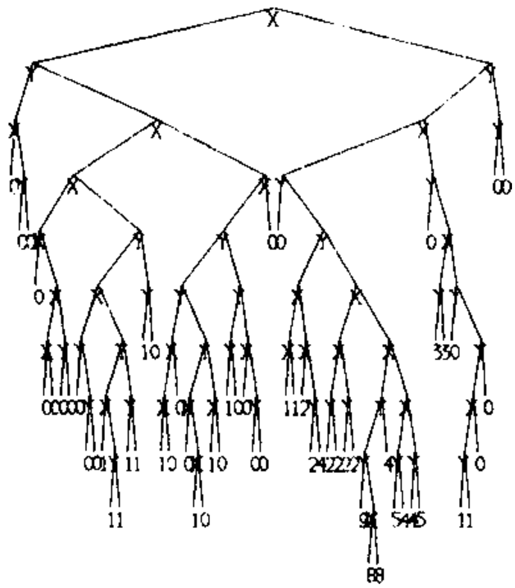

Figure 3. The tree structure of the DQ Space of Figure 2. Relative count densities are shown at the leaves.

Focus Capability: Figure I shows a 2D DQ Space containing 500 points that do not display any marked central tendency; the resulting cells are roughly the same size. The 500 points in Figure 2, however, cluster around the two points (3,7) and (8,4), and the cell sizes adjusted to focus in on these clusters. The *k-d* tree supporting the cells of Figure 2 is depicted in Figure 3. The splitting dimension is indicated at each node, and a single digit proportional to the density of counts in each cell is shown at the associated terminal node.

Resolution Control: The resolution of the Space can easily be controlled by parameters that limit the number of cells. The DQ Space tries to achieve its goals within the limits set, and "gracefully degrades* if the resources allocated are inadequate to the task.

Dimensionality Reduction: If the data is uniformly distributed along some dimension, then the DQ Space will tend not to discriminate (i.e., split) along that dimension (see Figure 4). In effect, the dimensionality of the space is reduced. This could be especially important in a high-dimensional space that has some inessential dimensions.

Change of Attention: In the current implementation, the count in each cell "fades" with time, so that a moving peak in parameter space can be "tracked" by the focusing capability of the DQ Space.

5. CURRENT AND FUTURE WORK

DQ Spaces are currently being used for motion detection and analysis (10][II][12] Some success has been achieved with a variation of Ullman's Two Cylinder Experiment* [19] using a 3D parameter space [12].

The complexity of DQ Spaces has rendered theoretical analysis difficult. * Currently the interrelationships between the split

See [10] for the meaner results that have been obtained.

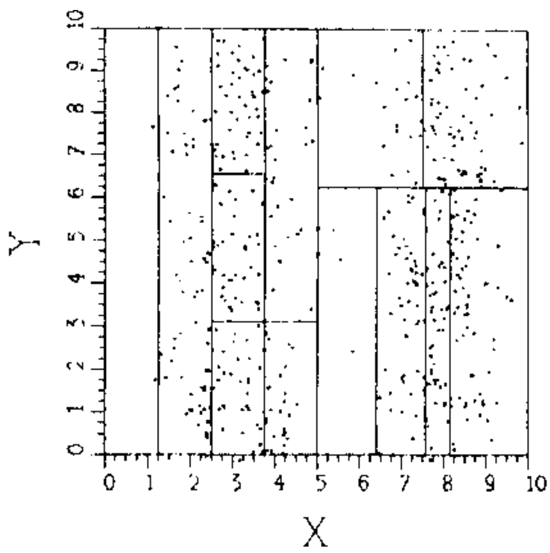

Figure 4. The data here is spread uniformly along the *y* dimension, and the DQ Space discriminates mainly across the *x* dimension.

and merge tolerances and the resolution parameters, which together control the behavior of the space, are unclear. One advantage of Sloan's DQ Pyramids is their simplicity relative to DQ Spaces.

