Behavior Research Methods, Instruments, & Computers
1986, 18 (6), 595-600

THEORETICAL TECHNIQUES

Pyramid algorithms for perceptual organization

AZRIEL ROSENFELD
University of Maryland, College Park, Maryland

Multiresolution (or *pyramid*) approaches to computer vision provide the capability of rapidly detecting and extracting global structures (features, regions, patterns, etc.) from an image. The human visual system also is able to spontaneously (or *preattentively*) perceive various types of global structure in visual input; this process is sometimes called *perceptual organization.* This paper describes a set of pyramid-based algorithms that can detect and extract these types of structure; included are algorithms for inferring three-dimensional information from images and for processing time sequences of images. If implemented in parallel on cellular pyramid hardware, these algorithms require processing times on the order of the logarithm of the image diameter.

During the past few years, there has been increasing interest in the use of multiresolution (*pyramid*) image representations in image analysis and computer vision. Several research groups have designed or built image-processing machines based on this approach. A pyramid-structured array of multiprocessors can perform many types of operations on an image (input to the base of the pyramid, one pixel per processor), in time proportional to the log of the image diameter. (For a recent collection of papers on pyramid methods in image processing and analysis, see Rosenfeld, 1984.)

One of the most important potential applications of pyramid machines is the fast detection and extraction of global structures (e.g., features, regions, patterns) in an image, by rapidly combining information collected from many parts of the image. A number of pyramid-based methods of extracting global image structures have been developed at the Center for Automation Research, and others have been proposed, as described below.

The rapid detection of global structure in images is an important, but not well-understood, capability of the human visual system. Humans tend to spontaneously (or *preattentively*) perceive various types of global structures in their visual input; this process is sometimes called *perceptual organization.* About 50 years ago, the Gestalt psychologists formulated a set of principles, or laws, that describe how image parts tend to group into global structures (Wertheimer, 1958). These include the laws of similarity, proximity, good continuation, and closure. In this paper, I will show how each of these types of groupings can be detected using fast pyramid algorithms.

Fast detection of global structure in an image seems to be an essential component of real-time perception. Humans are able to recognize objects in unexpected, complex images in 1 sec or less, a period of time during which only on the order of 100 neural computational steps could have taken place. Obviously, the human visual system is highly parallel, but conventional parallel processing concepts are not powerful enough to account for this performance. For example, a human can immediately detect a long straight line in the visual field; since the line may be several hundred retinal receptor cells (*pixels*) long, conventional methods would require several hundred computational steps, even on a two-dimensional parallel machine, to extract the line from the input image. Pyramid-style parallelism provides a much faster method of doing this. As shown below, a pyramid scheme can extract long straight lines from an image in tens, rather than hundreds, of steps. Schemes of this type seem to be essential in achieving fast recognition of global patterns. Such patterns cannot be reliably recognized using conjunctions of local features; some means of explicitly extracting global structures is needed. The techniques described in this paper provide such a means.

The grouping algorithms discussed here are primarily two-dimensional, but similar pyramid-based techniques can be applied to the analysis of images of three-dimensional scenes, or to time-varying images. These extensions will be briefly discussed at the end of the paper.

LAW OF SIMILARITY: DETECTING BIMODALITY

According to the Gestalt law of similarity, if an image consists of a mixture of two different types of local patterns, the human visual system can group the patterns of one type together and perceive them as a figure, while perceiving the remaining patterns to be part of the background. On the other hand, a mixture of three types of local patterns is hard to distinguish from a uniform dis-

Requests for reprints should be mailed to: Azriel Rosenfeld, Center for Automation Research, University of Maryland, College Park, MD 20742.

tribution of pattern types (Marr, 1982, p. 92). Apparently, the human visual system can detect bimodality (of a local property such as size or slope) and can segment an image containing a bimodal population of local patterns (in the sense that the two subpopulations can be perceived as distinct from one another); however, the human visual system cannot easily handle multimodal distributions, and tends to treat them as though they were uniform.

