Edge, Junction, and Corner Detection Using Color Distributions

Mark A. Ruzon and Carlo Tomasi, *Member*, *IEEE*

Abstract—For over 30 years researchers in computer vision have been proposing new methods for performing low-level vision tasks such as detecting edges and corners. One key element shared by most methods is that they represent local image neighborhoods as constant in color or intensity with deviations modeled as noise. Due to computational considerations that encourage the use of small neighborhoods where this assumption holds, these methods remain popular. This research models a neighborhood as a distribution of colors. Our goal is to show that the increase in accuracy of this representation translates into higher-quality results for low-level vision tasks on difficult, natural images, especially as neighborhood size increases. We emphasize large neighborhoods because small ones often do not contain enough information. We emphasize color because it subsumes gray scale as an image range and because it is the dominant form of human perception. We discuss distributions in the context of detecting edges, corners, and junctions, and we show results for each.

Index Terms—Edge detection, junction detection, corner detection, earth mover's distance, color distributions, perceptual color distance.

---

1 INTRODUCTION

DETECTION of low-level features in images such as edges, corners, and junctions is a classic problem in computer vision since it is believed that these features can be detected without knowledge of the objects in the world that caused them. Many different models have been proposed, some to detect one of these features, others to detect all of them. Some detectors are designed for real-time applications, while others are more complex.

In this article, we are primarily concerned with the representation of an image neighborhood used to find these features. If the representation is inadequate, no amount of cleverness in the algorithm will yield good results. On the other hand, a more complete description of the image data should give better results, though it may take longer to compute them.

Most feature detectors model the neighborhoods near a feature as constant in intensity or color and treat any deviations as noise, though some have foregone this property, at least for edge detection. Nonconstancy in one dimension was examined by Nalwa and Binford [1], who modeled intensity as constant along an edge and as a hyperbolic tangent across it. Leclerc and Zucker [2] modeled one-dimensional neighborhoods as polynomials up to degree 3, again assuming that intensity was constant along the edge. The later edge detectors of Wang and Binford [3] and Binford and Chiang [4] compensated for shading effects by modeling image neighborhoods as planes with arbitrary surface normals. For the most part, though, a weighted mean of the neighborhood has formed the representation.

We propose a more general model of a neighborhood as a *distribution* of values, allowing multiple pixel values along with their relative frequencies. The relative frequencies often sum to one, making distributions equivalent to probability density functions. Using distributions extends the conceptual range of feature detection in two important directions: large scales and color images.

When attempting to represent increasingly larger image regions with a weighted mean, the accuracy of the representation drops and, with it, the ability to detect features better. Fig. 1 plots the average quantization error per pixel against the region size for different numbers of clusters. The error when using one cluster (the mean) is not only higher but also grows faster. Most feature detectors are, therefore, applied only at small scales. Distributions model image content more accurately.

In addition, computing means may be inappropriate for a color image. If two or more perceptually different colors exist in the same neighborhood, the "mean color" may not appear related to any of the original colors, producing unintuitive results. This phenomenon does not exist in gray scale, even if black and white pixels are averaged together, because the range is one-dimensional and, therefore, "perceptually closed."

Our goal is to show that the increase in representational accuracy due to the use of distributions does indeed translate into better feature detection. To achieve this goal, we must overcome the conceptual difficulty in determining the perceptual distance between two color distributions, a problem we address after summarizing the relevant literature. We then present our algorithms and results based on this representation. This article unifies research presented in [5], [6].

- M.A. Ruzon is with Quindi Corporation, 480 S. California Avenue, Suite 304, Palo Alto, CA 94306. E-mail: ruzon@quindi.com.
- C. Tomasi is with the Computer Science Department, Stanford University, Stanford, CA 94305. E-mail: tomasi@cs.stanford.edu.

2 RELATED WORK

We focus on different approaches to color edge detection in this section since we are unaware of attempts to find

