

Extracting Straight Lines

J. BRIAN BURNS, ALLEN R. HANSON, MEMBER, IEEE, AND EDWARD M. RISEMAN, MEMBER, IEEE

Abstract—This paper presents a new approach to the extraction of straight lines in intensity images. Pixels are grouped into *line-support* regions of similar gradient orientation, and then the structure of the associated intensity surface is used to determine the location and properties of the edge. The resulting regions and extracted edge parameters form a low-level representation of the intensity variations in the image that can be used for a variety of purposes. The algorithm appears to be more effective than previous techniques for two key reasons: 1) the gradient orientation (rather than gradient magnitude) is used as the initial organizing criterion prior to the extraction of straight lines, and 2) the global context of the intensity variations associated with a straight line is determined prior to any local decisions about participating edge elements.

Index Terms—Boundary extraction, edge-analysis, gradient-based segmentation, image processing, line parameters, line representation, plane-fitting, straight lines.

I. INTRODUCTION

THE organization of significant local intensity changes into the more global abstractions called “lines” or “boundaries” is an early, but important, step in the transformation of the visual signal into useful intermediate constructs for interpretation processes. Despite the large amount of research appearing in the literature, effective extraction of straight lines has remained a difficult problem in many image domains. There are two goals of this paper: 1) the development of mechanisms for extracting straight lines from complex images, including lines of arbitrarily low contrast; and 2) the construction of an intermediate representation of edge/line information through which high-level interpretation mechanisms have efficient access to relevant lines.

To the degree that straight lines may be effectively extracted and efficiently represented, a variety of other intermediate processing goals are greatly facilitated. Curved lines can be approximated reasonably well as aggregates of piecewise-linear segments. In many cases, continuous representations of a boundary may be derived from adjacent linear segments by treating differences in their orientations as local curvature estimates. In addition, textured regions can be extracted as aggregates of line elements with specific common properties of length, contrast, orientation, etc.

Manuscript received February 1, 1985; revised August 14, 1985. Recommended for acceptance by W. E. L. Grimson. This work was supported in part by the Air Force Office of Scientific Research under Contract F49620-83-C-0099.

The authors are with the Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.

IEEE Log Number 8406515

A. Problems in Edge Extraction

Edges are usually defined as local discontinuities or rapid changes in some image feature, such as image luminance or texture. These changes are detected by a local operator, usually of small spatial extent with respect to the image, that measures the magnitude of the change and, in many cases, its orientation as well. Lines are commonly defined as collections of local edges that are contiguous in the image. Thus, many algorithms rely on a two-step process for line extraction: detection of local edges that are then aggregated into the more globally defined lines on the basis of various grouping criteria.

In the one-dimensional case, an ideal edge is a step change in the value of the underlying feature. In two dimensions, the ideal edge may be viewed as a step discontinuity in the values of the image feature in a direction perpendicular to the spatial orientation of the edge. We will refer to a straight line as a set of collinear and contiguous edges; i.e., a straight line has a length associated with a continuous discontinuity. Shortly, we will discuss the additional constraints that we impose on the intensity change to organize them into straight lines. Since ideal step changes are rarely found in real images, the magnitude of the feature change across a line is usually distributed over an area. Hence, the underlying image structure supporting a line has a width measured perpendicular to the line orientation in addition to its length. We refer to the collection of pixels so defined as a **line-support region**.

Note that our use of the term “line” differs from some researchers [26], [7], [13] who use the term “line” to refer to image events in which the intensity surface forms a ridge, possibly of narrow width, for which there is no distinct location for the boundaries on either side of the ridge. This view is related to the “roof” intensity profile of edges in the Binford–Horn line tracker [15]. In our view, these narrow linear image events will have a width formed by two locally parallel lines of opposite contrast. It is only the location of the lines that is ambiguous, not their existence. In the case where the ridge in the intensity surface is very narrow, even to a subpixel level, we are taking the position that if the difference in adjacent pixels is meaningful, it can be used to define a narrow region with parallel lines delimiting this image event.

The problems encountered with local edge operators are widely known and are related to 1) the possibly small spatial extent of the operator relative to the events they are designed to detect, 2) the deviation of actual image data from assumed models, and 3) aliasing due to the discrete

nature of the digitization process. The intensity variation representing a local edge is often spatially distributed over an extended area due to complex scene lighting conditions interacting with scene surfaces exhibiting varying surface orientation and reflectances. In real images, edges usually do not consist of step functions, but rather are formed by wider and more irregular changes in measured intensity. In most practical situations, the image data are noisy and, since edges are high spatial-frequency events, edge detectors enhance the noise. The edge maps resulting from application of a local edge detector are usually very dense and do not distinguish between edges resulting from object boundaries, shadows, and changes in surface reflectance and/or orientation. When the intensities on one side of a line change (e.g., a changing background behind an occluding surface), then there may be significant variation in edge contrast down the length of the line.

In order to overcome the problems caused by the mismatch between gradient widths and operator spatial extents, a family of approaches involving hierarchical edge masks have been proposed [28]. The most well known of these is the Marr-Hildreth zero-crossing operator [21], defined as the Laplacian of a Gaussian over increasingly larger spatial extents. However, since fine-detail (high-frequency) image events and coarse-structured (low-frequency) image events respond optimally to different size operators, the appropriate size of the operator must be determined in each different area of the image. Related algorithms involve the application of a set of hierarchical edge masks of varying resolution at selected orientations at all image locations.

Following the initial edge extraction process, various techniques have been proposed to aggregate the local information into more global line-like structures and to discard unimportant or redundant information, a difficult task in many domains. These methods include Hough transforms [5] that may be generalized to detect nonlinear boundaries and specific shapes [1]; edge tracking and contour following [23], curve fitting [25], graph theoretic methods [22], relaxation algorithms [30], hierarchical-refinement techniques [16], [8], [14], and high-level model-based processes [29].

The problems cited above in the discussion on edge operators pose difficulties for the aggregation processes as well. In many cases, the local operators misplace or entirely miss edges, a single real edge may result in several strong operator responses at different (often parallel) locations, and the underlying data may not conform to expectations built into the grouping process. Low-contrast lines, because of their low signal-to-noise ratio, are often troublesome.

B. Gradient Magnitude versus Gradient Orientation

The straight-line extraction technique developed in Section II is based on two observations about many line extraction algorithms: 1) they lack a global view of the underlying image structure prior to making local decisions about edge events and 2) they relegate information

about edge *orientation* to a *secondary role* in the processing. In most edge and line extraction algorithms, the magnitude of the intensity change is used in some manner as a measure of the importance of the local edge. While edge-orientation information may be used to modulate the grouping process applied to the strong edges, the edge magnitude usually has the central and dominating influence. It is our view that edge orientation carries important information about the set of pixels that participate in the intensity variation that underlies the straight line, particularly its spatial extent.

Gradient orientation is defined as the direction of maximum gray-level change as measured over a small area around a pixel, or equivalently, as the local direction of steepest ascent (or descent) on the intensity surface. Our model of the pixels comprising the intensity surfaces associated with straight lines in digitized images has two characteristics:

- 1) the local gradient magnitude (measured over a small local window) will vary significantly over the intensity surface, for reasons cited earlier, particularly in the direction orthogonal to the line; and
- 2) the local gradient orientation will vary relatively little throughout the entire intensity surface.

It is our observation that these characteristics are true of most of the straight lines that we wish to extract in digitized images. Based upon the consistency of the local gradient orientation, we have developed a simple algorithm for extracting the "line-support region," the entire set of pixels comprising each such intensity surface. In this way, the difficult step of extracting whole lines can, to a large extent, be reduced to a simple grouping and connected-components process. The additional benefit of isolating these support regions is that other aspects of the line, such as contrast and width (or fuzziness), can be more accurately measured.

Surprisingly, global approaches for straight-line extraction, such as Hough transform methods [5], [1], do not exploit orientation as much as one might think. Although the histogram buckets in (r, θ) coordinates encode edge orientation in terms of collinear sets of edges, once again the magnitudes of edges are likely to dominate. The global process for extracting lines is dependent upon finding strong peaks in the transform. All Hough techniques use edge magnitude in the voting process in some manner, either via a proportional weight or via thresholding so that only strong edges vote. Thus, it is very difficult to extract long, coherent, low-contrast lines in a general manner because their response in (r, θ) -space is reduced by the voting process, they may be hidden by high-contrast information and there may be other types of noise present.

C. A New Approach—Organizing Line-Support Contexts

The technique presented here was motivated by a need for a straight-line extraction method that would find straight lines in reasonably complex images, particularly those lines that are long but not necessarily of high con-

trast. A key characteristic of the approach that distinguishes it from most previous work is the *global organization* of the *supporting line context* prior to any decisions about the relevance of local intensity changes.

An estimate of the local gradient orientation at each pixel is the basis of these first organizing processes. Grouping pixels into line-support regions avoids the plethora of responses from masks of varying sizes and orientations, as well as unnecessary complexity in the subsequent organizing mechanisms. It allows the extraction of straight lines despite weaknesses in line clarity due to local variations in width, contrast, and orientation. It directly addresses the problems associated with the size of the edge operators and determines the extent of support to be given to edges and lines directly from the underlying data.

The approach has its roots in the “gradient-collection” processes of Hanson, Riseman, and Glazer [10]. In the terms discussed in this paper, the gradient-collection process utilized a data-directed mechanism to organize the full context of the edge in one direction at a time (the horizontal and vertical components) over the width of a monotonically increasing or decreasing section of the intensity profile contributing to the edge (i.e., where the gradient sign was constant). The total gradient contrast was then distributed around the location of the centroid of the local gradient magnitudes in the edge profile. This process organized contrast information across the width of an edge without committing to any fixed size or set of sizes for the edge operator. In a similar vein, Ehrich and Foith [6] organized one-dimensional intensity profiles into a hierarchical data structure before interpreting the information and making decisions about what constitutes a meaningful edge. Both of these techniques capture global gradient information that results in a more accurate assessment of total edge magnitude across its width.

Haralick [12] also processes the intensity surface in order to make decisions about lines, but the key difference is that his surface patches are local, and one faces the same sort of difficulties in organizing this information as one does in the output of local edge operators.

The approach in this paper has generalized the global, contextual organizing processes to two dimensions, grouping image pixels across the width of an edge as well as down the length of the edge, to form the basis for extracting a straight line. All pixels in these line-support regions contribute to both the final representation of the line and the generation of a set of descriptive attributes that are useful for further processing of the line data. The line-support regions might also be useful in separating the straight lines into intrinsic images [3] representing edges and lines of different types, such as illumination, texture, reflectance, orientation, etc.

II. A REPRESENTATION AND PROCESS FOR EXTRACTING STRAIGHT LINES

A. Overview

The general approach to extracting straight lines is to group the pixels into line-support regions on the basis

of gradient orientation, and then to extract from each region a straight-line segment. Note that every intensity variation, including very low magnitude changes, will initially be extracted as a weak line segment (sometimes of great width). During the interpretation of these lines, adjacent low-contrast support regions can be grouped into homogeneous regions and filtered so that they are not viewed as weak straight lines.

There are four basic steps in extracting straight lines.

- 1) *Group pixels into line-support regions* based on similarity of gradient orientation. This allows data-directed organization of edge contexts without commitment to masks of a particular size.

- 2) *Approximate the intensity surface by a planar surface.* The planar fit is weighted by the gradient magnitude associated with the pixels so that intensities in the steepest part of the edge will dominate.

- 3) *Extract attributes* from the line-support region and the planar fit. The attributes extracted include the representative line and its length, contrast, width, location, orientation, and straightness.

- 4) *Filter lines* on the attributes to isolate various image events such as long straight lines of any contrast; high-contrast short lines (heavy texture); low-contrast short lines (light texture); homogeneous regions of adjacent very low contrast lines; and lines at particular orientations and positions.

B. Grouping Pixels into Line-Support Regions via Gradient Orientation

Fig. 1 shows four representative images used to illustrate the grouping and straight-line extraction process. Fig. 2(a) is a 32×32 intensity subimage used to illustrate the details of the algorithm; results are shown for the full images in subsequent sections. Fig. 2(b) shows the intensity surface of this subimage, while Fig. 2(d) depicts the corresponding gradient image in which the length of the vector encodes gradient magnitude. Gradient magnitude and orientation have been estimated by convolving the image with the two masks shown in Fig. 2(c). Note that the sign of the gradient encodes dark-to-light or light-to-dark intensity changes that are 180 degrees apart. Thus, intensity surfaces that form a ridge will be detected as two different line-support regions.

- 1) *Choice of Mask for Computing the Gradient:* There are a variety of masks that can be employed in the computation of the gradient, including those organized hierarchically according to mask resolution. Large masks tend to smooth the image and reduce the clarity of fine detail, or even remove it completely. Since one of our primary goals is the recovery of lines corresponding to fine detail, we wish to select the smallest possible masks that will produce estimates of gradient orientation. The mask selected must maintain lines associated with alternating one-pixel-wide regions [such as parts of the rain-gutter, siding and window-trim in Fig. 1(a) and (b)] and also provide symmetric responses with respect to rotation of the line in the image.

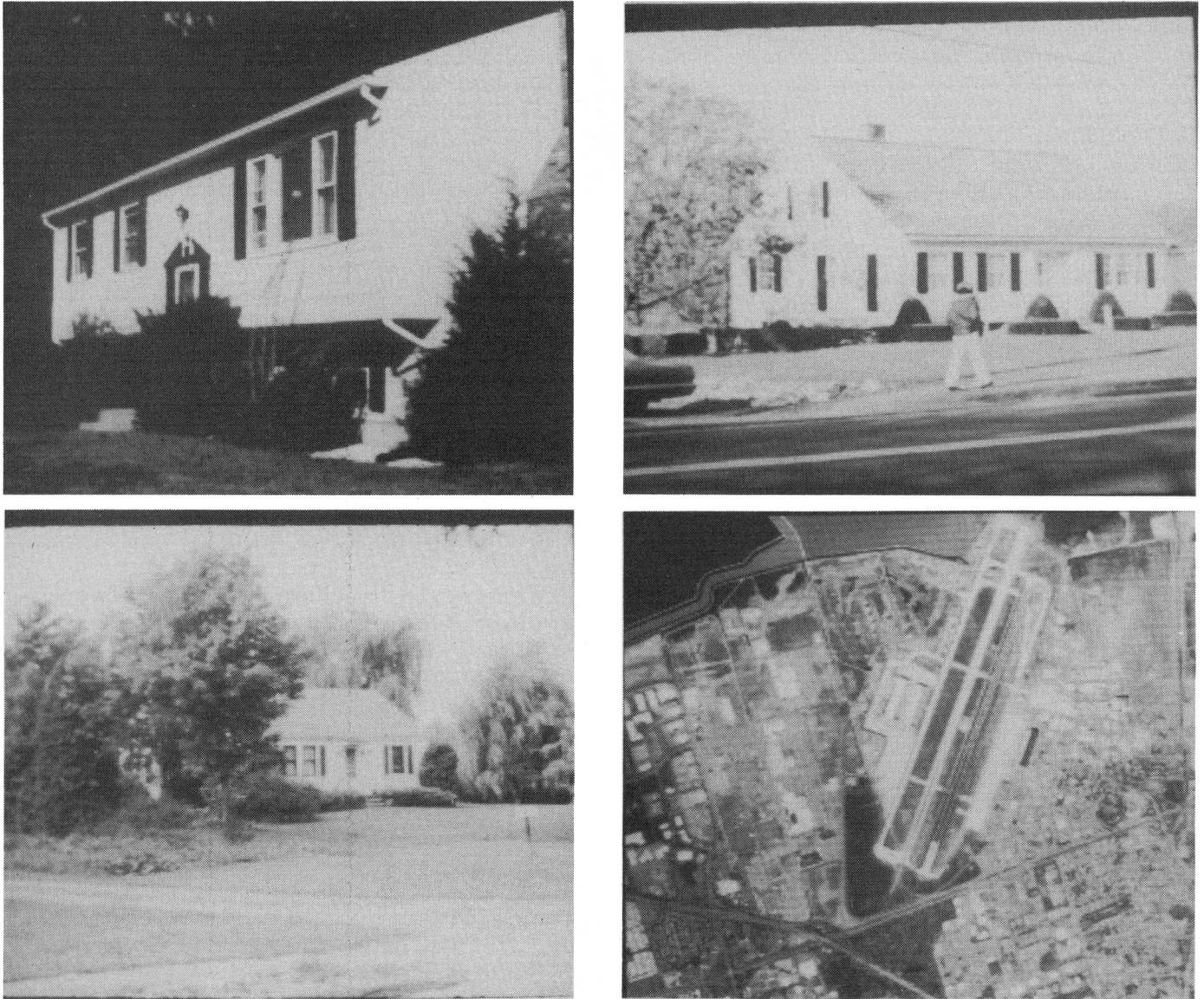


Fig. 1. Four natural images used to demonstrate straight-line extraction.

The sensitivity to detail and rotational symmetry of four small edge masks, 1×2 , 1×3 , 2×2 , and 3×3 , shown in Fig. 3(a) will be compared by applying them to two test images. Note that all masks are no larger than a 3×3 window, and one of them is the smallest possible edge operator, a 1×2 mask.

The first test image shown in Fig. 3(b) is composed of a field of alternating horizontal black and white strips of 1 pixel width and is intended to test the ability of the mask to respond to fine detail. Fig. 4(a)–(d) shows the results of applying the four masks to the dense field of strips. Note that the 1×3 and 3×3 completely fail to detect any intensity variation at all! Thus, these masks will be rejected because high-contrast 1-pixel-wide regions can be missed.

The test image of Fig. 3(c) is composed of a diagonal edge reflected about the vertical axis. This test image will give a sense of edge responses to rotated lines. Fig. 4(e)–(h) demonstrates the symmetric response of the 2×2

mask to the two diagonal lines versus the nonsymmetric response of 1×2 mask that is the smallest possible mask.

On the basis of the criterion described, the 2×2 mask appears to be the best choice. In addition, Haralick has shown that this particular mask is optimal among 2×2 operators [12]. Thus, the 2×2 mask was chosen as our operator to estimate gradient magnitude and orientation. All results shown in the following sections were obtained using the 2×2 mask. The local gradient orientation was computed by

$$\tan^{-1} G_V(i, j)/G_H(i, j)$$

where $G_V(i, j)$ and $G_H(i, j)$ are the vertical and horizontal components of the gradient obtained from the mask applied at pixel i, j . Further studies will be required to determine the impact of the size and form of the edge operator on the overall process.