In a computer vision system, it is easy to compute the histogram of values of a local image property and to detect multiple peaks in the histogram (provided they do not overlap). However, constructing a histogram is a somewhat slow process, because each of the distinct values must be separately counted. (A simple pyramid algorithm can be used to count the number of occurrences of any one value in time proportional to the log of the image diameter; however, in order to compute a complete k-valued histogram, this algorithm must be repeated k times.) The pyramid-based technique described here directly detects bimodality, without computing a histogram. It is possible that a scheme of this type could serve as a model for humans' bimodality detection ability.

This scheme is based on a simple pyramid structure in which each cell at level h receives inputs from a block of cells (its *children*) at level $h-1$. The image is initially input to the base of the pyramid (level 0), and the given local property is computed, wherever applicable, yielding a (sparse) array of local property values.

Each cell at level 1 now examines the set of values in its block of the level-0 array, and finds the partition of these values into two subsets that minimizes the variance of each subset about its mean. (This partition is sometimes called the *bimean* [Dunn, Janos, & Rosenfeld, 1983].) Let the means of the subsets be μ and ν, let the sizes of the subsets be r and s, and let the standard deviations be σ and τ. The cell then computes the *Fisher distance*

$$d \equiv |\mu-\nu| / \sqrt{\sigma^2+\tau^2}$$

between the two subsets. A larger Fisher distance means that the population of values in the cell's block is strongly bimodal (the means are many standard deviations apart). If the Fisher distance is not sufficiently large, the cell flags itself as *not bimodal*.

Each cell on level 2 now examines the data computed by its children on level 1. Suppose that all, or nearly all, of the children have bimodal populations, say with means and standard deviations $\mu_1, \nu_1, \ldots, \mu_m, \nu_m$ and $\sigma_1, \tau_1, \ldots, \sigma_m, \tau_m$. The cell finds the partition of the μs and νs into two subsets that minimizes the variance of each subset about its mean. (The means of the subsets can be computed as weighted averages from the μs and νs and their corresponding rs and ss; the variances can be computed from the μs, νs, rs, ss, σs, and τs.)

Finally, the cell estimates the Fisher distance for this new partition and decides whether its combined population of values is still bimodal. If several of the cell's children do not have bimodal populations, the cell does not carry out these computations, but simply flags itself as *not bimodal*.

This process is repeated for the cells on levels 3, 4, ... I suggest that bimodality is *perceptually conspicuous* if a cell on a sufficiently high level (representing a large block of the image) has a bimodal population.

Detection of anomalies can be regarded as a special case of bimodality detection. If there is a unique value that differs considerably from all the other values, it defines a subpopulation that has high Fisher distance from the rest of the population, so that we have strong bimodality.

In principle, this scheme could be generalized to detect trimodality, ..., k-modality, ..., but the amount of computation that the cells would need to perform would grow rapidly with k. In any case, since this method is based on repeatedly estimating the variances of the subsets, it is not capable of fine discriminations between subsets whose means are close together; thus, the method will detect bimodality only when the means are very far apart or when the variances are very small, and there would be little value in attempting to detect k-modality for $k > 2$. (In general, it would be of little value for the brain to perform computations whose results would be so inaccurate as to provide no useful information. If the brain contains pyramid-like structures, they would not be perfectly regular, i.e., the image blocks seen by the cells would neither be of equal sizes nor have equal degrees of overlap. Thus, it would be pointless to attempt to use them, e.g., for exact counting, but reasonable to use them only for rough population size estimation.)

LAW OF PROXIMITY: DETECTING COMPACT REGIONS

The Gestalt law of proximity states that local patterns that lie close together tend to be grouped together as a *figure*. For example, a dense cluster of dots on a sparsely dotted background is perceived as a figure. In this section, perception of groupings that are compact, or blob-like, is discussed. The detection of elongated, or ribbon-like, figures (or parts of figures) probably involves other processes, in addition to proximity-based grouping. This perception will be discussed later in the paper.

Several pyramid-based methods of detecting compact figures have been developed at our laboratory (Hong & Rosenfeld, 1984; Hong & Shneier, 1984). A recently developed, simpler method (Gross, 1986) seems to yield even better results: it is described below. As in the preceding section, assume that the image is input to the base of the pyramid and that the local property is first computed, so that the input becomes an array (not necessarily sparse) of local property values (e.g., gray levels).