The price for this simplicity is inflexibility: DQ Pyramids always allocate equal system resources for each dimension of the parameter space, whereas DQ Spaces can ignore irrelevant dimensions. In fact, it is possible for a DQ Space to have different dimensions in different sections of the space, constituting a sort of patchwork quilt of different parameter spaces. This corresponds well with our intuition that a change of context may engender a change of relevant features.

REFERENCES

[1] Ballard, D. R, "Generalizing the Hough Transform to Detect Arbitrary Shapes,* *Pattern Recognition,* Vol. 13, No. 2, 1981, pp. 111-122.

[2] Bentley, J. L., "Multidimensional Binary Search Trees Used for Associative Searching," *CACM,* Vol. 18, No. 9, Sept. 1975, pp. 509-517.

[3] Bentley, J. L., "Multidimensional Binary Search Trees in Daia Base Applications," *IEEE Trans, on Software Engineering,* Vol. SE-5, No. 4. July 1979, pp. 333-340.

[4] Cohen, M. and Toussaint, G. T., "On the Detection of Structures in Noisy Pictures," *Pattern Recognition,* Vol. 9, 1977, pp. 95-98.

[51 Duda, R. O. and Hart, P. E. "Use of the Hough Transform to Detect Lines and Curves in Pictures," *CACM,* Vol. 15, Jan. 1972, pp. 11-15.

[61 Friedman, J. R, Bendey, J. L., and Finkel, R. A., "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Trans, on Mathematical Software,* Vol. 3, No. 3, Sept. 1977, pp. 209-226.

[7] Hough, P. V. C, "Methods and Means for Recognizing Complex Patterns," U.S. Patent 3,069,654, 1962.

[8] Iannino, A., "Hough Transform Theory and Image Processing Experiments," PhD Dissertation, Stevens Institute of Technology, 1979.

[9] Kimme, C, Ballard, D. R, and Sklansky, J., "Finding Circles by an Array of Accumulators," *CACM,* Vol. 18, No. 1, Feb. 1975, pp. 120-122.

[10] O'Rourke, J., "Dynamically Quantized Spaces Applied to Motion Analysis," Johns Hopkins University Technical Report JHU-EE 81-1, Jan. 1981.

[11] O'Rourke, J., "Automatic Visual Attention Control via a Dynamic Data Structure," Proceedings of the Conference on Information Sciences and Systems, Johns Hopkins University, Baltimore, Maryland, Mar. 1981, pp. 301-306.

[12] O'Rourke, J., "Motion Detection using Hough Techniques," Proceedings of the IEEE Conference on Pattern Recognition and Image Processing, Dallas, Texas, Aug. 1981.

[13] Roscnfeld, A. and Kak, A. C, *Digital Picture Processing,* Second Edition, Academic Press: New York, 1980, p. 275.

[14] Sloan, K. R , Jr., and Ballard, D. R, "Experience with the Generalized Hough Transform," 5th International Conference on Pattern Recognition Proceedings, Miami Beach, Florida, Dec. 1980, pp. 174-179.

[15] Sloan, K. R., Jr., "Dynamically Quantized Pyramids," this Proceedings, Aug. 1981.

[16] Srihari, S. N., "Hierarchical Representations for Serial Section Images," 5th International Conference on Pattern Recognition Proceedings, Miami Beach, Florida, Dec. 1980, pp. 1075-1080.

[17] Stockman, G. and Kopstein, S., "Image Registration from Edge Content," Eighth Annual Automatic Imagery Pattern Recognition Symposium, R. A. Kirsch and R. N. Nagle, Eds., Electronic Industries Associates, Apr. 1978, pp. 139-157.

[18] Wechsler, R and Sklansky, J., "Automatic Detection of Ribs in Chest Radiographs," *Pattern Recognition,* Vol. 9, Jan. 1977, pp. 21-30.

[19] Ullman, S., *The Interpretation of Visual Motion,* Cambridge, Massachusetts, MIT Press, 1979.