2) *Segmentation of the Gradient-Orientation Image Using Fixed Partitions*: Once local gradient orientations

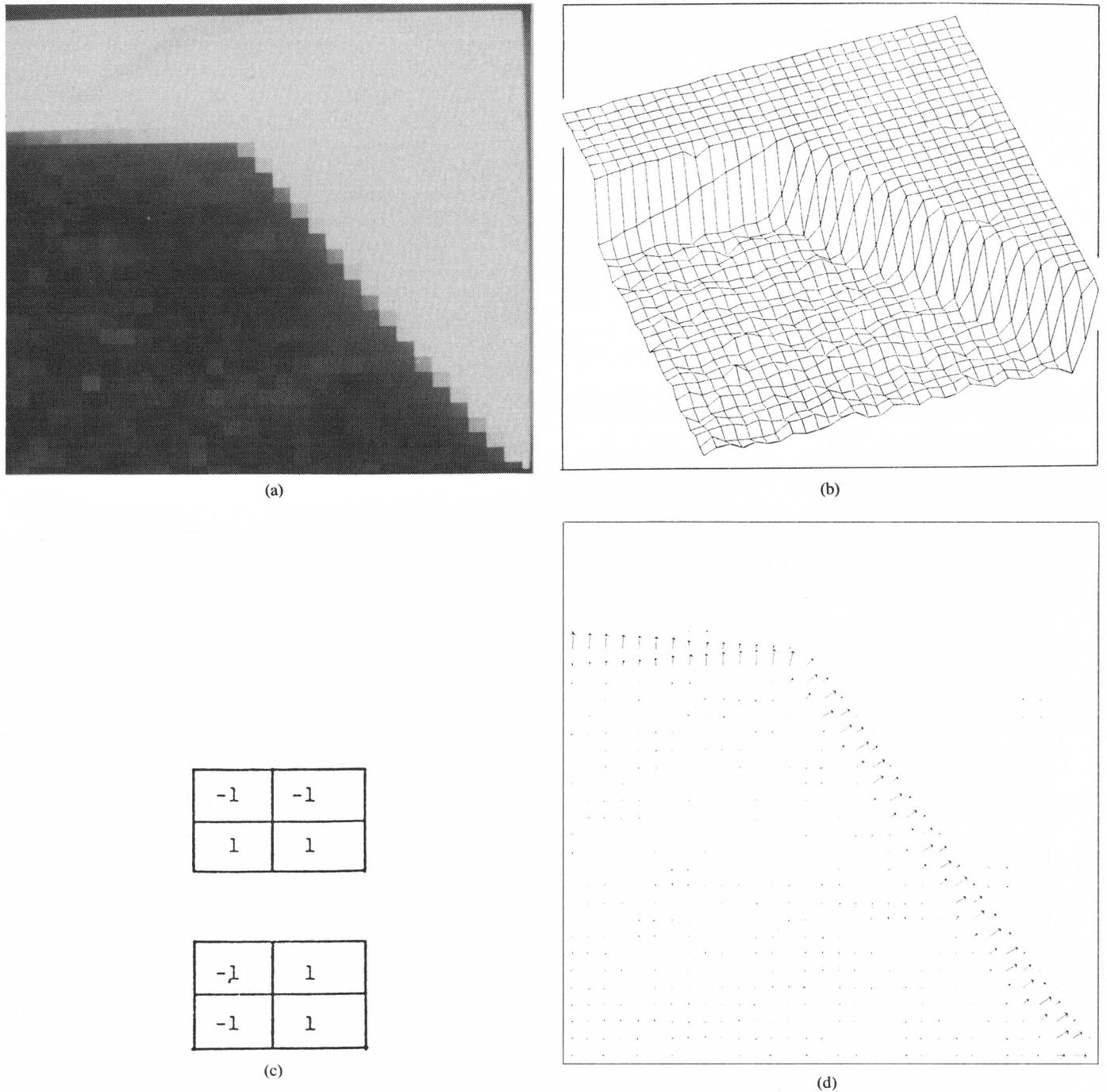


Fig. 2. The first step in forming gradient regions involves estimating the gradient direction (orientation) at all points in the image. (a) A 32×32 subarea of a house image that will be used to illustrate the process. (b) An intensity-profile representation of the intensity array. (c) The 2×2 operators used to estimate dI/dx and dI/dy , from which the local gradient orientation is obtained. (d) The resulting gradient vectors encoding magnitude (vector length) and orientation.

have been estimated, they are grouped into regions. The problem can be viewed as one of segmenting the gradient-orientation image, and the usual difficulties of region-segmentation algorithms are encountered. Although local groupings can assure local similarity, regions can be formed that include pixels with very dissimilar orientation attributes due to a slow drift in the orientation from pixel to pixel. Thus, region-growing techniques [5], [2] are not

applicable because even occasional over-grouping errors can cause disastrous results. Changes in line orientation at corners and junctions of straight lines [as in the image in Fig. 5(b)] can produce intermediate gradient orientations instead of a clear discontinuity in gradient orientation; the result could be undesirable pixel groupings if local region-growing is employed.

A grouping process was employed that avoids some of

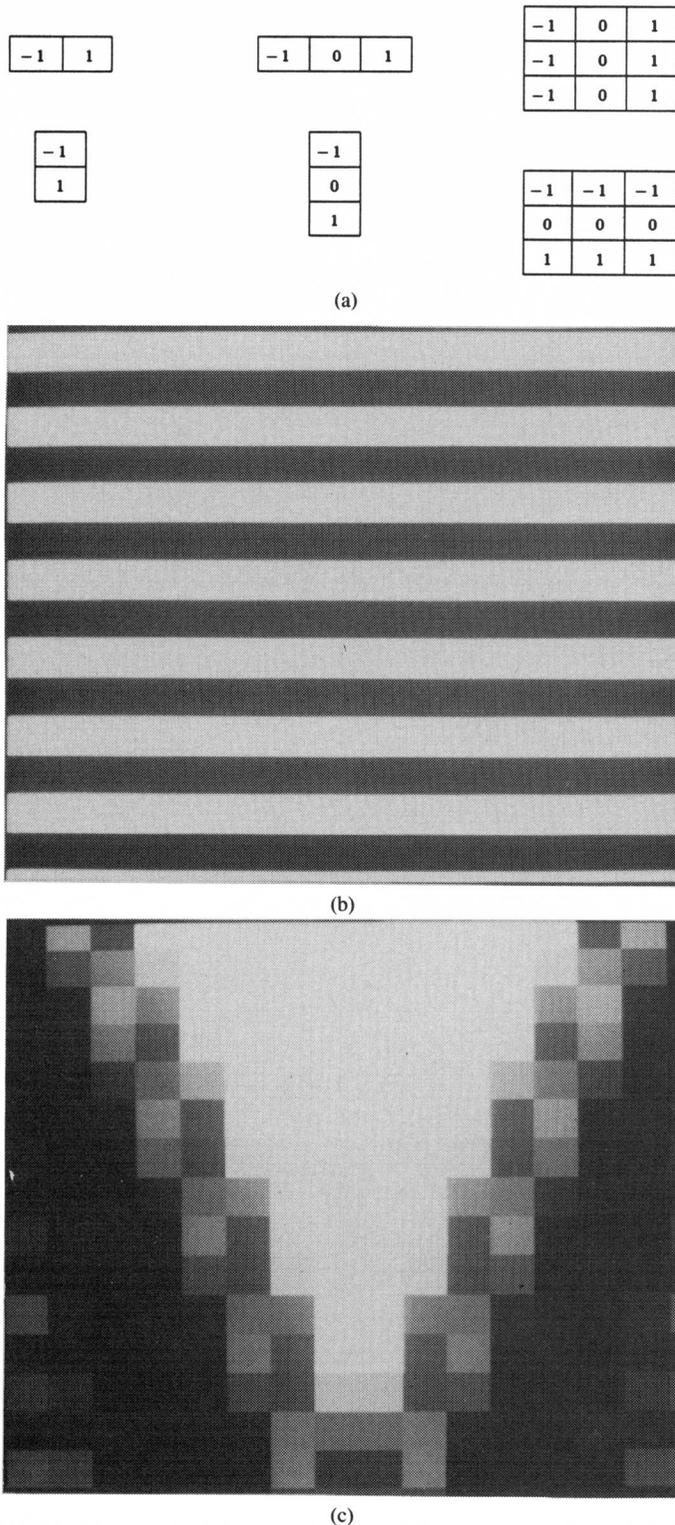


Fig. 3. Edge masks and test images. (a) Three other small edge masks that could be used to estimate the horizontal and vertical components of gradient magnitude. (b) Test image of horizontal stripes for analyzing responses for high frequency data. (c) An image of a diagonal boundary reflected about its central, vertical axis to compare symmetry of edge-mask responses.

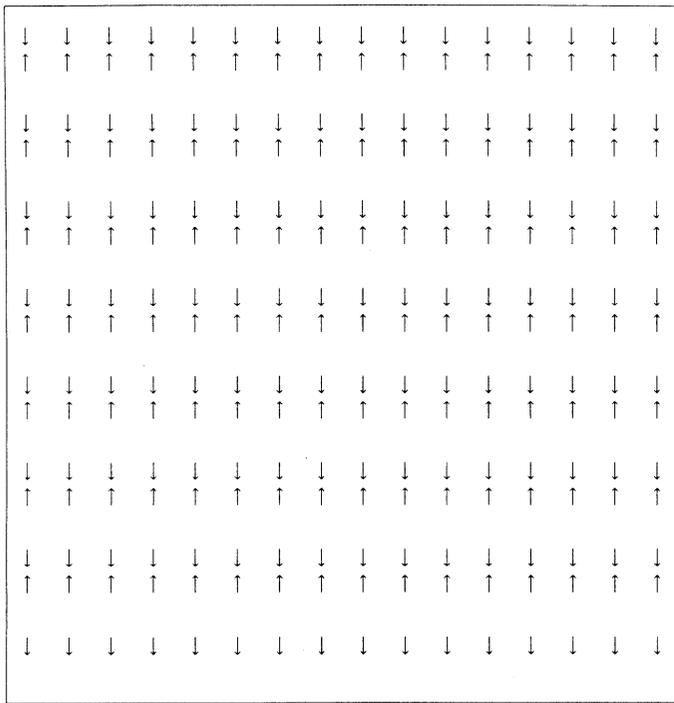
these problems. We shall introduce the basic idea with a simplified approach using fixed partitioning of the orientation feature; in the next subsection we extend it to a more effective process using overlapping partitions. In the

fixed-partition scheme, the 360 degree range of gradient directions is arbitrarily quantized into a small set of regular intervals, say, eight 45 degree intervals or sixteen 22.5 degree intervals [Fig. 5(a)], and each gradient vector is labeled according to the partition into which it falls. A simple connected-components algorithm is then used to form distinct region labels for groups of adjacent pixels with the same orientation label [Fig. 5(b)]. If our conjectures about edge orientation are correct, then pixels participating in the line-support context of a straight line will be in the same (or sometimes adjacent partitions), and adjacent pixels that are not part of the same straight line will usually have different orientations.

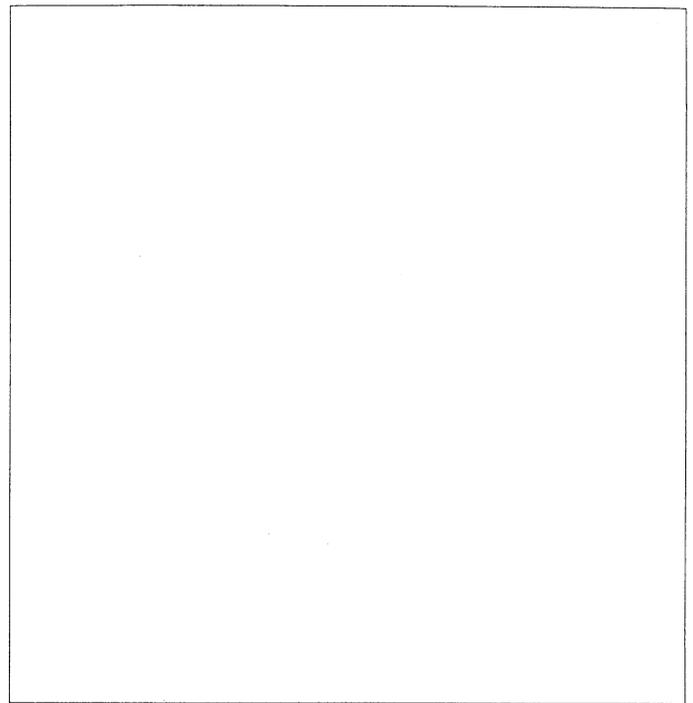
3) *Segmentation of the Gradient-Orientation Image Using Overlapping Partitions:* Problems with the simple approach to grouping described in the previous section are related to the arbitrary placement of the boundaries of the fixed partitions and the resulting insensitivity to the possible distributions of edge directions of any particular straight line. First, visually distinct straight lines that are spatially contiguous can be improperly overmerged because they have similar orientations and (partially) fall in the same orientation bucket. Second, a straight line can produce fragmented support regions if the distribution of gradient orientations happens to lie across a partition boundary. The overmerging problem tends to be reduced as the partition size gets smaller, but the fragmentation problem demands larger sizes. Fig. 6 illustrates a gradient orientation segmentation of two subimages using 8 and 16 partitions (45 degree and 22.5 degree intervals, respectively) on each. These examples demonstrate the overmerging/fragmentation tradeoff.

A reasonably simple and effective extension to the fixed-partitioning scheme involves the use of two overlapping sets of partitions: when one partition fragments a line because it lies across a partition boundary, the other will tend to place this same line entirely within a partition. Fig. 7(a), (b) shows a roof boundary that serves as a clear example. Fig. 7(a) was generated by eight 45-degree partitions, with the first centered on 0 degrees; Fig. 7(b) was also produced using eight 45-degree partitions, but the set was rotated half a partition so that the first bucket is centered at 22.5 degrees. Note that the first set recovers the top boundary, while the second set recovers the side. Regions of low contrast outside those shown were removed to isolate the gradient regions for illustrative purposes.

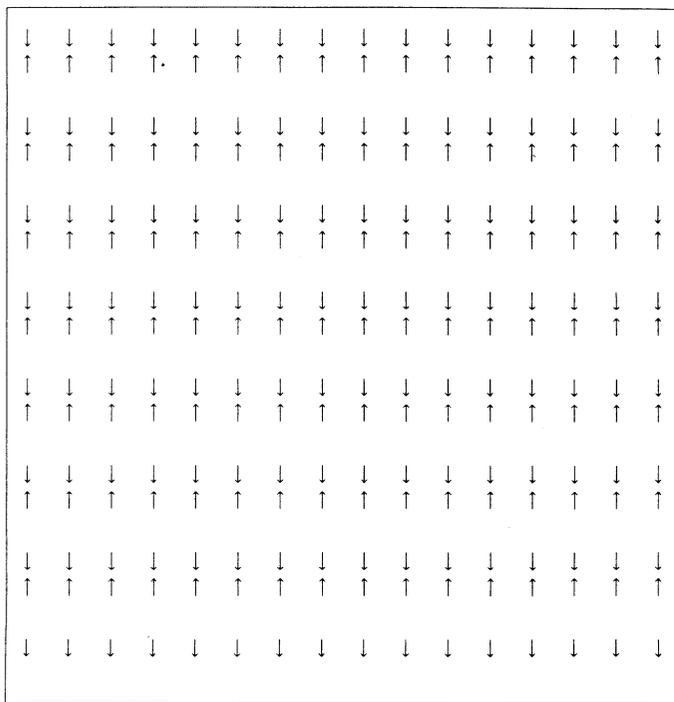
The critical problem of this approach is the merging of the two representations in such a way that a single line in the image is principally associated with a single line-support region. The region considered best for pixel associations is the one that provides an interpretation of the line that is longest. The following scheme is used to select regions and lines from the overlapping representations: 1) line lengths are determined for every region; 2) since each pixel is a member of exactly two regions (one in each segmentation), every pixel votes for and is associated with that region of the two that provides the longest interpretation; 3) each region receives a count of the number of



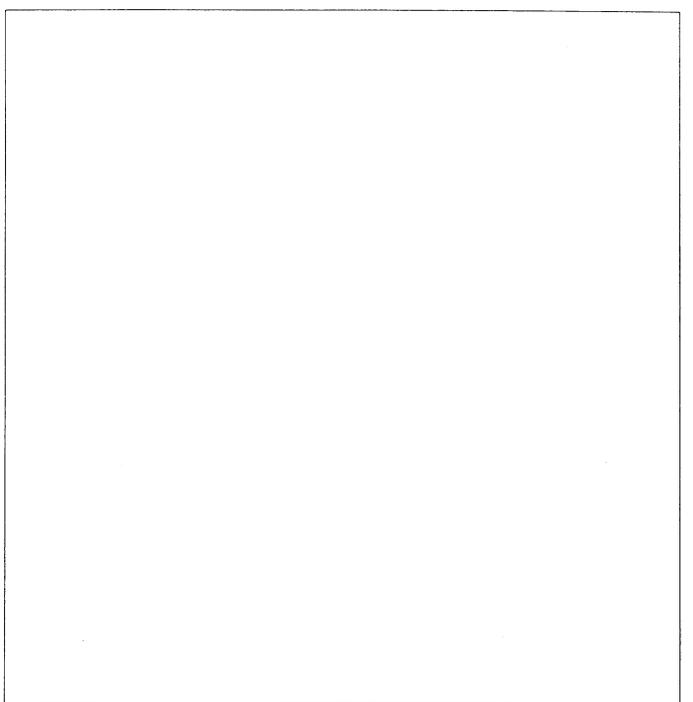
(a)



(b)



(c)



(d)

Fig. 4. (a)-(d) Gradient estimates for the horizontal-stripe image (1×2 , 1×3 , 2×2 , and 3×3 , respectively). Note that the 1×3 and 3×3 masks are unable to detect any of these intensity changes.

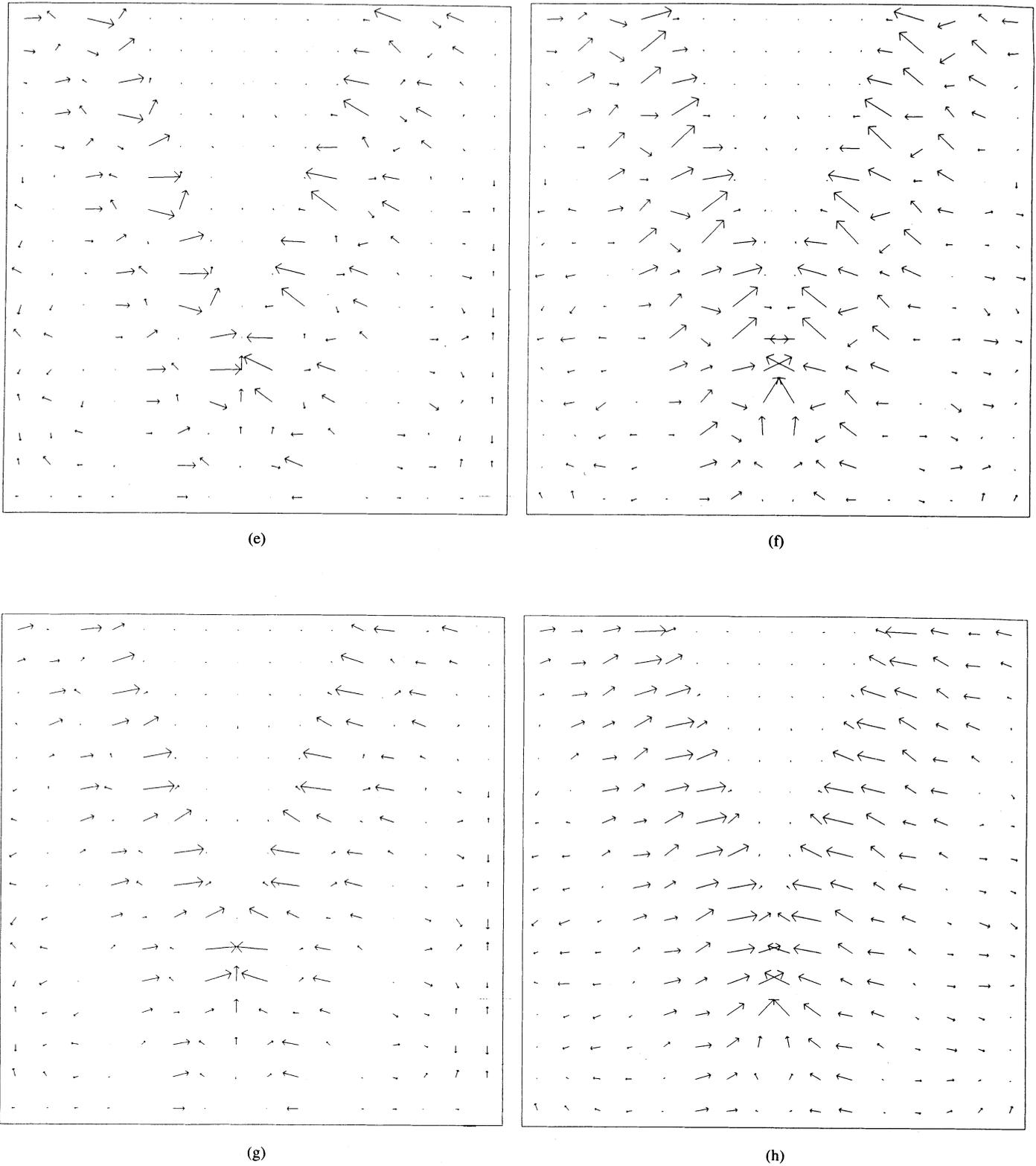
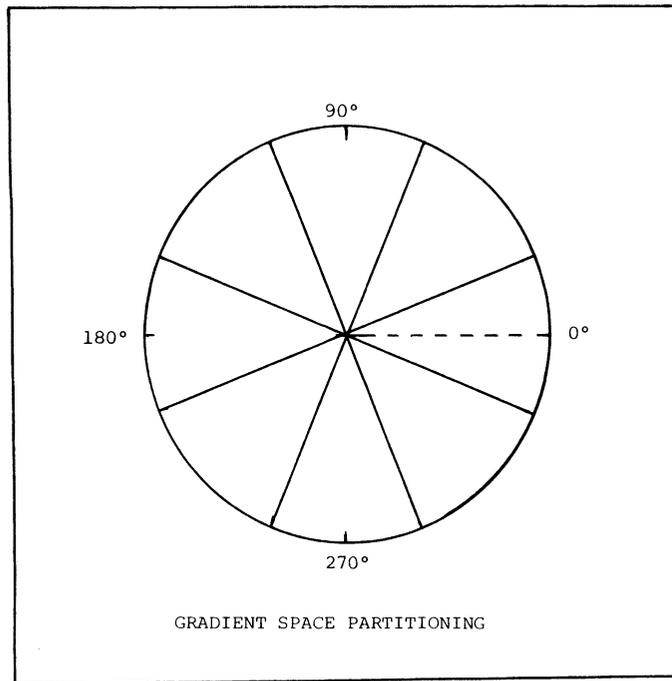
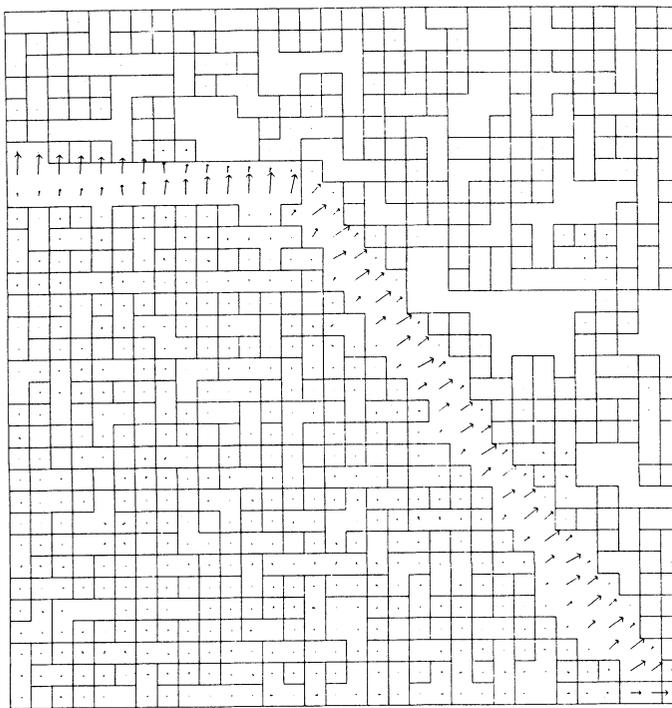