Each cell on level 1 computes the mean and variance of the values in its block, and this process is repeated at levels 2, 3, ... (In this case, the variance of a cell at any level can be computed directly from the means and variances of its children, because the children's population sizes are all equal [Burt, 1979].) Let P be a cell on

level h, and let $Q_1, \ldots, Q_k$ be the cells on level $h+1$ whose blocks of the image contain P's block. Let σ be the variance of P, and $\tau_1, \ldots, \tau_k$ be the variances of the Qs; let the area of P's block be s, and the areas of the Qs' blocks be t. We call P a *root node* if $\tau_1/t, \ldots, \tau_k/t$ are all significantly higher than σ/s. Intuitively, this will happen if P's block lies (mostly) within a *homogeneous* region of the image (e.g., a region whose values are independent samples of the same normal distribution), but the Qs' blocks are too big to lie inside that region. (For low levels h, if the image is noisy, this criterion will not be reliable, because the τs may be too variable to be good estimates of the variability of the region's population of values; but for larger values of h, the estimates will be more reliable, and τ_i/t will be smaller than σ/s [since t is larger than s] if Q_i's block lies inside the region. Thus, for large hs, the *root nodes* will represent maximal blocks of the image that approximately coincide with homogeneous regions.)[1]

The homogeneous regions detected in this way can be extracted from the image by a top-down tree-growing process such as the following: Take the root node P as the root of the tree. Let P' be a cell on some level $i < h$, and let $Q'_1, \ldots, Q'_k$ be the cells on level $i+1$ whose image blocks overlap that of P'. Let Q' be that one of the Qs having smallest Fisher distance from P'. Then make P' a tree node if and only if Q' is a tree node. This process is carried out for $i = h-1, h-2, \ldots$. The leaves of the tree at level 0 are the pixels constituting the region. (For details on several variations of this tree-growing procedure, see Gross, 1986.)

The region extracted from the image in this way is necessarily compact, since it cannot contain pixels that are far away from the original root node's block of the image. (Indeed, at successive stages of the tree-growing process, one can only add to the tree cells whose blocks overlap that of P and then to cells whose blocks overlap these blocks, and so on, where the sizes of the blocks are exponentially decreasing.) Of course, parts of P's block can be lost, since some blocks that overlap P's block will not meet the criterion for inclusion in the tree. At the levels immediately below h, it is likely that the cells whose blocks are contained in P's block will nearly all join the tree, since they represent relatively large samples of the same image region; thus, the union of these blocks will define a solid region. At still lower levels, the blocks may no longer resemble that of P, due to noise, but if they are near the center of P's block they will have no choice but to join the tree, so that the region will remain solid. Blocks near the edges of the region, however, will not be forced to join the tree, so the tree-growing process will be able to closely approximate the shape of the region.

Preliminary experiments (Gross, 1986) indicate that this approach is quite effective at extracting compact homogeneous regions from an image, even if they have irregular shapes. In our laboratory, we plan to conduct a more extensive series of experiments in which the approach is applied to the detection and extraction of various types of homogeneously textured regions (i.e., regions having stationary distributions of the values of local properties other than gray level). In a textured image, this method will be able to detect both *texture primitives* (i.e., pieces of the image having nearly constant gray levels) and *textured regions* (i.e., clusters of these pieces); the former should give rise to root nodes at low levels, and the latter to root nodes at higher levels. This approach can also be generalized to nonstationary distributions; for example, we can extract regions over which the distribution of property values varies linearly, by least squares fitting linear functions to the sets of property values in each image block, and calling P a root node if the fits to the Qs are all worse than P's fit. (Recall that when the mean and standard deviation of a block are computed, we are least squares fitting a constant function to the values in the block.)

A related approach can be used to extract compact regions that are smoothly varying, but not necessarily homogeneous (i.e., their spatial distribution of property values does not globally fit a linear function). To do this, P is called a root node if its pairs of adjacent children (i.e., cells on level $h-1$ whose blocks are contained in P's block) all have small Fisher distances (i.e., P contains no edges) and if none of the Qs (on level $h+1$) whose blocks contain Ps have this property. (If the values in P's block vary smoothly, two adjacent children of P cannot have a large Fisher distance; their variances and the difference between their means will both be large if the variation is steep and small if it is gradual.) Then tree-growing methods similar to those described above can be used to extract the detected region.