Comparison of Quantization Error for Color and Gray Scale Images

- ○——○ Color, 1 cluster
- ○- - -○ Gray, 1 cluster
- ☐——☐ Color, 5 clusters
- ☐- - -☐ Gray, 5 clusters
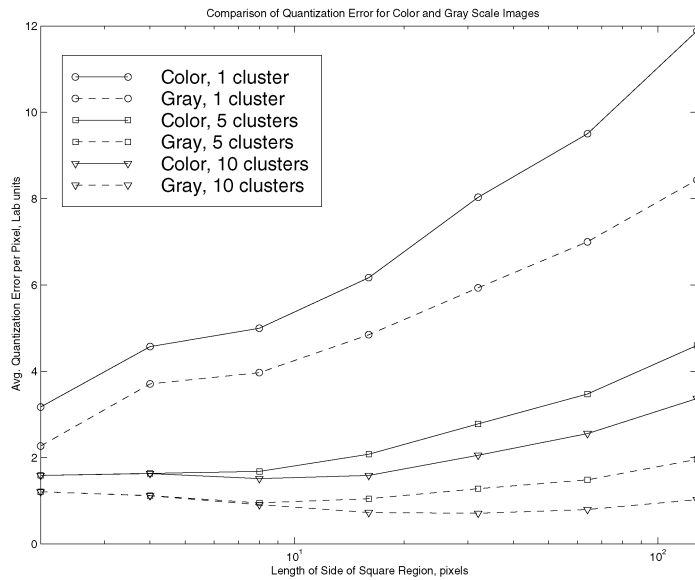- ▽——▽ Color, 10 clusters
- ▽- - -▽ Gray, 10 clusters

Fig. 1. As neighborhood size increases, average quantization error increases, but the error is much smaller if distributions are used instead of the mean (1 cluster). This graph is the result of a quantization algorithm applied to random windows over a large set of natural images.

corners or junctions in color images. A more complete version of this survey is found in [7].

Edges are detected by determining the degree to which each image window matches an edge model, followed by a decision stage that labels each pixel or subpixel point as "edge" or "nonedge." The use of color images adds one important step, image recombination, which can be inserted at different places in the pipeline (see Fig. 2). This insertion translates into performing some set of operations on each color component (including the null set). The intermediate results are then combined into a single output. The point at which recombination occurs is key to understanding the different categories of color edge detection algorithms: output fusion methods, multidimensional gradient methods, and vector methods. We consider each category in turn.

2.1 Output Fusion Methods

In output fusion methods, gray-scale edge detection is carried out independently in each color component; combining these results yields the final edge map. Nevatia [8] developed the first output fusion method. He computed edges by running Hueckel's edge detector [9] on the luminance component and on two chromaticity components. Each was independent, but the orientation at each point was constrained to be the same. The three separate orientations were weighted to give a final orientation, after which all other parameters were recomputed.

Most other approaches, however, computed a weighted sum of the results of an operator on each component. Shiozaki [10], Malowany and Malowany [11], Carron and Lambert [12], [13], and Weeks and Myler [14] all proposed different combinations of operators, weights, and color spaces.

A more sophisticated approach came from Salinas et al. [15]. They proposed regularization as a way to fuse the outputs of three separate edge maps found by using Canny's edge detector [16], while at the same time introducing "well-posedness" to the inherently ill-posed problem of edge detection. The final edges minimized a functional that summed the perturbations between the final edge map and each component's edge map plus a curvature measure.

2.2 Multidimensional Gradient Methods

Multidimensional gradient methods are characterized by a single estimate of the orientation and strength of an edge at a point. The first such method belongs to Robinson [17], who also appears to have published the first paper on color edge detection. He computed 24 directional derivatives (8 neighbors $\times$ 3 components) and chose the one with the largest magnitude as the gradient.

However, it was Di Zenzo [18] who wrote the classic paper on multidimensional gradients. His method was derived algebraically, but it is perhaps better explained in terms of matrices. A 2×2 matrix is formed from the outer product of the gradient vector in each component. These matrices are summed over all components, and the square
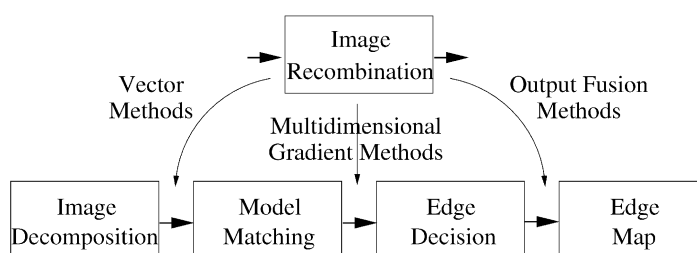
Fig. 2. Flowchart of color edge detection algorithms. Placing the image recombination step at different points results in different types of algorithms.

root of the principal eigenvalue (i.e., the principal singular value) becomes the magnitude of the gradient. The corresponding eigenvector yields the gradient direction. This approach was used in various forms by Cumani [19], Drewniok [20], Saber et al. [21], and Chapron [22].

Others have developed distinctly different approaches. Normalized hue contrast (Moghaddemzadeh et al. [23]), a heuristic choice of component gradients (Tsang and Tsang [24]), relaxation (Macaire et al. [25]), and local averaging (Scharcanski and Venetsanopoulos [26]) round out this set.