Fig. 4. (Continued.) (e)-(h) Gradient estimates for the reflected boundary (1×2 , 1×3 , 2×2 , and 3×3 , respectively). Note that the response of the 1×2 mask on the two boundaries is not symmetric.



(a)



(b)

Fig. 5. Gradient-orientation regions are formed by grouping pixels based on the similarity of the local gradient-orientation estimates. (a) A partitioning of the data into fixed-orientation classes. (b) Regions produced by a connected-components algorithm applied to the labels of the orientation partitions.

its pixels that voted for it; and finally, 4) the “support” each region of the two representations is given is the percentage of the total number of pixels voting for it. The regions selected are those that have a majority, i.e., the support is greater than 50 percent. It should be noted that most regions have either very low or very high support. Fig. 7(c)–(d) show the regions in each segmentation that

received support greater than 50 percent: notice how the edges are now uniquely represented by gradient regions that cover their full extent.

C. Interpreting the Line-Support Region as a Straight Line

Each line-support region represents a candidate area for a straight line since the local gradient estimates share a common orientation. Once the line is extracted, its positional parameters will serve as the core of the structure’s symbolic description as well as a local coordinate system about which other attributes will be measured—such as length, width, contrast, and straightness. The key problem is to use the information contained in the underlying intensity surface to find the line.

The intensity surface associated with each line-support region is assumed to be a noisy representation of an ideal ramp, or sloped facet [12], that can be characterized by a planar surface. The parameters of this plane may be obtained by a weighted least-squares fit to the feature values. The weighting is a linear function of the local gradient magnitude and determines the contribution of a pixel’s intensity (or feature) value to the final planar fit. This has the effect of allowing strong local intensity variations to dominate the fit. In practice, the weighting is necessary to reduce the contribution of pixels near the tails of the intensity change where the magnitude of the change is often fairly small. Fig. 8 presents two examples of the weighted planar fit.

Now that a planar fit to the intensity surface is available, an obvious constraint on the orientation of the line is that it be perpendicular to the gradient of the fitted plane. Thus, this leaves only the problem of locating the line along the projection of the gradient. The region depicted in Fig. 9(a) and as dots in the surface plot of Fig. 9(b) will serve as our example. Note that the line-support region shown in Fig. 9(d) includes all the pixels contributing to the gradient estimate, as determined by taking the union of the set of pixels to which the mask was applied in computing the grouped gradient vectors. (Fig. 9(a) shows only the point estimates of the gradient.) A simple approach is to intersect the fitted plane with a horizontal plane representing the average intensity of the region weighted by local gradient magnitude as shown in Fig. 9(c); the straight line resulting from the intersection of the two planes is shown in Fig. 9(d) overlaid on the line-support region.

This approach is related to the gradient-collection mechanism used in [10], where the position of an edge in a one-dimensional intensity profile is determined as the centroid of the local gradient magnitudes along the one-dimensional profile. Thus, the larger gradient components of monotonically increasing or decreasing intensity changes are weighted more in determining the edge position. It is also similar to the slope-facet model proposed by Haralick [12], where the local intensity surface in the neighborhood of a pixel is modeled as a planar surface patch. This planar fit served as a model of the region

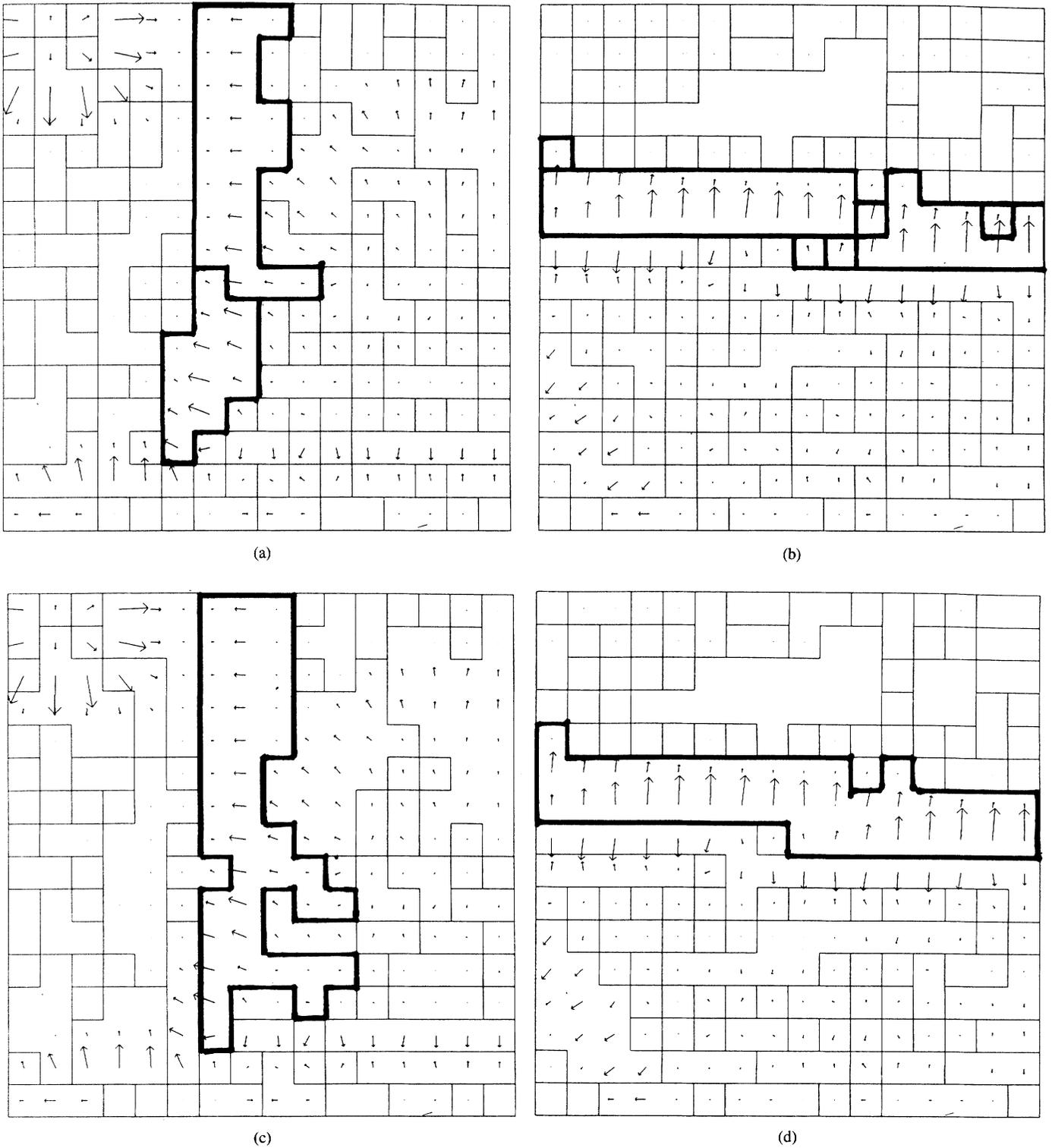


Fig. 6. Orientation partitions of different sizes affect the over-merging and fragmentation tradeoff. (a), (b) Use of a large number of partitions, say 16 intervals of 22.5 degrees, can have the desirable effect of reducing over-merging as shown in (a), but causes some undesirable fragmentation as in (b); (c), (d) Use of a smaller number of partitions, say 8 intervals of 45 degrees each, has the opposite effect. These segmentation problems cannot be solved by simple parameter tuning, although the partitioning technique provides a starting point for more sophisticated algorithms.

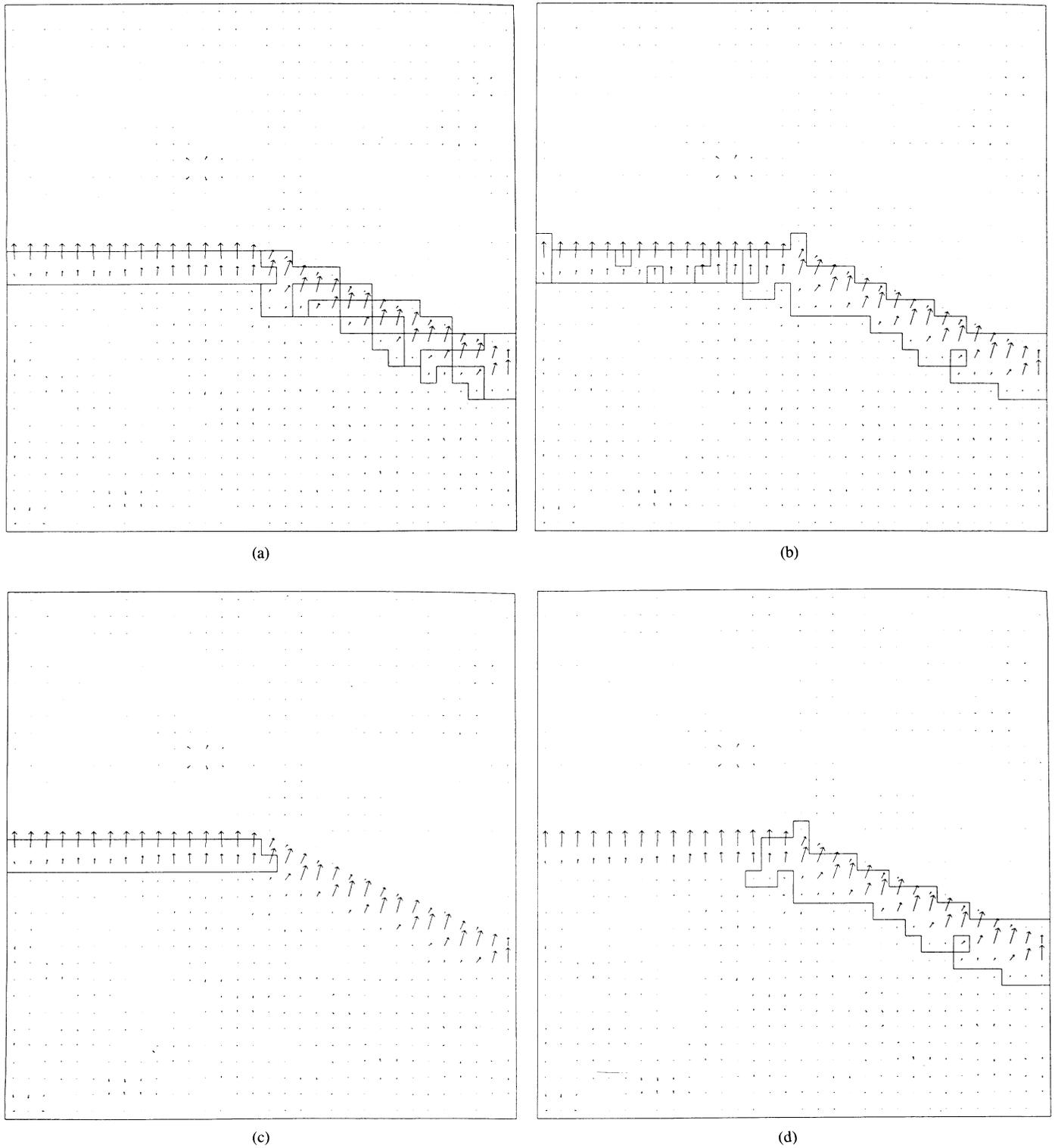
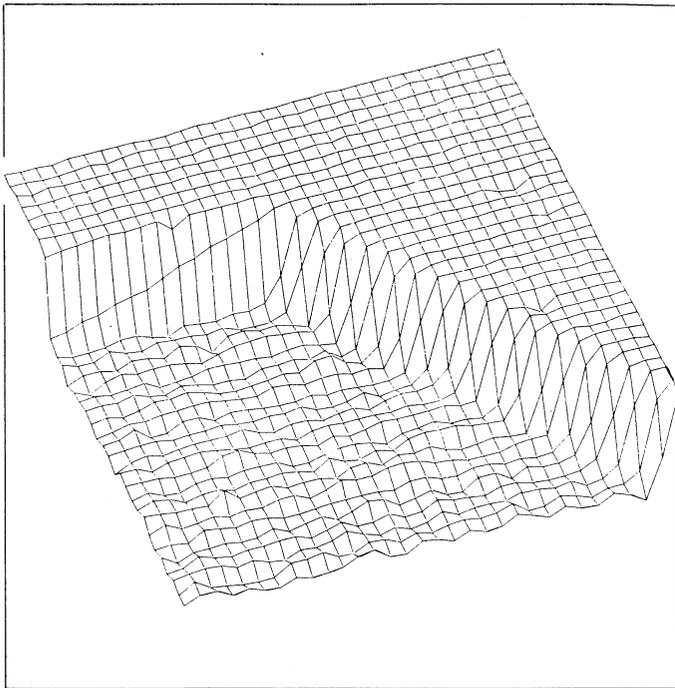
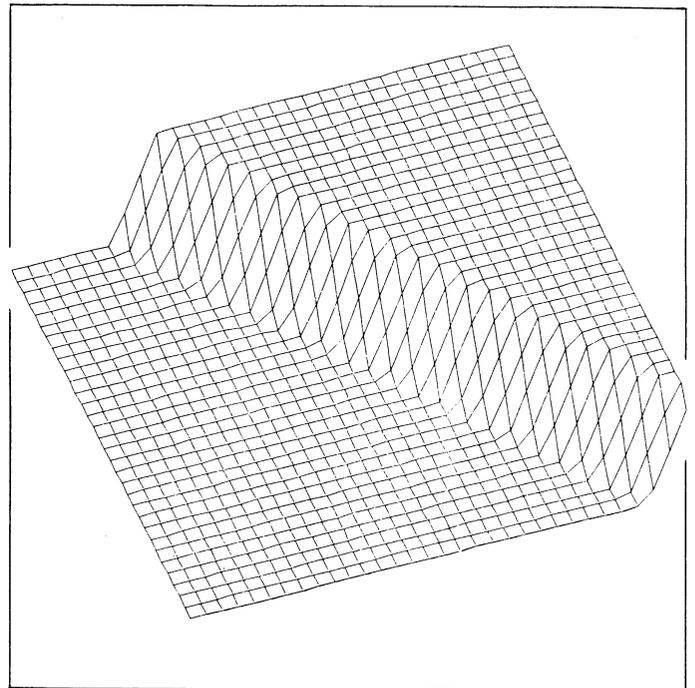


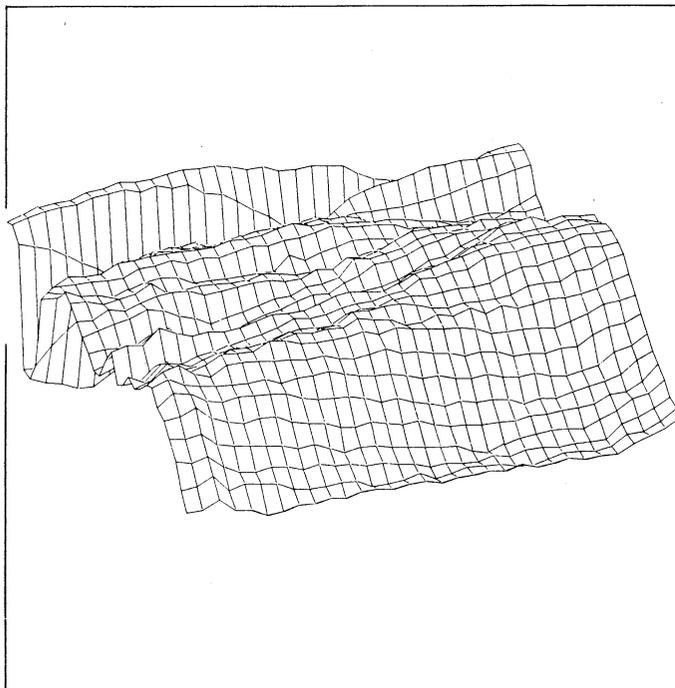
Fig. 7. Example of dual overlapping-partition technique applied to a simple roof-sky boundary (note that regions of low contrast have been removed for clarity). (a) Result of 8 partitions (45 degrees) with the first bucket centered at 0 degrees. (b) Result of partitioning with same bucket sizes, but with the first bucket centered at 22.5 degrees. (c) Removal of regions in (a) with support < 0.5 . (d) Removal of the regions of (b) with support < 0.5 . Note that there is exactly one region that covers the area completely for each straight line, though these regions may overlap slightly at their junction.