LAW OF GOOD CONTINUATION: DETECTING SMOOTH CURVES

Approaches analogous to those in the preceding section can be used to detect and extract curves that are smooth or that are good global fits to given functions (e.g., to straight lines). A local edge or curve detection operation is first applied to the image, and the resulting set of feature points is input to the base of the pyramid.

To detect smooth curves, each cell at level 1 examines its block of the image; if many of the feature points in the block lie on a smooth curve, the cell records the positions and slopes of the endpoints of the curve and stores pointers to the feature points that constitute the curve. Each cell at level 2 then examines the data provided by its children at level 1. If many of the endpoint data are consistent (i.e., they can be arranged in a sequence such that successive endpoints closely agree in position and slope),[2] the level-2 cell records the positions and slopes of the first and last endpoints and stores pointers to the level-1 cells that contributed the consistent data. This process is repeated at levels 3, 4, ... Thus, if the image contains a long, smooth curve, some high-level cell will

contain its endpoint data and will also be the root of a tree of pointers whose leaves are the feature points constituting the curve.

This approach requires only a bounded amount of computation by the cells at each level, and thus can be carried out in O(log image diameter) time, even if the curve is long. There are difficulties, however, if the image contains several long, smooth curves, or a single curve that doubles back on itself. In such cases, some of the cells will detect more than one smooth curve and will need the capacity to store several sets of endpoint data. In an early implementation of this approach, the cells were given the ability to store up to five sets of data (Hong, Shneier, Hartley, & Rosenfeld, 1983). In any case, when a cell's capacity is exceeded, the cell no longer stores complete information about all the curves it sees; it stores only statistics (including information about bimodality, anomalies, etc.).

In general, for each stored curve (or for the ensemble of curves, if there are too many to be stored individually), the cell should record not only endpoint data, but also various global properties, such as average gray level (or the averages of the adjacent gray levels on each side, in the case of an edge), arc length, *wiggliness* (i.e., total absolute curvature), and so forth. These properties can be estimated by combining property values obtained from the cells on the level below. It would also be useful to fit straight lines (or possibly polynomials of degree higher than 1) to the curves; this, too, can be done recursively, based on the fits computed on the level below. As in the previous section, the fit error measure should be divided by the number of points being fitted. This reflects the intuitive fact that the fit to a long piece of wiggly straight line is better than the fit to a short piece.

This recursive fitting process can also be used to detect corners, or angles, on a curve (Hartley & Rosenfeld, 1985). A cell detects an angle if two of its children contain good straight line fits that have different slopes and that approximately meet at a common endpoint. The closer the fits, the smaller the slope difference need be for an angle to be detected.

LAW OF CLOSURE: DETECTING BLOBS AND RIBBONS

Another pyramid-based approach to extracting regions from an image is based on detecting parts of the image that are surrounded by edges or curves. This approach can be used to extract ribbon-like as well as blob-like (i.e., compact) regions.

One begins by detecting edges or curves in the input image and inputting the resulting set of feature points to the base of the pyramid. One then passes this information up through the pyramid, condensing and summarizing it at each stage (e.g., by straight line fitting, as in the preceding section). The cells at each level also examine their neighboring cells (e.g., in a 5×5 neighborhood) and check whether they are locally (nearly) surrounded by lines that run approximately broadside to them, and in the case of edge data, that have consistent contrasts (i.e., their dark sides all face inward or all face outward). If a cell discovers that these conditions are satisfied, it has detected a *blob*.

An early implementation of this blob-detection scheme is described by Hong and Shneier (1984). A nonpyramid implementation, which detected blobs by searching out from each pixel in a set of directions, up to a distance equal to the maximum expected blob radius, is described by Minor and Sklansky (1981). The pyramid approach has the advantage that it requires only local searches; blobs of any size will become locally detectable at some level of the pyramid.

The edge-based approach to blob detection may be more perceptually plausible than the function fitting scheme described earlier. In the absence of edges, a region may look uniformly bright even when its brightness varies substantially from one side to another, due to shading. This suggests that the visual system cannot easily determine the slope, for example, of a linear fit to the brightness data. Conversely, suppose a region is surrounded by *pseudoedges* at which there is a local brightness change, but the change smooths out away from the edges in either direction, so that the regions inside and outside the edges actually have the same uniform brightness, except near the edges. In this situation, the region on the dark side of the edges looks uniformly darker than the region on the bright side, even though the brightnesses of the regions are actually the same. This phenomenon is known as the Craik-O'Brien-Cornsweet illusion (Cornsweet, 1970). It suggests that the visual system assigns a brightness to a region based on the local brightnesses along the edges surrounding the region, rather than by fitting a function globally to all of the gray levels in the region.

Hong and Shneier (1984) described a relatively complicated scheme for extracting and "coloring in" a blob detected using the edge-based approach. Another way of defining the precise boundaries of the blob is to project its surrounding edges downward through the pyramid, one level at a time, and at each level to locally adjust them so as to maximize the gray level gradient magnitude along the edges. However, if this simple method of edge adjustment is used, noise at the lower levels of the pyramid will cause the edges to break up, since they are not constrained to be continuous (Rosenfeld, Thurston, & Lee, 1972). Better results are obtained if one starts with a connected region (the union of the pixels surrounded by the edges), projects it downward through the pyramid, and at each step, adjusts its border by adjoining or deleting pixels in such a way as to maximize the contrast around the border. This method constrains the border to remain continuous and yields good region delineations, even for noisy images (Baugher & Rosenfeld, 1986).

An edge- or curve-based pyramid technique can also be used to extract ribbon-like regions from an image. Here, too, one begins by inputting the feature (edge or curve) points to the base of the pyramid, and by perform-

ing repeated, for example, straight line fitting at successively higher levels. The cells at each level also examine their neighborhoods (e.g., 5×5) and look for antisymmetric pairs of edges located on opposite sides of them (i.e., pairs of edges such that the given cell is on the dark side of both edges, or on the light side of both). In effect, this process detects *smoothed local symmetries* (Brady & Asada, 1984) at each level of the pyramid. When a cell detects such a configuration, it stores the estimated position and orientation of the local axis of symmetry, as well as the distance and angle between the pair of edges. The latter information defines a linear fit to the *width function* defined by the pair of edges. Cells on higher levels can then link pieces of symmetry axis into smooth curves along which the width function varies smoothly, by a straightforward generalization of the methods described in the preceding section. (Global functions, such as straight lines, can also be fitted to the axes and to the width function, if desired.) Such a smooth curve defines a *generalized ribbon*. By storing links between the axis segments and their associated edge segments, the edges of the ribbon can be found and projected downward through the pyramid so that the ribbon can be extracted from the image, using the methods described in the previous paragraph.

EXTENSIONS TO THREE-DIMENSIONAL AND TIME-VARYING SCENES

In the preceding sections, I have, for simplicity, treated the image data two-dimensionally, without reference to the fact that the underlying scene may be three-dimensional or that the image may be part of a time sequence. However, the approach can also be applied to infer three-dimensional scene information from an image and to analyze optical flow in a sequence of images.

Global nonlinear variations in the gray level of a region may be an indication that the region is the image of a curved surface. Some work has been done using global function fitting to the image gray levels to detect specific types of surfaces in a scene (Bolle & Cooper, 1984). This type of computation could be efficiently implemented using pyramid techniques, such as those described earlier. Similarly, global variations (linear or nonlinear) in the sizes and spacings of texture elements over a region can serve as an indication that the region is the image of a slanted or curved surface. Thus, the function fitting methods described earlier in this paper can serve as aids in the computation of surface shape from shading or texture, or from both.

The function fitting methods can be applied to vector-valued as well as to scalar-valued data; for example, they can be applied to color data. If one is given a time sequence of images, and can estimate a set of displacement vectors relating points in one image to those in the next, one can also apply the function fitting approach to detect global patterns in the resulting (possibly sparse) *optical flow field*. This approach has been successfully used to segment noisy synthetic optical flow fields into regions representing different types of motions, including translation, planar rotation, and scale change (Hartley, 1985). It should also be applicable to real data. The global grouping of pixels having common velocity vectors was called the law of common fate by the Gestaltists.