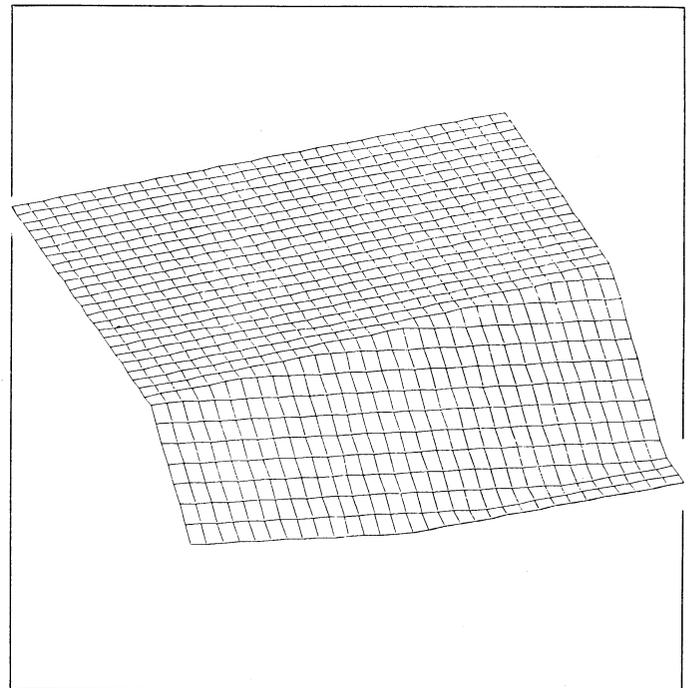
(a)



(b)



(c)



(d)

Fig. 8. Typical lines appearing in outdoor scene images and simple models of their profiles. (a) Surface plot of a relatively sharp and high-contrast edge. (b) Planar model of the line obtained by a least-squares fit weighted by gradient magnitude. (c) The set of pixels in the foreground of the surface plot represents a wide, low-contrast shadow boundary. (d) The planar model (foreground of plot) reasonably approximates the information in the shadow boundary line.

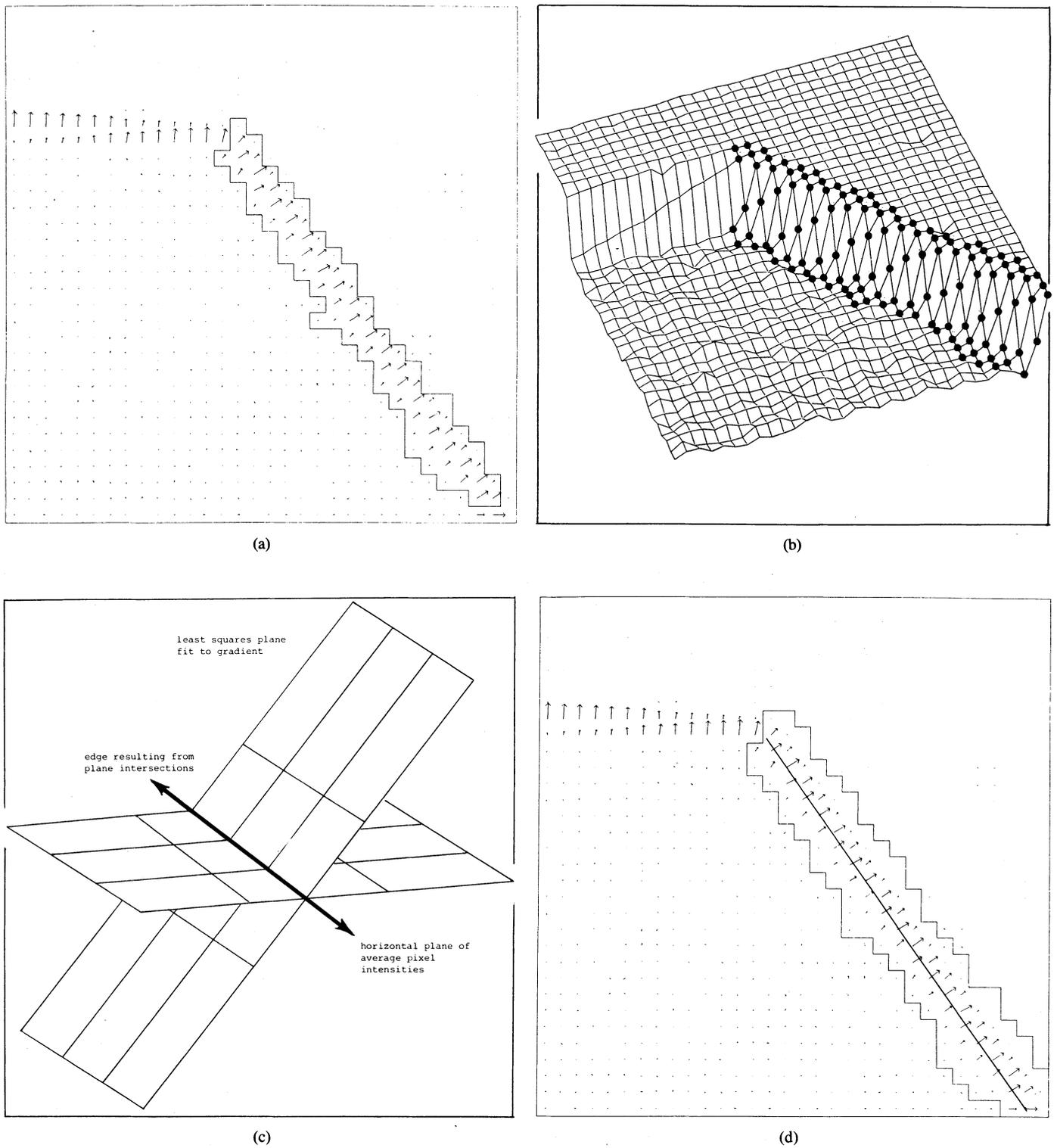


Fig. 9. A straight line is obtained from the group of pixels comprising the line-support region. (a) Gradient region resulting from grouping on the basis gradient orientation. (b) Pixels included in the line-support region are highlighted by dots. (c) The straight line is obtained by intersecting the weighted planar fit to the intensities with the horizontal plane representing the average intensity, again weighted by magnitude. (d) The resulting straight line overlaid on the set of pixels making up the line-support region.

structure and was used to determine whether or not the pixel was at a region boundary. In our case, however, the neighborhood size has been determined dynamically by the process that groups on orientation, rather than being fixed *a priori*.

III. ATTRIBUTES OF THE LINE-SUPPORT REGION

The line-support regions provide an excellent opportunity to study aspects of the line's behavior beyond the basic orientation and position parameters. In this section we demonstrate how the line-support region can be used as a source of detailed information about the line by using the underlying intensities to characterize the line and its associated region as a structure with length, width, contrast, steepness and, finally, straightness. The measures defined here should be considered first approximations to more complex analyses of the underlying structure of the intensity surface that are possible. These properties provide selection criteria for lines with particular properties.

Some properties of the line are immediately derivable from the parameter extraction process. Length is simply the distance between the two endpoints. Another measure is the gradient-weighted average intensity level of the plane used to find the line's lateral position.

A. Contrast, Steepness, and Width of a Line

Some other potentially interesting attributes of a line are properties of its profile (perpendicular to the edge). Characteristics of the profile of the intensity surface across a line include 1) *contrast*—the cumulative intensity change that occurs across the line; 2) *width* or *fuzziness*—the size of the interval, across the profile, within which the bulk of the intensity change occurs; and finally 3) *steepness*—the surface's slope within this interval. Clearly, these three properties are roughly related by $\text{contrast} = \text{width} * \text{steepness}$; and having measured two, one can use them to roughly estimate the other. These relationships are very clear and precise in the ideal ramp structure shown in Fig. 8(b).

Unfortunately, many actual intensity profiles across a line are highly nonlinear and the way that the intensity change is concentrated along the profile varies considerably; Fig. 8(a) and (c) shows the intensity surfaces associated with typical straight lines.

The slope of the plane fit to the intensity surface is one possible measure of the edge's steepness. However, the presence of wide soft shoulders on the profile will lower this slope below that which would be obtained by measuring the slope over the area of maximum (or "high") rate of change of the intensity surface. Of course the gradient weighting mechanism for pixel contribution to the plane fit reduces this effect, but there is still an issue of whether the whole intensity profile should be included. An alternative could be to fit piecewise-planar surfaces to more accurately reflect the nonlinear intensity surface, but in practice it may not be obvious how to accomplish this. Thus, for our treatment here the slope of the weighted plane fit will be used as the measure of the steepness.

Contrast seems to be much less affected by the actual shape of the profile, since it strictly involves the cumulative intensity change. For a single slice across the edge, contrast will simply be defined as the difference between the Max and Min of the values in the line-support region. A more effective version of this would be the average of the high feature values on one side of the line and the average of the low feature values on the other side of the line. This can be achieved by averaging the feature values of pixels in the upper and lower N percent of the line-support region, where N might be in the range of 10 percent.

This leaves the width of the line. It can be measured directly or taken as contrast divided by steepness from the above relationship. One would think that the spread of the total intensity change across the profile should be the major factor in determining width. However, it is hard to clearly relate this spread to some "width" measure because very wide low-gradient shoulders can distort this measure. One way is by means of the steepness measure presented above that already reflects the degree of spread: since it is derived by weighting with local gradient magnitude, the narrower the spread, the larger this measure is with respect to the average slope of the profile. If width is set to $\text{contrast}/\text{steepness}$, a narrower width than the direct computation from the intensity surface is obtained. The definition also forces the relationship between the three properties to be strict—allowing us to express our description of the actual edge in form of a simple, planar ramp approximation with a clear width, height and steepness (see Fig. 8(b) and (d) for resulting models).

B. Straightness of a Line via Orientation Variance and Iso-Contours

One final attribute that is quite important to estimate is straightness, that is, the degree to which the intensity surface of the support region truly represents a straight line. A wavering line whose local orientation remains within the range of a single partition can produce a line-support region and planar surface fit that is equivalent to that of a straight line. The global attributes of contrast, orientation, length, position, and average width of a wavering and a straight line could all be the same. However, the spatial distribution of the gradient-orientation vectors in the support region may be quite different.

A simple straightness measure is the variance in orientation, since it will be low if all vectors are similarly oriented. Of course, this measure is slightly complicated because it is computed on a circular scale. Mardia's [19] measure of the mean minimizes the cumulative cord length from the data to the estimate. The center of mass is found for the sample set by vector addition followed by division by the total magnitude. The resulting vector, the center of mass, points in the mean direction and the arcsine of its magnitude (radius) is the standard deviation (standard deviating angle) from the mean direction.

Another method for generating a straightness measure is to compute properties of the intensity iso-contour passing through the region. The contours are generated by

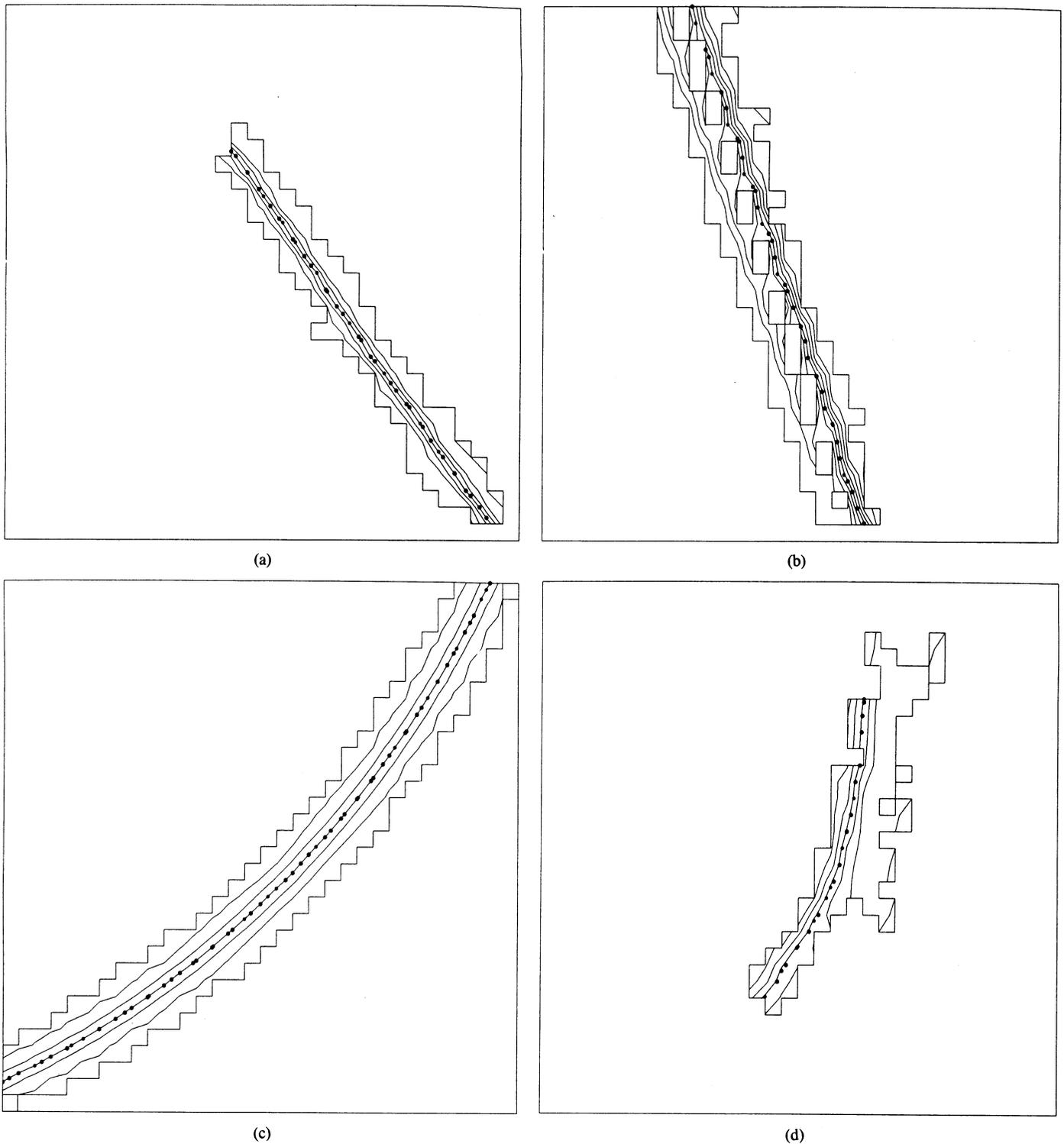


Fig. 10. Line straightness can be measured as the straightness of an iso-intensity contour passing through the line-support region. The contours depicted are of regularly spaced intensities, with the dotted contour being at the average intensity of the region, weighted by local gradient magnitude. (a) A relatively straight smooth line. (b) A straight but locally rough line. (c) A synthetic circular boundary. (d) Curved line obtained from an image of a fold in a shirt.

computing the intersection of a horizontal plane representing an intensity value with the actual intensity surface. In general, the iso-contours tend to run parallel to the extracted line because, of course, the orientation will be constrained to lie within a single orientation partition

in the gradient segmentation. However, the iso-contours will vary locally as a function of the intensity surface and may reflect any lack of straightness. Fig. 10(a)–(d) shows several intensity iso-contours for various types of lines: 1) the smooth straight line used as an example in Section

II-C and Fig. 9; 2) a rough but still straight roof line; 3) a synthetic, 45-degree circular arc; and 4) a rougher curved line. Simple linear interpolation of the intensity surface was used to compute the contour-scanline intersections.

The particular iso-contour used as the basis for the comparison is the gradient-weighted average intensity of the line-support region. This contour will tend to pass through points on the steepest parts of the intensity surface (depicted with dots in Fig. 10). These dots represent the interpolated locations between pixels where the contour has been detected.

The basic idea is to measure the average distance of the contour points to a straight-line fit to them and then divide this amount by the line length. This latter step is taken to make the measure scale independent: the straightness measure is then a unitless proportionality of lateral deviation of the curve over its length. The smaller this ratio is, the straighter the line is. Since this value tends to be very small, we multiply it by a thousand to give it a more reasonable range. The average distance of the contour points to a straight-line fit is computed using an eigenvector technique [5] and is taken to be the square-root of the mean-squared distance (SMSD).

This measure was applied to the four examples of Fig. 10 and appears to be effective. The smooth, straight edge [Fig. 10(a)] has a ratio of 2.0 (per 1000); the rougher straight edge [Fig. 10(b)] has a value 50 percent larger (3.0 per 1000). The curved edges have ratios more than ten times that of the smooth straight edge [27.0 per 1000 for both Fig. 10(c) and (d)]. Thus, the measure tends to separate the straight edges from the curved ones, although some smoothing of the contour or raw data might reduce some of the minor local variations due to digitization or noise. In comparison, the standard deviating angle of the gradient distribution and the average plane-fit error do not separate the straight from the curved as dramatically: the standard deviating angles (in degrees) for Fig. 10(a)–(d) were 6.1, 8.6, 12.7, and 11.7, respectively. The average plane-fit errors are 7.0, 23.2, 23.6, and 29.7, respectively (see Sections IV-C and IV-D for further results).

C. Straightness of a Line via Piecewise Average Orientation

This measure is more complex, but focuses upon the spatial distribution of orientation vectors that carries most of the information about curvature of the line. Once the representative line for each support region is determined, the best planar fit of the pixels associated with each sequential subsection of this line (say in sections of $1/2$, $1/4$, $1/8$, \dots , depending upon its length), can be computed. In other words we are applying the same algorithm for computing the surface fit and line orientation of the whole support region to spatially contiguous subsections of the support region. This will give us the orientation of any line, curved or straight, up to the angular resolution of the subsections. If a linear feature is present, all por-

tions will produce approximately the same orientation; however, the spatial distribution of wavering, irregular, or smoothly curving convex or concave lines can be approximately determined. In the limit, the gradient direction at every point on the representative line of the support region can be computed. Rather than carrying out this computation for every support region, this could be done for selected regions, e.g., those whose variance is greater than some threshold. Unfortunately, this approach was not programmed and no results using this technique are presented. However, if the reader examines particular examples in the following section, it should be obvious that the nonlinear portions of some of the lines should be detectable by this technique.

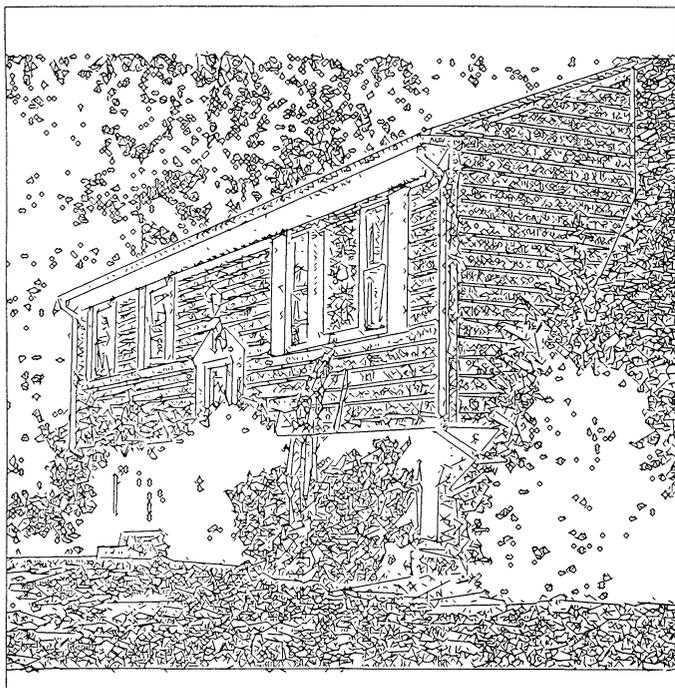
IV. EXPERIMENTAL RESULTS

A. Basic Line Extraction and Filtering Strategies

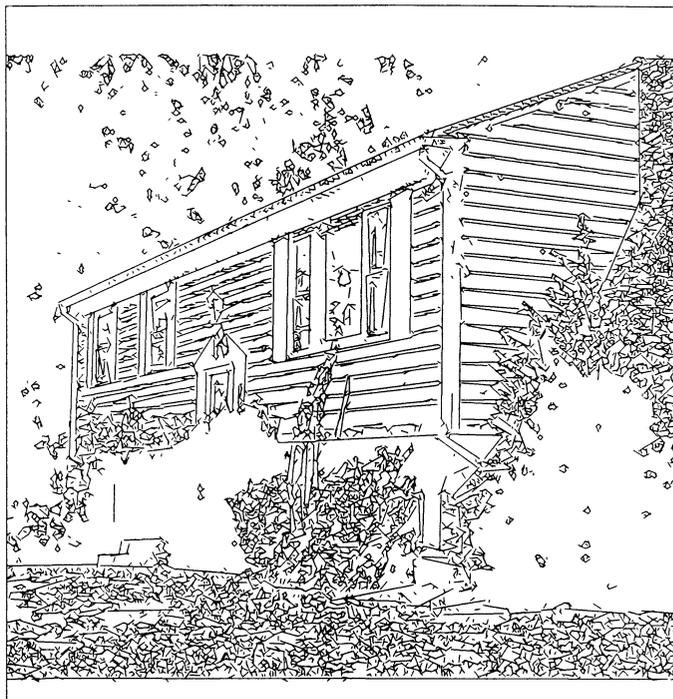
The line and feature extraction algorithms described in the preceding sections were applied to the full images shown in Fig. 1. The algorithm utilized overlapping partitions as described in Section II-B; the partition size was 45 degrees, rotated by 22.5 degrees. Figs. 11–14 demonstrate the performance of the algorithm on a variety of image types. In each case, the first image contains all the lines produced by both partitions that have a support greater than 0.5. This means that each line selected must have at least 50 percent of the pixels in its line-support region to choose it over the line generated from the other partitioning. The decision is based on which region generated the longest line, as described in Section II-B. Notice that this filter-step reduces the redundancy of representation of the lines in the image caused by the overlapping partitions and the resulting dual representation of a line.

Subsequent portions of Figs. 11–14 demonstrate the application of different filters applied to the initial data; these filter on the properties of line length, gradient steepness, line orientation, and image position measured in pixel coordinates. The “steepness” of the gradient was taken to be the slope of the plane fit to the intensity surface, the contrast as the difference between Max and Min values in the line-support region, and the width as contrast divided by steepness, as described in Section III.

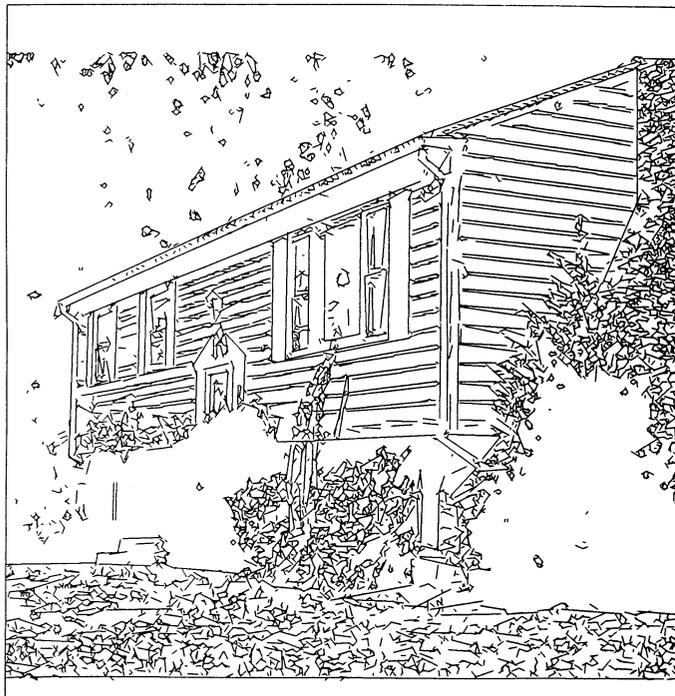
Fig. 11 shows results on a typical house image. The white areas in 11(a) are areas of the original image that are very dark and have little edge activity. Fig. 11(c) was formed from 11(a) by selecting those lines that have a length greater than 5 or a gradient steepness of greater than 10 gray levels per pixel. Clearly, this type of filtering does a reasonable job of removing lines that are due primarily to low-contrast noise and some textural edges. Filtering on the basis of gradient slope, or steepness, alone [Fig. 11(b) and 11(d)] or length alone [Fig. 11(e)] results in different classes of line events being retained. Fig. 11(e), for example, represents long lines of some significance; in this case many of the structural lines in the house are isolated. Fig. 11(f)–(h) demonstrates the effects



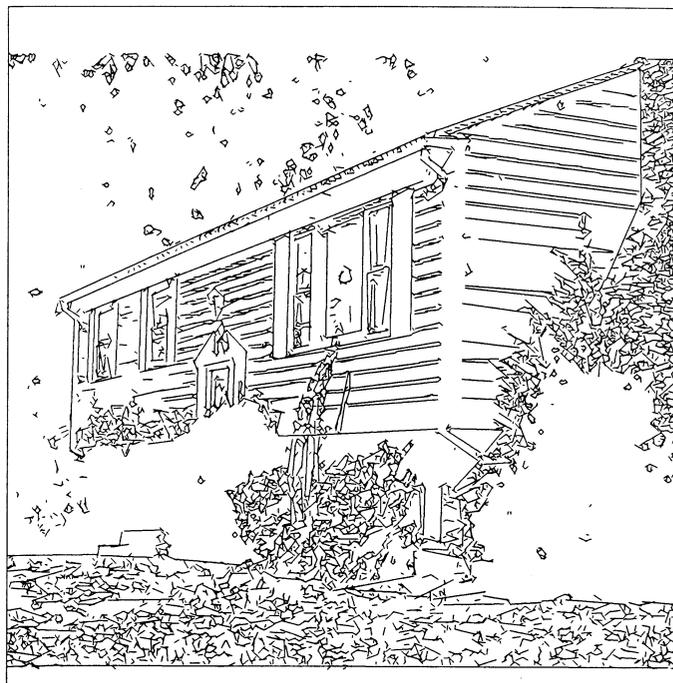
(a)



(b)

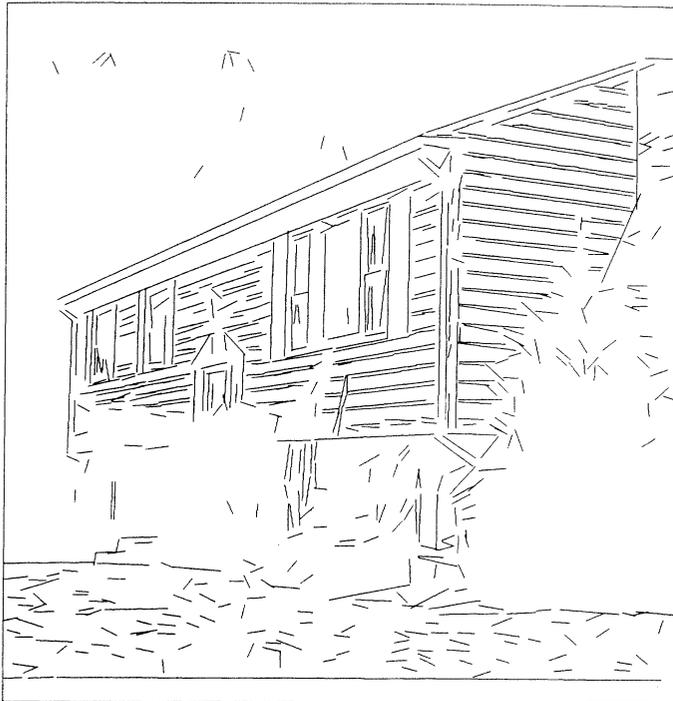


(c)

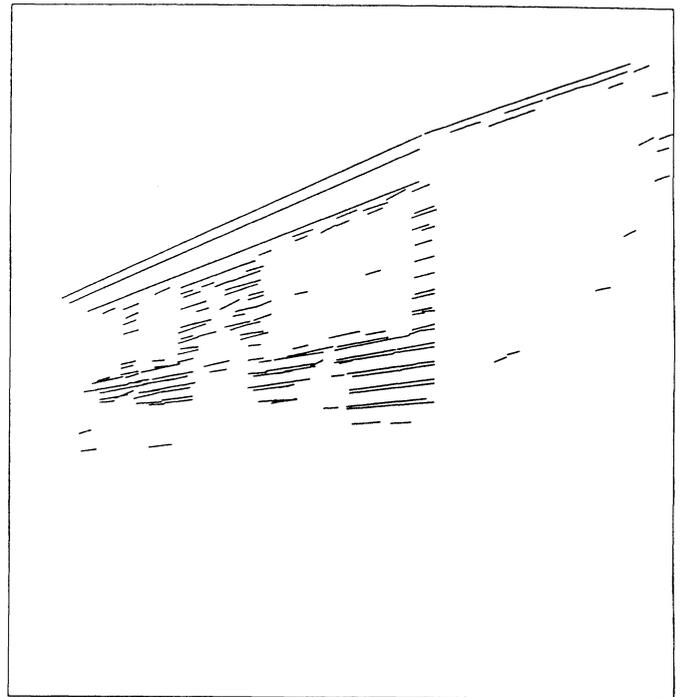


(d)

Fig. 11. Line extraction and filtering on the house image of Fig. 1(a). (a) Initial line set (with support ≥ 0.5). (b) Gradient steepness ≥ 2.5 gray levels per pixel. (c) Line length ≥ 5 OR gradient steepness ≥ 10 gray levels per pixel. (d) Steepness ≥ 10 gray levels per pixel.



(e)

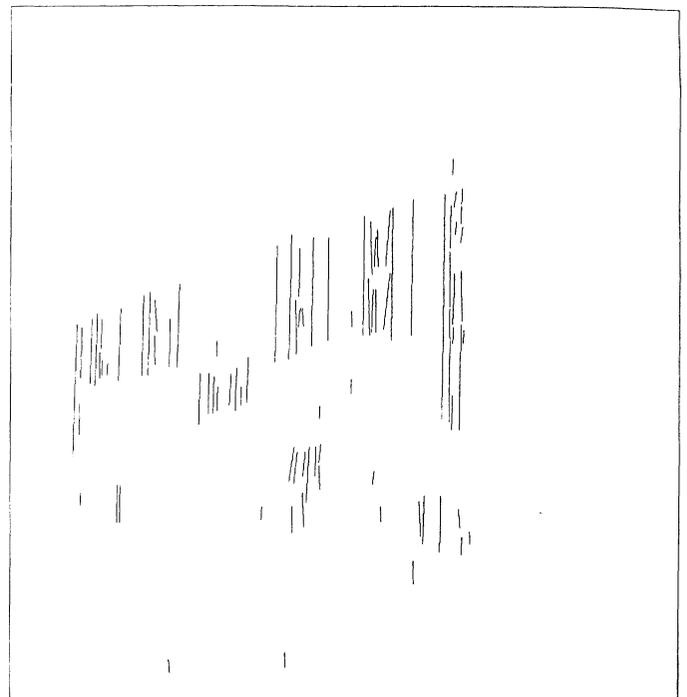


RANGE IN $\rho_0 = (-73.742 \ 28.362)$, RANGE IN $\theta = (1.08 \ 1.522)$

(f)



(g)



(h)

Fig. 11. (Continued.) (e) Length ≥ 5 . (f)-(h) Results of filtering on orientation and image location. (f) Orientation in range 3-28 degrees. (g) Orientation in range 165-177 degrees. (h) Orientation in range 81-95 degrees.

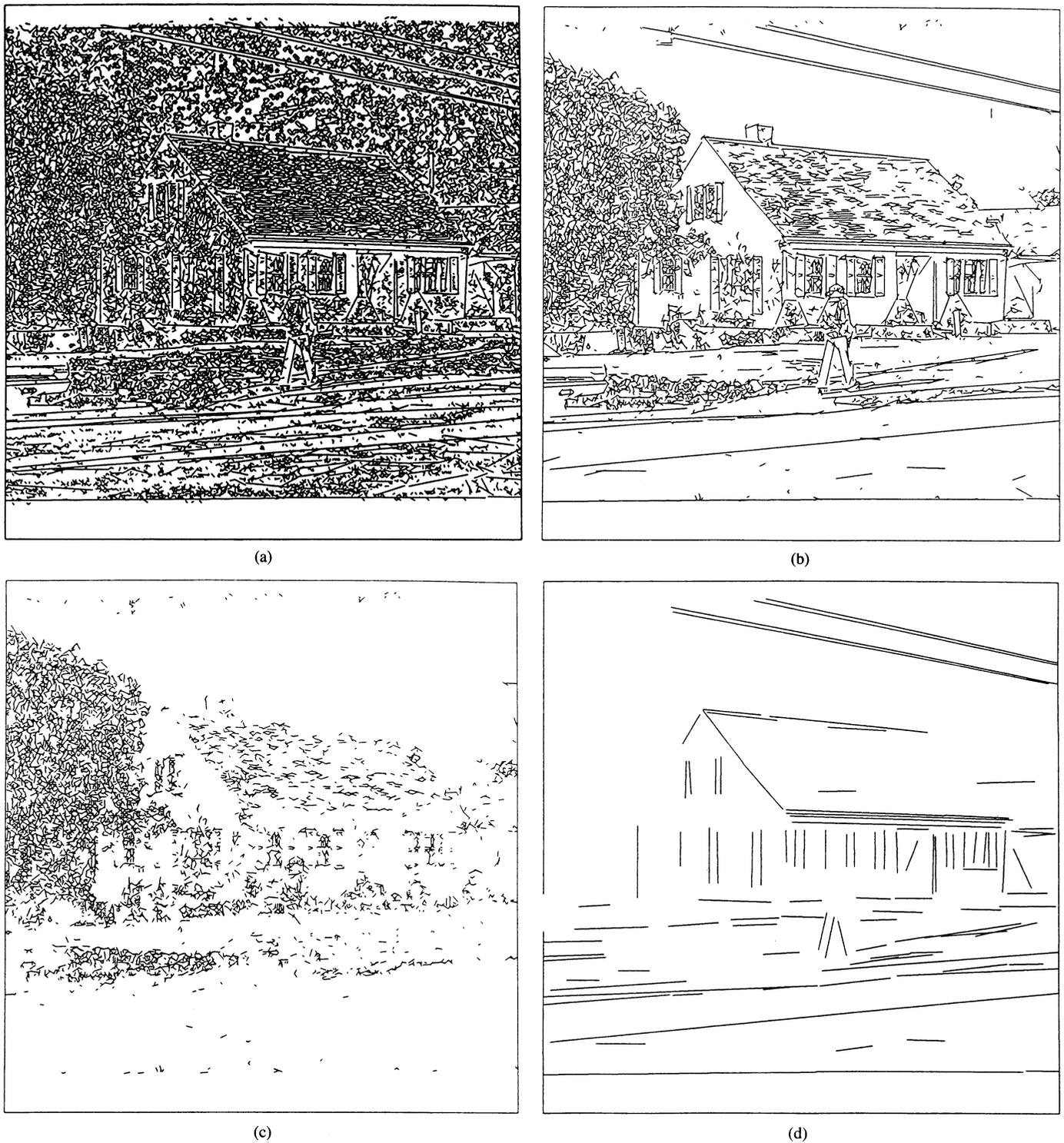


Fig. 12. Line extraction and filtering on the house image of Fig. 1(b). (a) Initial line set (with support ≥ 0.5). (b) Gradient steepness ≥ 10 gray levels per pixel. (c) Short steep lines (length ≤ 5 , gradient steepness ≥ 10 gray levels per pixel). (d) Steep, long lines (length ≥ 15 and gradient steepness ≥ 10 gray levels per pixel).

of filtering to produce lines for vanishing points and perspective analysis; they are obtained by filtering on line orientation and lateral image position at ranges in angular orientation of 3–28 degrees [Fig. 11(f)], 165–177 degrees [Fig. 11(g)], and 81–95 degrees [Fig. 11(h)]. A variety of other filters are possible and we are now exploring the

implications of various types of filters and the nature of the edge and line events they select. An interesting extension concerns dynamic generation of filters based on knowledge of the type of object and/or line-event sought.

Fig. 12(b) shows the results of filtering Fig. 12(a) on the basis of gradient steepness alone and thus should be

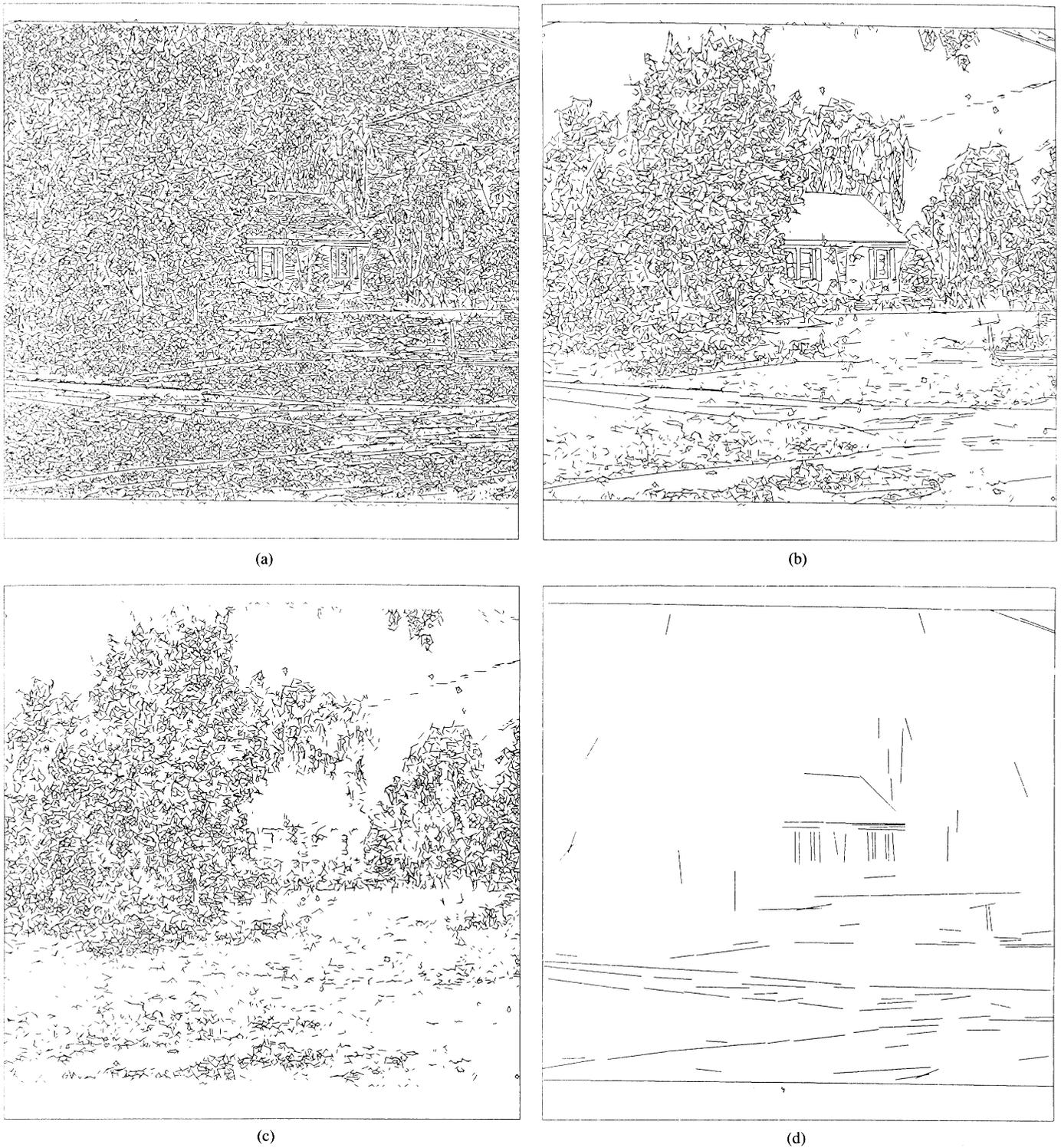


Fig. 13. Line extraction and filtering on the house image of Fig. 1(c). (a) Initial line set (with support ≥ 0.5). (b) Gradient steepness ≥ 10 gray levels per pixel. (c) Steep short lines (steepness ≥ 10 gray levels per pixel and length ≤ 5). (d) Filtering for steep long lines (steepness ≥ 10 gray levels per pixel and length ≥ 10).