It should be pointed out, in conclusion, that the pyramid techniques described in this paper are quite different from the ways in which pyramids have been used by other investigators (see, e.g., Rosenfeld, 1984). Pyramids are often used to generate a set of bandpass-filtered, sampled versions of an image. At the Center for Automation Research, the use of pyramids is quite different; they are employed for model *fitting* rather than for *filtering*.

SUMMARY

The concepts outlined in this paper constitute a basic contribution to the methodology of vision systems design. This approach makes use of the pyramid cellular architecture to rapidly compute global information about an image in a recursive fashion, but its main contribution lies in the nature of the computations that are performed, which involve model fitting rather than filtering. The following are some of the key aspects of the approach: (1) It provides a unified method of detecting various types of global patterns by bottom-up recursive fitting of low-order polynomial models to the data. (2) It provides a method of delineating the detected patterns by top-down recursive refinement of the fitted data. (3) It allows for the detection of more complex types of global patterns by applying local feature detection processes to the fitted models. (4) The methods can be applied to gray level surfaces, to edges or curves, or to vector-valued data such as disparity or optical flow fields. The transition from local to global—from pixel arrays to descriptive data structures—has traditionally been a major point of discontinuity in vision systems. The approach described in this paper offers the promise of making this discontinuity much less abrupt.

REFERENCES

Baugher, E. S., & Rosenfeld, A. (1986). Boundary localization in an image pyramid. *Pattern Recognition*, **19**, 373-395.

Bolle, R. M., & Cooper, D. B. (1984). Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **6**, 418-429.

Brady, J. M., & Asada, H. (1984). Smoothed local symmetries and their implementation. *International Journal of Robotics Research*, **3(3)**, 36-61.

Burt, P. (1979). *Hierarchically derived piecewise polynomial approximations to waveforms and images* (Report No. TR-838). College Park: University of Maryland, Computer Vision Laboratory.

Cornsweet, T. N. (1970). *Visual perception* (p. 27). New York: Academic Press.

Dunn, S. M., Janos, L., & Rosenfeld, A. (1983). Bimean clustering. *Pattern Recognition Letters*, **1**, 155-160.

GROSS, A. D. (1986). *Multiresolution object detection and delineation* (TR-1613). College Park: University of Maryland, Center for Automation Research.

HARTLEY, R. L. (1985). Segmentation of optical flow fields by pyramid linking. *Pattern Recognition Letters*, **3**, 253-262.

HARTLEY, R. L., & ROSENFELD, A. (1985). Hierarchical line linking for corner detection. In S. Levialdi (Ed.), *Integrated technology for parallel image processing* (pp. 101-119). New York: Academic Press.

HONG, T. H., & ROSENFELD, A. (1984). Compact region extraction using weighted pixel linking in a pyramid. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **6**, 222-229.

HONG, T. H., & SHNEIER, M. (1984). Extracting compact objects using linked pyramids. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **6**, 229-237.

HONG, T. H., SHNEIER, M. O., HARTLEY, R. L., & ROSENFELD, A. (1983). Using pyramids to detect good continuation. *IEEE Transactions on Systems, Man, & Cybernetics*, **13**, 631-635.

MARR, D. (1982). *Vision* (p. 95). San Francisco: Freeman.

MINOR, L. G., & SKLANSKY, J. (1981). The detection and segmentation of blobs in infrared images. *IEEE Transactions on Systems, Man, & Cybernetics*, **11**, 194-201.

ROSENFELD, A. (ED.). (1984). *Multiresolution image processing and analysis*. Berlin: Springer.

ROSENFELD, A., THURSTON, M., & LEE, Y. H. (1972). Edge and curve detection: Further experiments. *IEEE Transactions on Computers*, **21**, 677-715.

WERTHEIMER, M. (1958). Principles of perceptual organization. In D. C. Beardslee & M. Wertheimer (Eds.), *Readings in perception* (pp. 115-135). Princeton, NJ: Van Nostrand.

NOTES

1. Many variations of this simple root node selection criterion can be formulated, based on different assumptions about the distribution of local property values in the image.

2. Overall measures of the *good continuation* between two curve ends can be defined in terms of the total bending energy of the minimum-energy curve joining the ends.