compared to Fig. 11(c). Of the lines remaining in Fig. 12(b), filtering on length results in the separation of the lines into two disjoint sets, one corresponding to short texture edges [Fig. 12(c)] and the other to longer lines related to the macro structure of the image [Fig. 12(d)]. We are examining ways in which texture descriptors may

be constructed from the line-set remaining when a filter similar to that which produced Fig. 12(c) is applied to the initial low-level line representation. For example, it should be possible to separate this line set into at least two classes based on orientation or the lack of a preferred orientation; this would result in the extraction of the roof

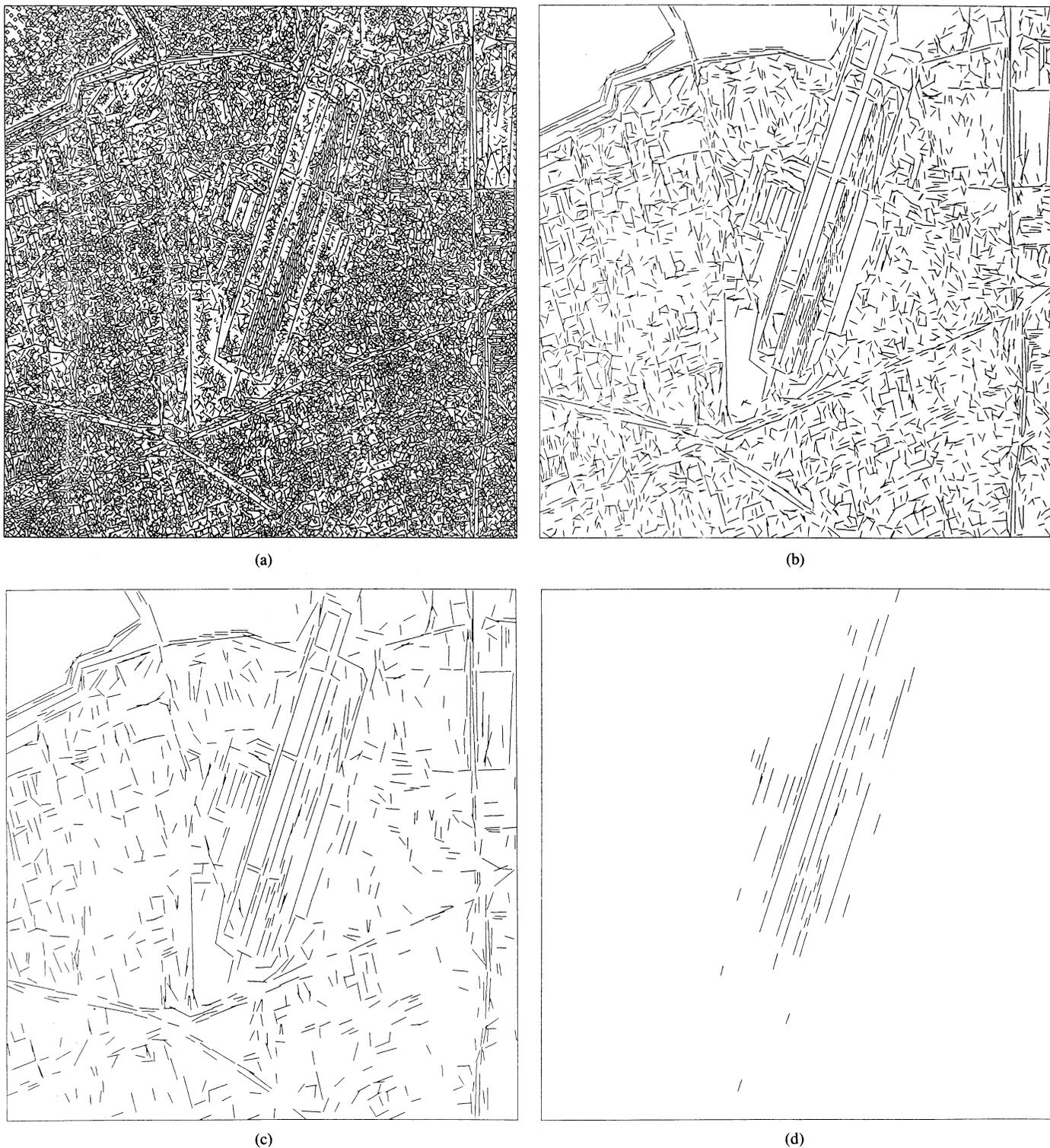
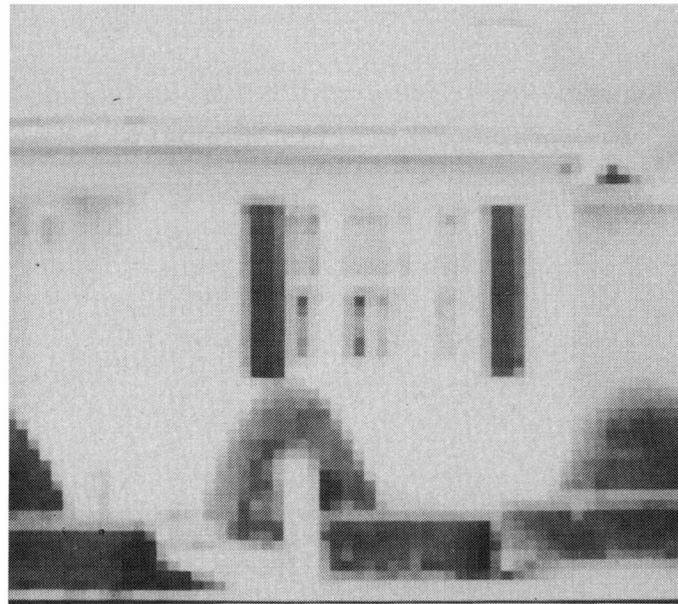


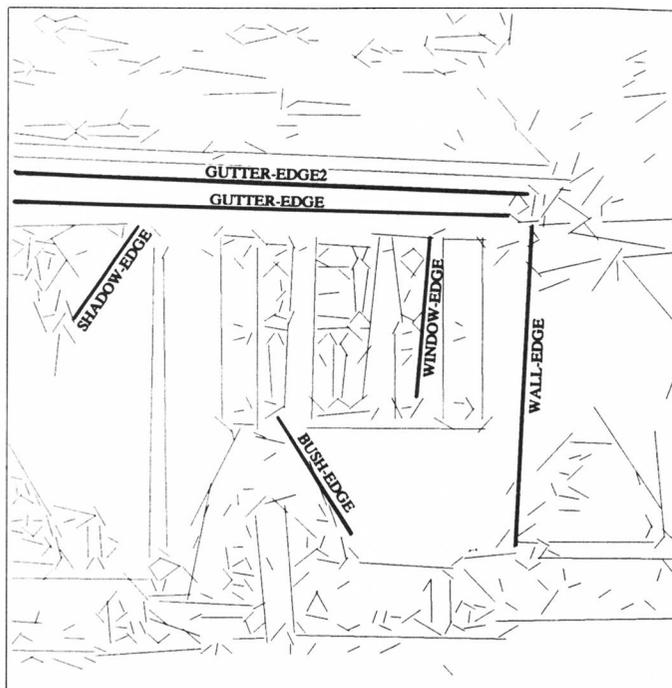
Fig. 14. Results of line extraction and filtering on the aerial image of Fig. 1(d). (a) Initial line set (with support ≥ 0.5). (b) Length ≥ 3 and gradient steepness ≥ 10 gray levels per pixel. (c) Same as (b) but length ≥ 5 . (d) Filtering on orientation (70–76 degrees) and image location.

texture (horizontally oriented) and the foliage texture (no preferred orientation). In Fig. 12(d), the structural edges representing the telephone wires were extracted from a thin one-pixel-wide diagonally-oriented region, a very difficult problem for many line extraction processes. Fig. 13 provides similar results for a different house image.

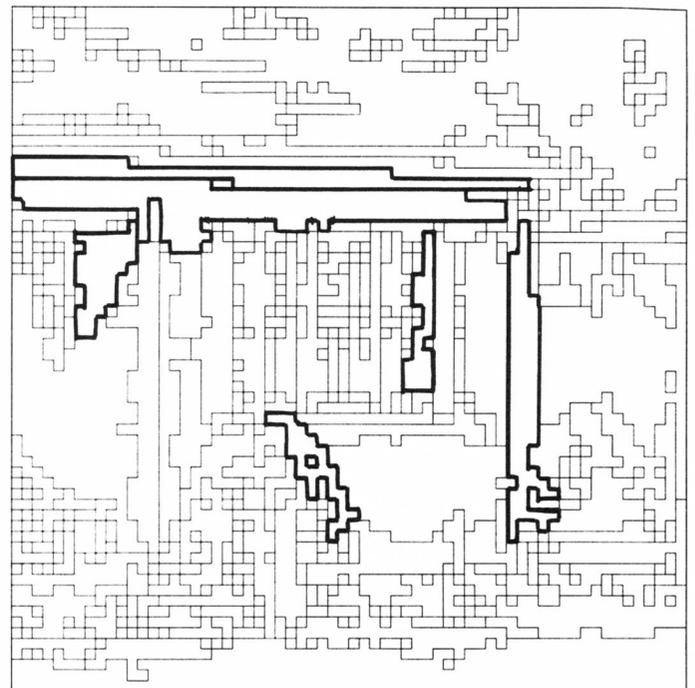
Fig. 14 illustrates results on an aerial image. Fig. 14(b) and (c) differ only in the line-length parameter of the filter. Fig. 14(d) is the result of filtering on angular orientation (70–76 degrees) and image position. This figure, as well as Fig. 11(f)–(h), were actually produced by hand-selecting local peaks in a (rho-theta) Hough transform of



(a)



(b)



(c)

Fig. 15. A subwindow of the house image in Fig. 1(b) used to explore a variety of lines in more detail. (a) The subwindow. (b) Straight lines after filtering on steepness ≥ 10.0 ; the lines selected for further analysis have been emphasized and labeled. (c) Corresponding gradient-support regions.

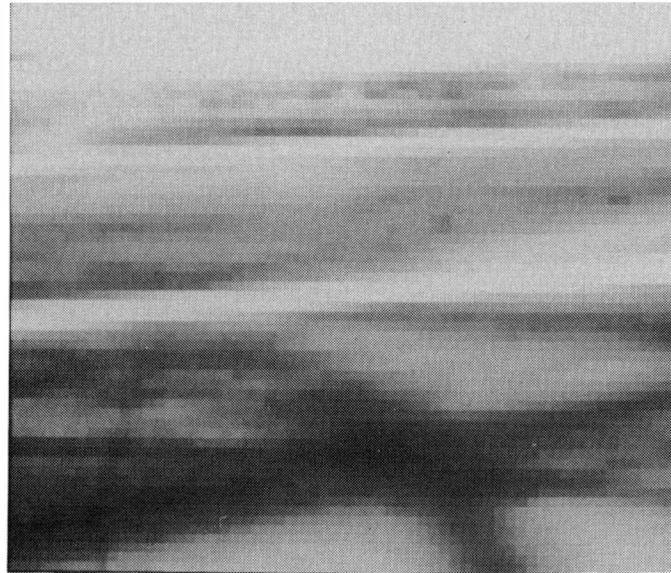
the original line set. Automatic peak selection methods, such as those developed in [17], [27], will be examined in future work.

B. Specific Examples with Attributes

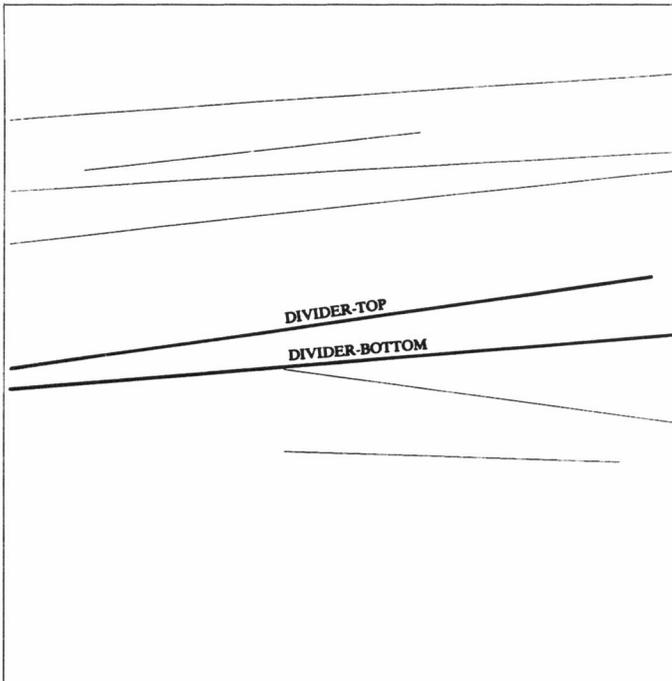
Figs. 15(a) and 16(a) are two subimages of Fig. 1(b), from which particular examples of lines have been selected. In each figure, (b) shows the lines selected and (c)

shows the corresponding line-support regions. The line set includes examples of lines that are truly straight, lines that, for a variety of reasons, are not actually straight, and lines whose placement has been distorted by anomalies in the intensity-surface data.

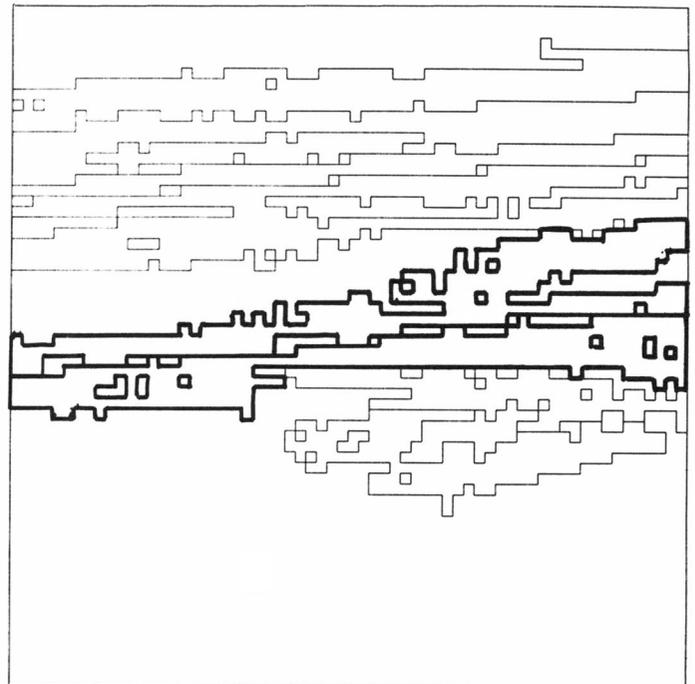
Figs. 17-22 provide a more detailed analysis of the behavior of the intensity surface in the support region and provide descriptor data for each line. In each figure, (a)



(a)



(b)



(c)

Fig. 16. A subwindow from Fig. 1(b) containing a road fragment. (a) The subwindow. (b) Straight lines after filtering on length ≥ 30.0 ; the lines selected for further analysis have been emphasized and labeled. (c) Corresponding gradient-support regions.

shows the local gradient vectors overlaid on the support region, (b) is a histogram of the local gradient orientation, (c) shows the resulting straight line superimposed on the intensity region corresponding to the line-support region as well as the isocontour generated by the intersection of the intensity surface with a plane whose altitude is equal to the average intensity, and (d) shows the full feature descriptor for each line. Fig. 17 has been included as a

basis of comparison for the remaining figures and as a demonstration that the curvilinear nature of the true line can be detected from the attributes associated with the extracted straight line.

The feature descriptor forms the basis of the line data base for an image, and over which all of the filtering operations described earlier were performed. The manner in which the values of many of these features are computed

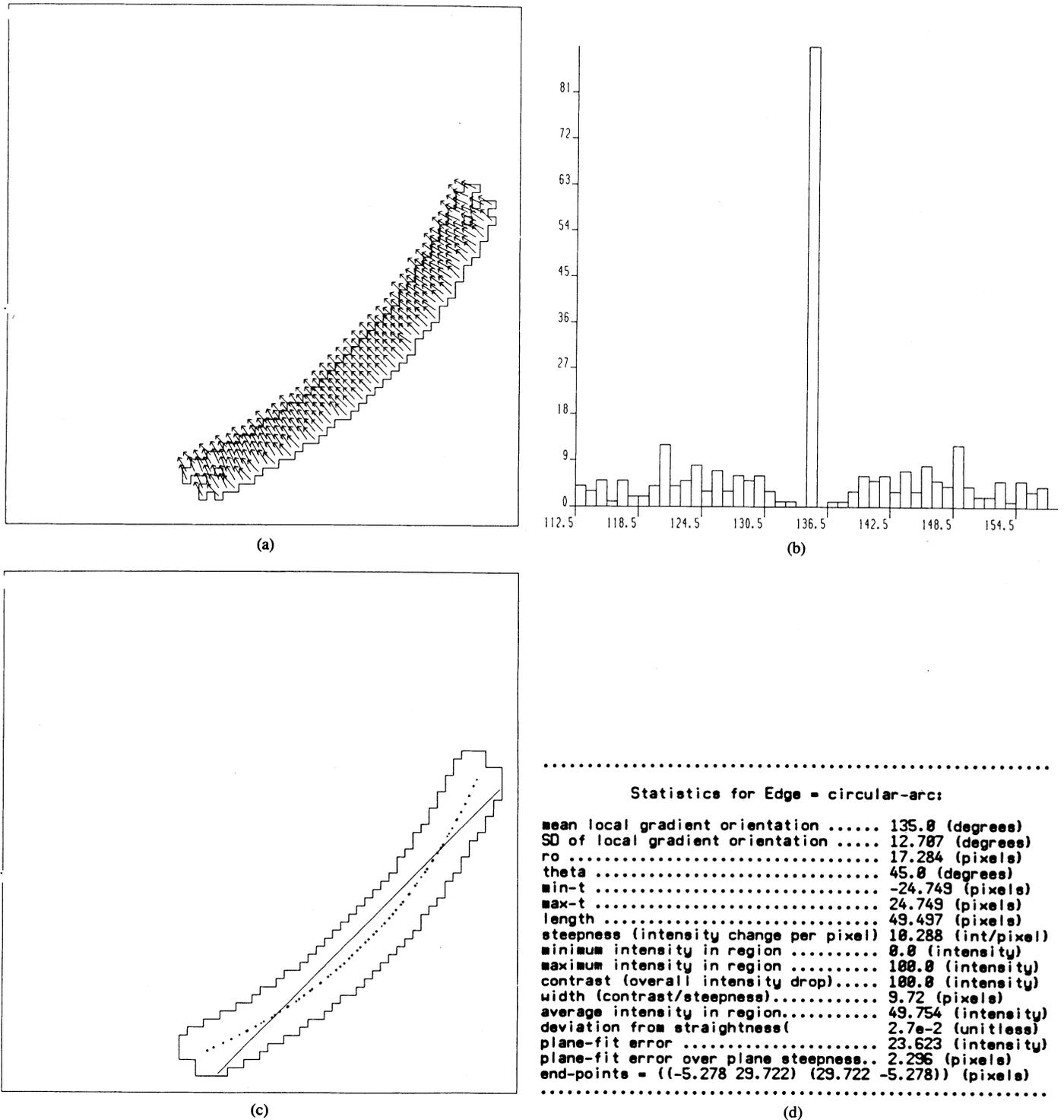
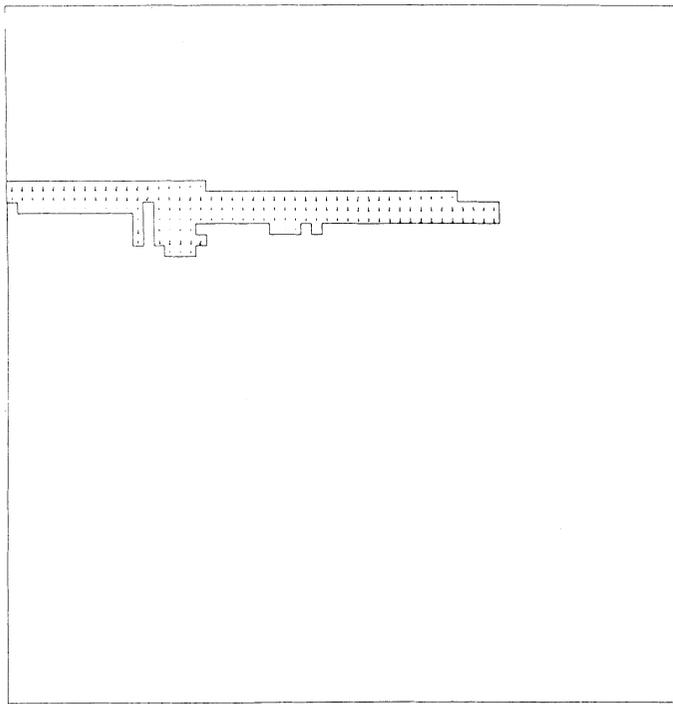


Fig. 17. A synthetic circular arc. (a) Associated gradient-support region. (b) Local gradient-orientation histogram of region. (c) Interpreted straight-line, region of pixels used in plane-fit and iso-contour used in straightness test. (d) Statistics for region and straight line.

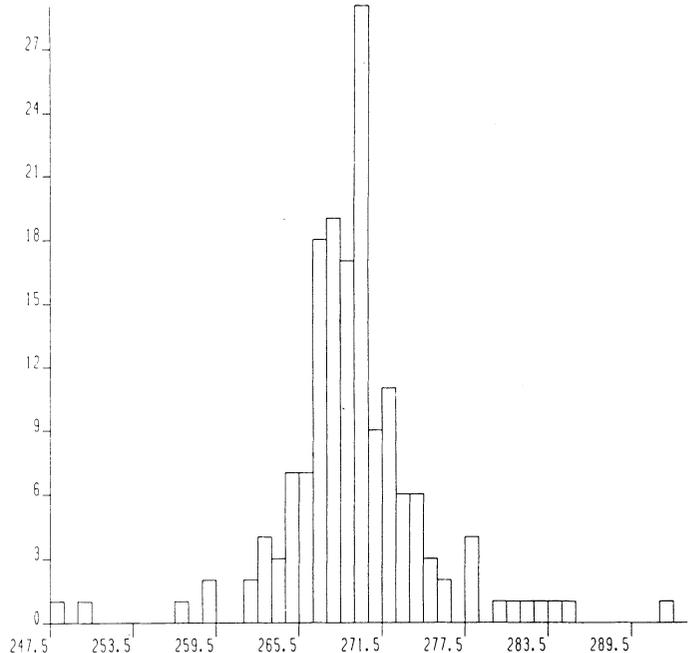
has been described in earlier sections, particularly Sections II-C and III. Theta is the angle between the horizontal (x -) image axis and the perpendicular to the line. Min- t , max- t are the coordinates of the endpoints of the line in a parameterized representation formed by rotating the x - y picture coordinate frame theta degrees with respect

to the x axis. In the new (ρ, t) frame, ρ is the distance from the origin to the line and t is the coordinate of points on the line [5].

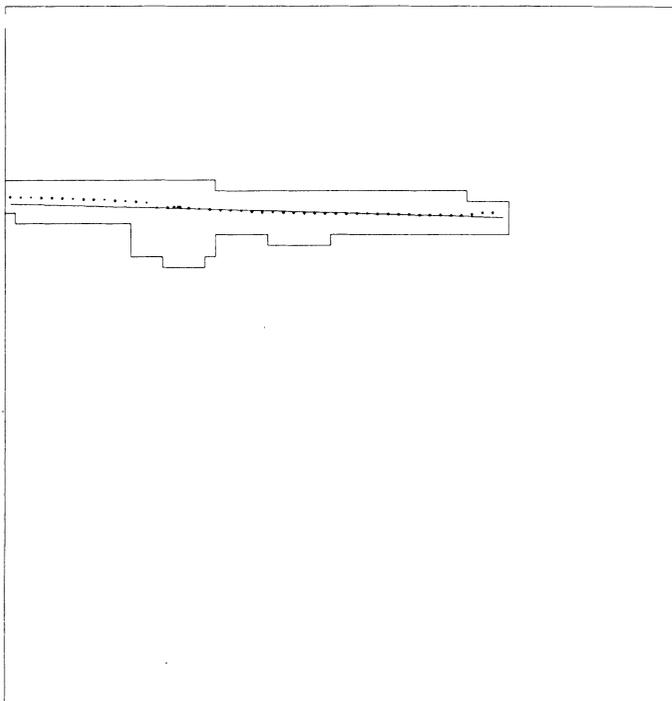
The set of lines has been chosen in order to demonstrate the effectiveness of the algorithm, as well as to point out some of the known limitations and remaining problems.



(a)



(b)



(c)

.....

Statistics for Edge = gutter-edge:

mean local gradient orientation	269.075 (degrees)
SD of local gradient orientation	4.688 (degrees)
ro	12.959 (pixels)
theta	-88.414 (degrees)
min-t	-15.148 (pixels)
max-t	31.862 (pixels)
length	47.01 (pixels)
steepness (intensity change per pixel)	17.918 (int/pixel)
minimum intensity in region	108.0 (intensity)
maximum intensity in region	236.0 (intensity)
contrast (overall intensity drop).....	128.0 (intensity)
width (contrast/steepness).....	7.144 (pixels)
average intensity in region.....	159.609 (intensity)
deviation from straightness(5.e-3 (unitless)
plane-fit error	6.9 (intensity)
plane-fit error over plane steepness..	0.385 (pixels)
end-points = ((15.501 -12.534) (-31.491	-13.836)) (pixels)

.....

(d)

Fig. 18. "Gutter-edge" from Fig. 15. (a) Associated gradient-support region. (b) Local-gradient-orientation histogram of region. (c) Interpreted straight line, region of pixels used in plane-fit and iso-contour used in straightness test. (d) Statistics for region and straight-line.

The reader is invited to examine and compare the absolute and relative values of the attributes and to qualitatively assess their effectiveness. In the next section, we will discuss particular remaining weaknesses of our approach and outline possible directions for future research.

C. Discussion of Specific Classes of Problems

In this section, we discuss some of the specific problems with the algorithm that are known to exist and that we are continuing to examine. In general, the algorithm

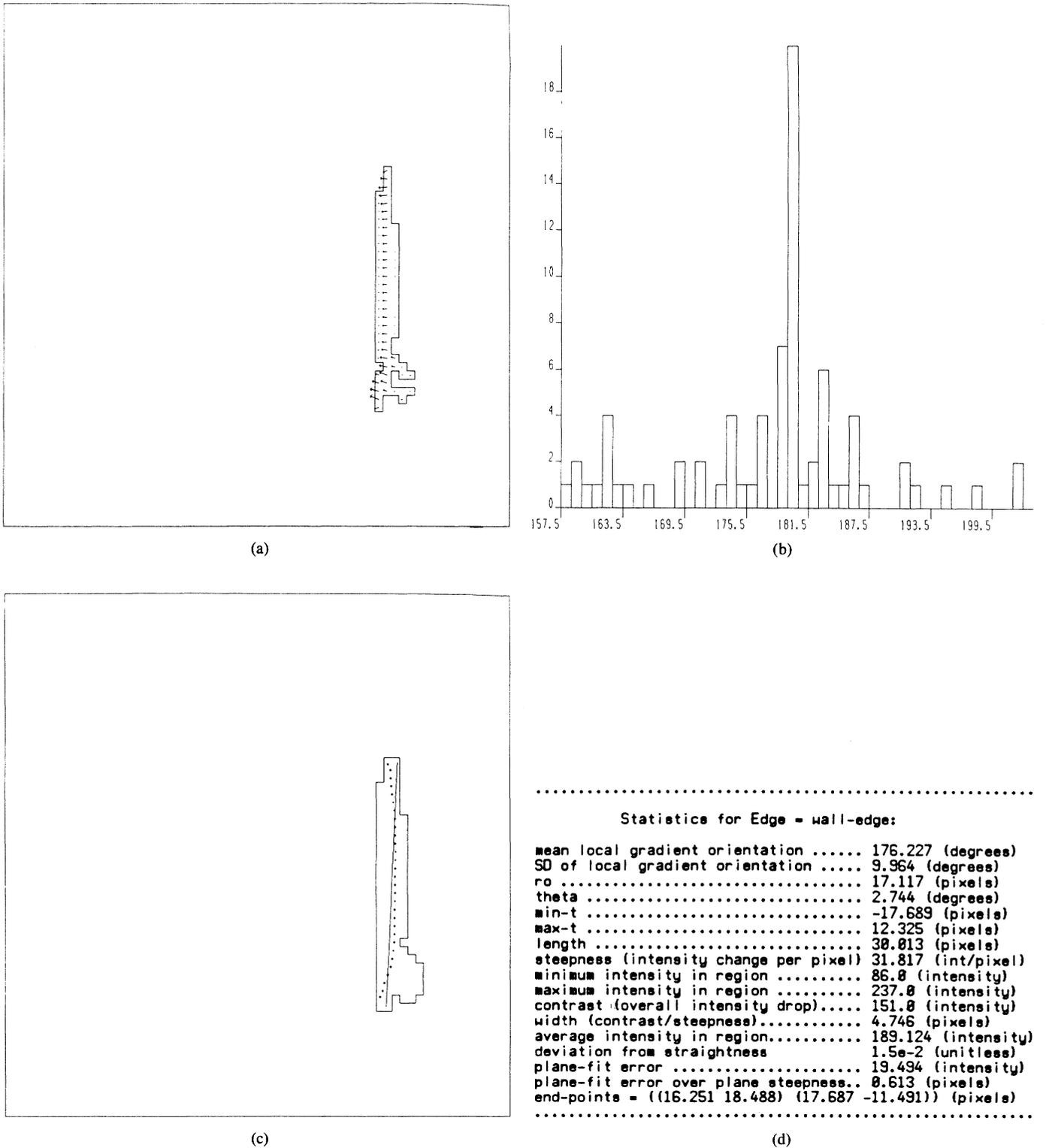


Fig. 19. "Wall-edge" from Fig. 15. (a) Associated gradient-support region. (b) Local gradient orientation histogram of region. (c) Interpreted straight line, region of pixels used in plane-fit and iso-contour used in straightness test. (d) Statistics for region and straight line.

will group pixels into a line-support region when the local gradient-orientation estimates are sufficiently similar. There are several cases where this may not produce desirable results, including:

- 1) a low-magnitude gradient that does not represent a real line;
- 2) distortion of the location or orientation of a line due to grouping errors;

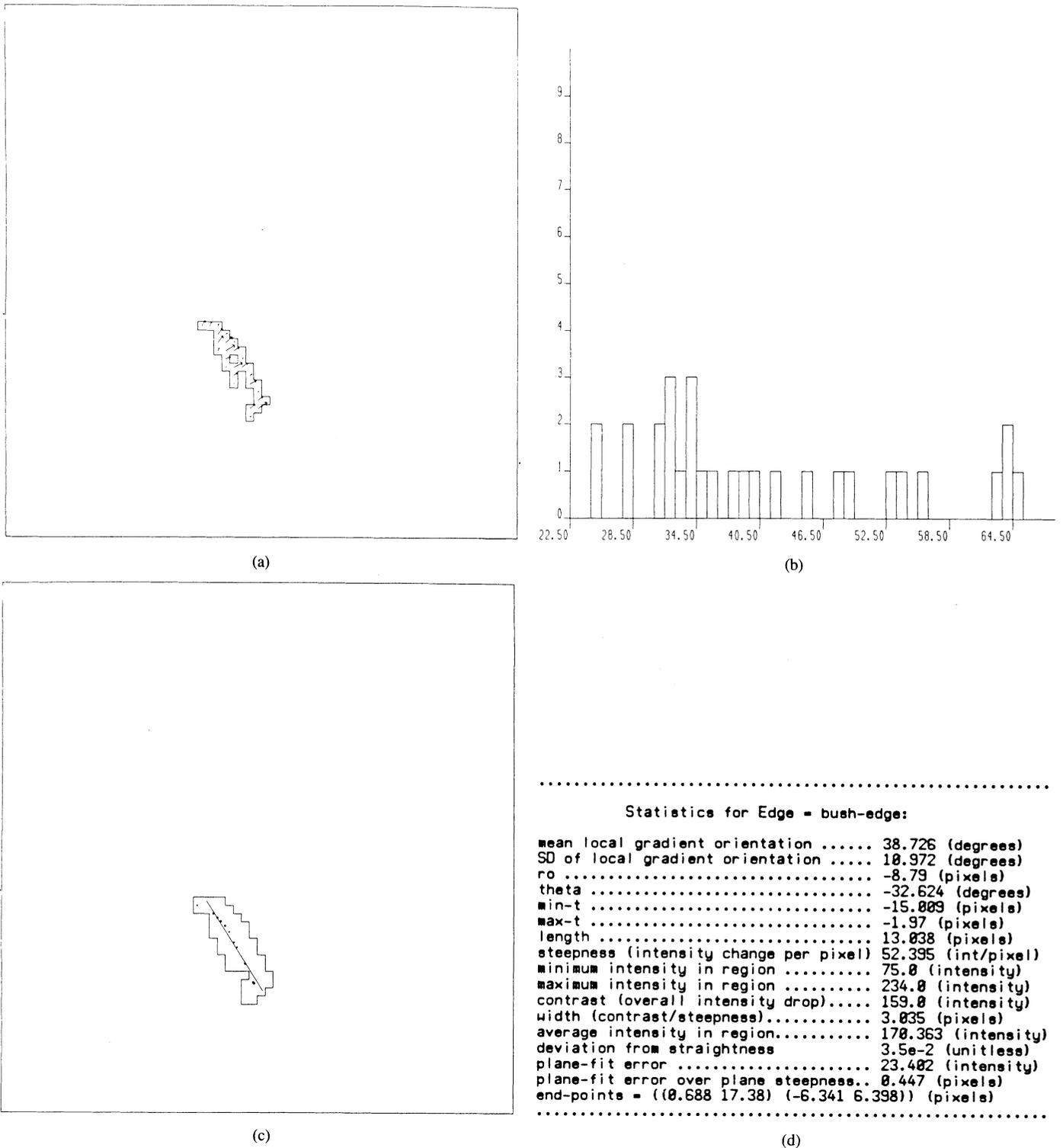


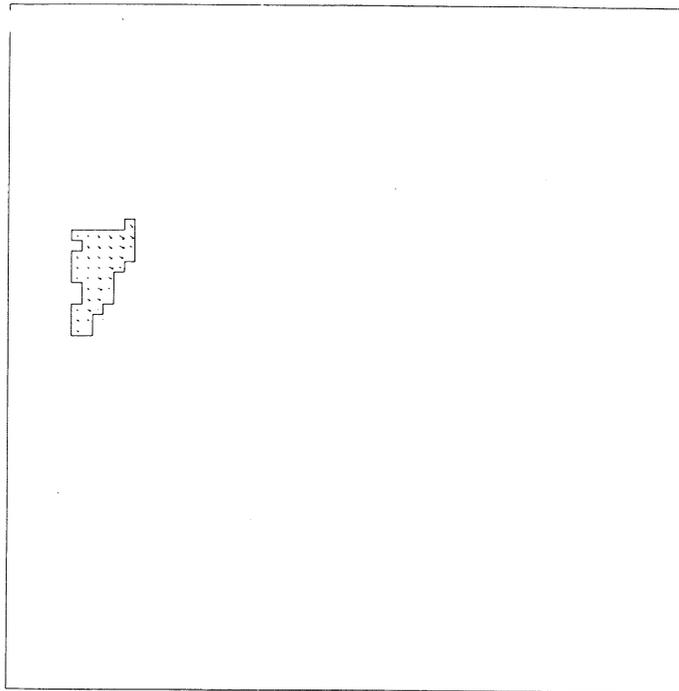
Fig. 20. "Bush-edge" from Fig. 15. (a) Associated gradient-support region. (b) Local gradient-orientation histogram of region. (c) Interpreted straight line, region of pixels used in plane-fit and iso-contour used in straightness test. (d) Statistics for region and straight line.

- 3) a curved line being approximated as a straight line;
- 4) multiple straight lines being grouped into a single line.

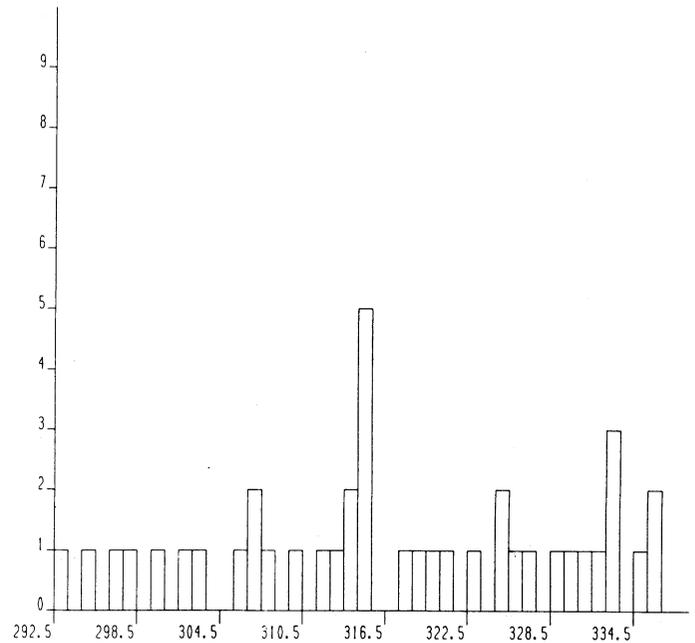
In this section, we will briefly consider each of these problems. Our approach in the next section will be to examine measures for detecting these problems so that such

line-support regions can be processed more carefully and accurately.

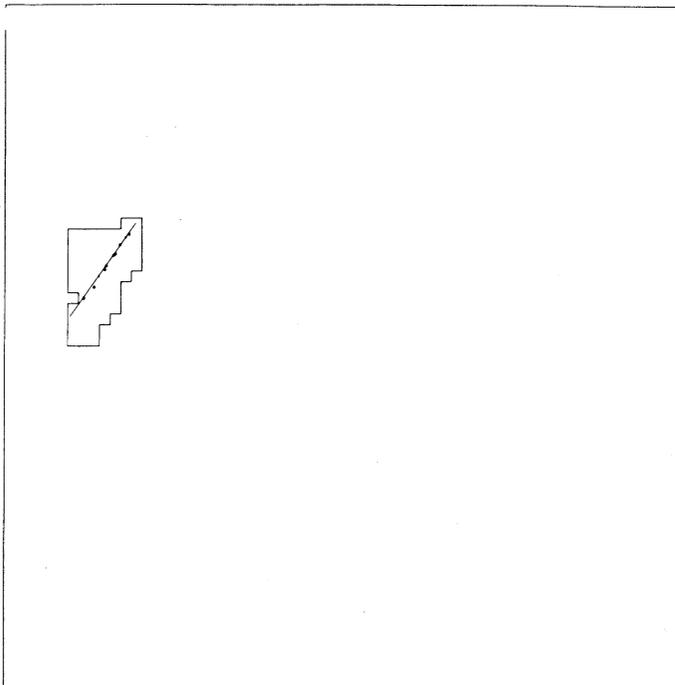
There are obvious situations where the algorithm extracts low-contrast low-steepness lines that are not meaningful. In Fig. 12(a), the sky is filled with short lines from small line-support regions produced by very minor inten-



(a)



(b)



(c)

.....

Statistics for Edge = shadow-edge:

mean local gradient orientation	319.582 (degrees)
SD of local gradient orientation	11.815 (degrees)
ro	-22.634 (pixels)
theta	35.374 (degrees)
min-t	-12.724 (pixels)
max-t	-1.912 (pixels)
length	18.812 (pixels)
steepness (intensity change per pixel)	28.953 (int/pixel)
minimum intensity in region	121.0 (intensity)
maximum intensity in region	248.0 (intensity)
contrast (overall intensity drop).....	119.0 (intensity)
width (contrast/steepness).....	5.679 (pixels)
average intensity in region.....	197.391 (intensity)
deviation from straightness	8.e-3 (unitless)
plane-fit error	6.421 (intensity)
plane-fit error over plane steepness..	0.386 (pixels)
end-points = ((-25.822 -2.728) (-19.562 -11.544))	(pixels)

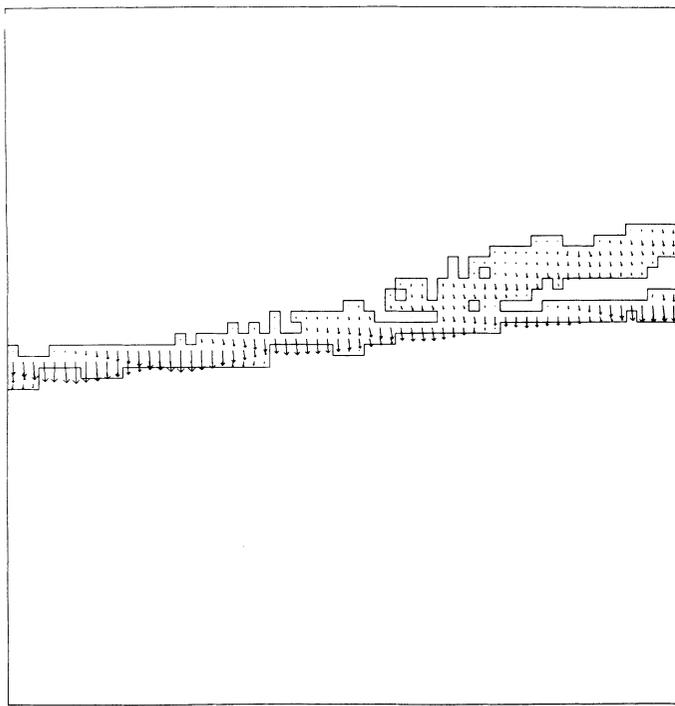
.....

(d)

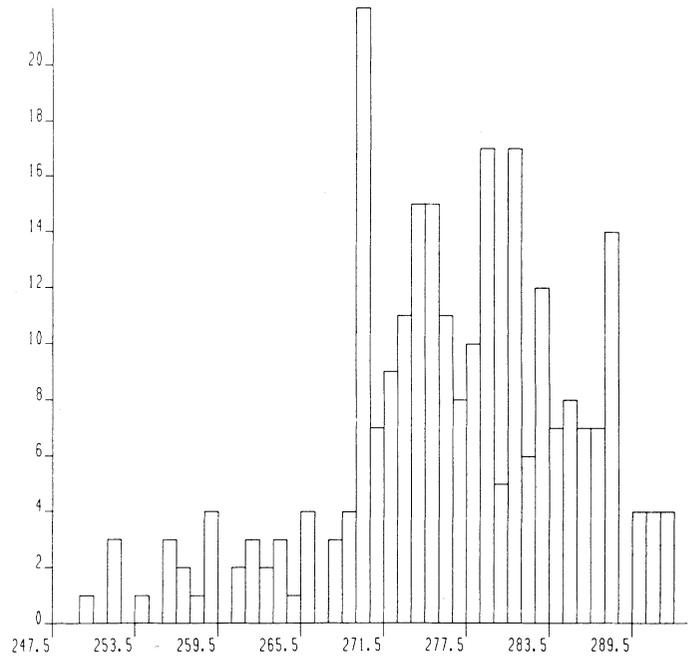
Fig. 21. "Shadow edge" from Fig. 15. (a) Associated gradient-support region. (b) Local gradient-orientation histogram of region. (c) Interpreted straight line, region of pixels used in plane fit and iso-contour used in straightness test. (d) Statistics for region and straight line.

sity changes, possibly at the noise level. In other cases, a very wide low-magnitude gradient will produce a large support region that should really be viewed as a homogeneous region without any associated line. For example, if a planar surface of uniform reflectance oriented at an angle to the viewer is illuminated, then a uniform lighting

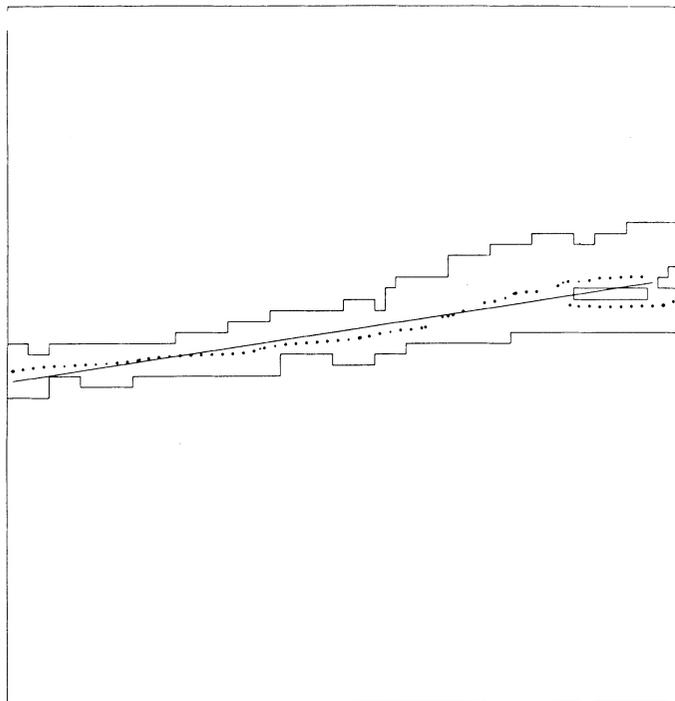
gradient is obtained over the surface. The algorithm would group all the pixels associated with this gradient into a line-support region and extract a straight-line interior to the surface boundaries. Although this line can be removed by the filtering processes discussed in the next section, in some cases it may be necessary to examine the data in the



(a)



(b)



(c)

.....

Statistics for Edge = divider-top:

mean local gradient orientation	275.897 (degrees)
SD of local gradient orientation	6.33 (degrees)
ro	-2.145 (pixels)
theta	81.616 (degrees)
min-t	-31.528 (pixels)
max-t	30.132 (pixels)
length	61.66 (pixels)
steepness (intensity change per pixel)	12.753 (int/pixel)
minimum intensity in region	75.0 (intensity)
maximum intensity in region	187.0 (intensity)
contrast (overall intensity drop).....	112.0 (intensity)
width (contrast/steepness).....	8.782 (pixels)
average intensity in region.....	124.664 (intensity)
deviation from straightness	1.4e-2 (unitless)
plane-fit error	15.282 (intensity)
plane-fit error over plane steepness..	1.198 (pixels)
end-points = ((-31.504 2.475) (29.498 -6.515))	(pixels)

.....

(d)

Fig. 22. "Divider-top" from Fig. 16. (a) Associated gradient-support region. (b) Local gradient-orientation histogram of region. (c) Interpreted straight line, region of pixels used in plane-fit and iso-contour used in straightness test. (d) Statistics for region and straight line.

line-support region in detail to determine whether the line corresponds to a real line or to a widely distributed gradient.

Places in the algorithm where homogeneous regions can be detected are: 1) immediately after the computation of the gradient, low-magnitude values can be grouped; this

is the most efficient, but suffers from the danger of losing portions of true line-support regions; 2) after the plane fit, but before further extraction of attribute values—this allows global information via low slope of the planar fit of the intensity surface to be utilized; 3) after the entire low-level representation has been computed, which allows all

TABLE I
COMPARISON OF ERROR FILTERING ATTRIBUTES

Figure	Line Name	SD of Gradient Orientation (degrees)	Deviation From Straightness (times 1000)	Plane Fit Error (intensity levels)	Plane Fit Error Over Plane Steepness (pixels)	Actually a Straight Line	Location and Orientation of Line
17	Synthetic Circular Arc	12.7	27	23.6	2.30	NO	BAD
18	Gutter-Edge	4.7	5	6.9	.385	YES	GOOD
19	Gutter-Edge 2	1.9	2	9.6	.167	YES	GOOD
20	Wall-Edge	10.0	15	19.5	.613	PARTLY	BAD
21	Window Edge	9.7	10	8.9	.264	YES	FAIR
22	Bush Edge	11.0	35	23.4	.447	NO	FAIR
23	Shadow Edge	11.8	8	6.4	.306	NO	GOOD
24	Yellow Road Divider Top	6.3	14	15.3	1.20	YES	BAD
25	Yellow Road Divider Bottom	5.2	3	12.8	.56	YES	GOOD

of the line attributes that are available to be employed. Probably the latter case would be the most general because perception of homogeneity is undoubtedly affected by several factors including width, steepness, and contrast.

A more difficult related case occurs when a low-gradient magnitude region is incorrectly grouped with a correct line-support region. When a slow gradient is combined with a sharper change, because of similar orientations, the effect is to skew the placement of the resulting line. Several examples of this can be seen in Fig. 11(b), where the lines corresponding to some of the siding boundaries on the side of the house are not parallel as would be expected. A second extreme example is shown in Fig. 22; the line-support region corresponds to a section at the right end of the line bordering the top of a yellow stripe down the center of the road in Fig. 1(b). The line-support region has a "bulge" on the right end of the line that results in the line being skewed from the position it would normally be placed given the data from the rest of the region.

The next type of problem is obvious and involves detecting line-support regions that are actually associated with curved lines. The degree of possible curvature, of course, is related to the size of the orientation partitions. However, we do not want these partitions too small or else there will be significant undesirable fragmentation.

Finally, we would like to detect two distinct straight lines with similar gradient orientation that have been grouped. If such a situation is detected, such lines can be processed more carefully.

D. Detection of Errors

As a first step we have constructed Table I, listing the set of lines and just the attributes that measure some form of *error* or *straightness*, so that their potential for focusing attention and filtering mistakes can be qualitatively judged. First, note that the three lines that are listed in the table as actually being straight lines and that are qualitatively judged to be located and oriented properly have the three lowest standard deviations (variance) for orien-

tation and the three lowest straightness deviation measures. Thus, there is strong promise for their utility. The line with the highest smooth curvature, the circular arc, produced the highest value of orientation variance and the second largest value in the nonstraightness measure. High values of plane-fit error and plane-fit error/steepness also show the utility of these measures for detecting anomalies. In particular, the difficulties produced by the top yellow-road divider breaking into two gradient-region arms is detected by the second highest plane-error/steepness. The curved line of the bush is detected by the second highest standard deviation or orientation, the highest non-straightness value, and a plane-fit error approaching that of the synthetic arc. We believe these measures, as well as other possibilities, will allow us to detect many of the mistakes that the algorithm could potentially generate.

V. CONCLUSIONS

This paper has presented a novel method for extracting straight lines and a general low-level representation of straight lines that can be used for a variety of purposes. The technique for finding straight lines is effective because it globally organizes the spatial extent of a straight line without local decisions about the meaningfulness of an edge event. It does this by utilizing gradient orientation to provide a gradient segmentation of the pixels in the formation of line-support regions. Analysis of the intensity surface of the pixels in these regions yields the information required to extract lines and characterize the intensity variations in a variety of ways. The algorithm is very robust and accurately extracts many low-contrast long lines.

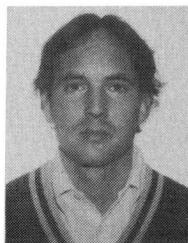
An additional contribution of this paper is the technique employed for segmentation of gradient orientation. It takes a very simple approach of partitioning via labels over fixed intervals of orientation, and extends it to a dual version with a second fixed-partition segmentation rotated a half interval; results of the two partitions are integrated by selecting lines from the two representations based upon a pixel voting scheme.

The low-level line-support representation has potential as the basis for intermediate processes for linking collinear line segments that were not extracted as a single line or that are spatially disconnected. It also should allow linking of piecewise-straight segments as an approximation of a curved line and mechanisms for extracting textured regions by treating the smaller lines as texture elements and grouping them (using such line attributes as length, orientation, contrast, and average intensity). Finally, the gradient regions associated with each line can be a very useful tool for integrating intensity-based image segmentations with the extracted lines [27].

REFERENCES

- [1] D. H. Ballard, "Generating the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, pp. 111-122, 1981.
- [2] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [3] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene

- characteristics from images," in *Computer Vision Systems*, A. Hanson and E. Riseman, Eds. New York: Academic, 1978, pp. 3-26.
- [4] J. F. Canny, "Finding edges and lines in images," Massachusetts Inst. Technol., AI Lab. Tech. Rep. 720, June 1983.
- [5] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
- [6] R. W. Ehrlich and J. P. Foith, "Topology and Semantics of Intensity Arrays," in *Computer Vision Systems*, A. Hanson and E. Riseman, Eds. New York: Academic, 1978, pp. 111-127.
- [7] M. Fischler and H. Wolf, "Linear delineation," in *Proc. CVPR*, 1983, pp. 351-356.
- [8] A. Hanson and E. Riseman, "Preprocessing cones: A computational structure for scene analysis," Univ. Massachusetts, COINS Tech. Rep. 74C-7, Sept. 1974.
- [9] A. Hanson and E. Riseman, "Segmentation of natural scenes," in *Computer Vision Systems*, A. Hanson and E. Riseman, Eds. New York: Academic, 1978, pp. 129-163.
- [10] A. Hanson, E. Riseman, and F. Glaser, "Edge relaxation and boundary continuity," Univ. Massachusetts, COINS Tech. Rep. 80-11, May 1980.
- [11] A. Hanson and E. Riseman, "Processing cones: A computational structure of image analysis," in *Structured Computer Vision*, S. Tamimoto and A. Klinger, Eds. New York: Academic, 1980. Also Univ. Massachusetts, COINS Tech. Rep. 81-38, Dec. 1981.
- [12] R. M. Haralick and L. Watson, "A facet model for image data," *Computer Graphics and Image Processing*, vol. 15, pp. 113-129, 1981.
- [13] R. M. Haralick, "Ridges and valleys on digital images," *Computer Vision Graphics and Image Processing*, vol. 22, no. 1, pp. 28-39, 1983.
- [14] T. Hong, M. Shneier, R. Hartley, and A. Rosenfeld, "Using pyramids to detect good continuation," *IEEE Trans Syst., Man, Cybern.*, vol. SMC-13, pp. 631-635, 1983.
- [15] B. K. P. Horn, "Understanding image intensities," *Artificial Intell.*, vol. 8, pp. 201-231, 1977.
- [16] M. D. Kelly, "Edge detection in pictures by computer using planning," *Machine Intell.*, vol. 6, pp. 397-409, 1971.
- [17] R. R. Kohler, "Integrating non-semantic knowledge into image segmentation processes," Ph.D. dissertation and COINS Tech. Rep. 84-04, Univ. Massachusetts, Sept. 1983.
- [18] S. Levialdi, "Finding the edge," in *Digital Image Processing*, S. C. Simon and R. M. Haralick, Eds. Dordrecht, The Netherlands: D. Reidel, 1981.
- [19] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*. New York: Academic, 1979.
- [20] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," in *Proc. Roy. Soc. London*, vol. B200, pp. 269-294, 1977.
- [21] D. Marr, *Vision*. San Francisco, CA: Freeman, 1982.
- [22] J. Montanari, "On the optimal detection of curves in noisy pictures," *Commun. ACM*, vol. 14, pp. 335-345, 1971.
- [23] R. Nevatia and K. Babu, "Linear feature extraction and description," *Computer Graphics and Image Processing*, vol. 13, pp. 257-269, 1980.
- [24] C. Parma, A. Hanson, and E. Riseman, "Experiments in schema-driven interpretation of a natural scene," Univ. Massachusetts, COINS Tech. Rep. 80-10, 1980. Also in *Nato Advanced Study Institute on Digital Image Processing*, R. Haralick and J. C. Simons, Eds. Bonas, France, 1980.
- [25] K. A. Paton, "Conic sections in automatic chromosome analysis," *Machine Intell.*, vol. 5, 1970.
- [26] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978.
- [27] G. Reynolds, N. Irwin, A. Hanson, and E. Riseman, "Hierarchical knowledge-directed object extraction using a combined region and line representation," in *Proc. Workshop on Computer Vision Representation and Control*, Annapolis, MD, Apr. 30-May 2, 1984, pp. 238-247.
- [28] A. Rosenfeld, "A non-linear edge detection technique," *Proc. IEEE*, vol. 58, pp. 814-816, May 1970.
- [29] Y. Shirai, "Recognition of real-world objects using edge cues," in *Computer Vision Systems* A. Hanson and E. Riseman, Eds. New York: Academic, 1978, pp. 353-362.
- [30] S. Zucker, R. Hummel, and A. Rosenfeld, "An application of relaxation labelling to line and curve enhancement," *IEEE Trans. Comput.*, vol. C-26, no. 4, Apr. 1977.



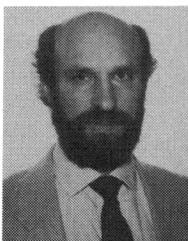
J. Brian Burns is currently working toward the Ph.D. degree in the Department of Computer and Information Science, University of Massachusetts at Amherst. His doctoral work is in matching three-dimensional models to two-dimensional natural scene images.



Allen R. Hanson (S'65-M'69) was born in Jamaica, NY, on August 4, 1942. He received the B.S. degree from Clarkson College of Technology, Potsdam, NY, in 1964, and the M.S. and Ph.D. degrees from Cornell University, Ithaca, NY, in 1966 and 1969, respectively, all in electrical engineering.

In 1969 he joined the Department of Computer, Information, and Control Sciences, University of Minnesota, Minneapolis. In 1973 he joined the faculty of Hampshire College, Amherst, MA, where he was responsible for computer science curriculum development and the design and implementation of a networked microprocessor lab. Since February 1981 he has been on the faculty of the Department of Computer and Information Science, University of Massachusetts, Amherst. The author of numerous technical articles and papers in the areas of pattern recognition, artificial intelligence, and machine vision, he has also served on many conference and technical committees, and was General Chairman of the 1985 Computer Vision and Pattern Recognition Conference. He is the coauthor of *Fundamentals of the Computing Sciences* (Prentice-Hall) and coeditor of *Computer Vision Systems* (Academic Press) and *Vision, Brain, and Cooperative Computation* (MIT Press, to appear). For the past several years he has been engaged in research on computer vision and image processing. Recent applications areas have included biomedical image processing, industrial automation, and remote sensing.

Dr. Hanson is a member of the Association for Computing Machinery, the Pattern Recognition Society, AAI, Eta Kappa Nu, and Tau Beta Pi.



Edward M. Riseman (S'65-M'68) was born in Washington, DC, on August 15, 1942. He received the B.S. degree from Clarkson College of Technology, Potsdam, NY, in 1964, and the M.S. and Ph.D. degrees from Cornell University, Ithaca, NY, in 1966 and 1969, respectively, all in electrical engineering.

Since 1969 he has been on the faculty of the Department of Computer and Information Science, University of Massachusetts, Amherst. In 1978 he was promoted to Professor and became Director of the Laboratory for Computer Vision Research. From 1981 to 1985 he served as Chairman of the Department of Computer and Information Science. In 1978 he was on leave at SRI International. He has served on a variety of program committees for conferences in the fields of artificial intelligence and pattern recognition. He has conducted research in context analysis for character recognition, computer-aided instruction, parallelism in programs, concept learning, and communication systems for nonvocally severely disabled individuals. His primary research interests for the past ten years have spanned a wide range of research issues in constructing effective integrated computer vision systems, including segmentation, techniques for knowledge-based interpretation, motion analysis, parallel architectures for vision, and navigation of mobile vehicles. He is coeditor of *Computer Vision Systems* (Academic Press, 1978).

Dr. Riseman is a member of the Association for Computing Machinery, AAI, the Pattern Recognition Society, Tau Beta Pi, and Eta Kappa Nu.