2.3 Vector Methods

In vector methods, the decomposition and recombination steps nullify each other; the vector nature of color is preserved throughout the computation. How to represent and use these vectors has varied greatly, however. Machuca and Phillips [27], the first such method, created one-dimensional vectors, as they felt that color was useful only where gray-scale edge detection failed.

Huntsberger and Descalzi [28] used fuzzy membership values, while Pietikainen and Harwood [29] used histograms of vector differences. Both Yang and Tsai [30] and Tao and Huang [31] used vector projections, but the first projected colors into gray scale, while the latter projected vector differences onto segments connecting color clusters. Djuric and Fwu [32] found edges using the MAP (maximum a posteriori) rule.

Perhaps the most compelling work in vector methods so far has been that of Trahanias and Venetsanopoulos [33]. Their method used the median of a set of vectors, which is the vector in that set whose distance to all other vectors is minimized. Once the vector median has been determined, vectors in a neighborhood are sorted by their distances from the vector median, and various statistics are measured and used for edge detection.

We believe that algorithms that incorporate more vector operations are preferable to those with fewer. Even though the mechanics of color perception involve three separate processes, our sensation of color (and, therefore, color edges) is unitary. We do not perceive a "red edge" between a yellow and a green region, for example.

3 REPRESENTING AND COMPARING COLOR DISTRIBUTIONS

The algorithms proposed here require computing the perceptual distance between the representations of adjacent image neighborhoods. In this section, we consider the problems of representing a neighborhood as a color distribution, computing the similarity between individual colors, and computing the dissimilarity between two color distributions.

3.1 Creating Color Signatures

A *color signature* is a data structure consisting of a set of ordered pairs $\{(x_1, \mathbf{v}_1), (x_2, \mathbf{v}_2), \ldots, (x_n, \mathbf{v}_n)\}$, where the $\mathbf{v}_i$s are vectors in a color space to which the weights x_i are assigned. A signature is equivalent to a probability mass function when the x_is sum to one. Signatures are superior to histograms because they adapt to the data; they do not force an arbitrary partitioning of color space. See [34] for a more detailed comparison.

In theory, every distinct color vector in a neighborhood could become a separate point mass. However, we prefer more compact signatures for two reasons: 1) large signatures are more computationally expensive to compare than small signatures, and 2) humans cannot distinguish dozens of different colors in a neighborhood anyway.

Therefore, we use vector quantization to produce compact signatures. In particular, we use the binary split algorithm of Orchard and Bouman [35] since it was developed for color images and allows control over the signature's size. It is a greedy algorithm that, at each step, splits the cluster whose covariance matrix has the largest principal eigenvalue. We terminate the algorithm if the maximum number of clusters (usually 10) has been reached, or if the largest eigenvalue falls below a threshold.

3.2 A Perceptual Color Distance

Before computing the perceptual distance between two color signatures, we must first compute the *ground distance* between each pair of colors in the window. For this distance measure to agree with human perception, we must find a good combination of a color space and a function on that space.

There is general agreement that the organization of color in our perceptual systems is three-dimensional, but the actual assignment of coordinates to colors depends on the task involved. As a result, many color spaces exist (see [36] for details on many of them). Few of these spaces were designed to mimic perceived color distances, however. In particular, the Red, Green, and Blue (RGB) color space has hardly ever been advocated as a good space for measuring color distances.

One of the few color spaces that was designed from a perceptual standpoint is the CIE-L*a*b* color space [37]. CIE-Lab (we drop the asterisks) was constructed from the results of psychophysical color similarity experiments. The Euclidean distance between two nearby colors in this space is intended to be equivalent to their perceptual distance. CIE-Lab is not ideal; in particular, the experiments that led to its creation used large uniform patches rather than pixel-sized elements, which has a noticeable effect on our perception. However, we have found it to be effective in providing an accurate measure of perceptual color distance.

Unfortunately, simply measuring Euclidean distance in CIE-Lab is insufficient. An important caveat is that the equivalence between Euclidean and perceptual distances holds for small distances only. For larger distances, the most we can say about a pair of colors is that they are different. For feature detection, this is exactly what is required; once two colors are far enough apart that we can perceive contrast between them, their actual Euclidean distance is irrelevant.

This key observation turns out to be independent of our choice of color space. We are not interested in any *physical* properties of the color stimuli; only the amount of perceptual dissimilarity between two stimuli is important. Stimuli that are infinitely far apart should have a distance of one, not infinity. Including the requirements that the function be smooth, monotonic, and a metric leads to a choice of

$$d_{ij} = 1 - \exp\{-E_{ij}/\gamma\}.$$

In other words, the ground distance between color i and color j is an exponential measure, with steepness governed by γ (we use $\gamma = 14.0$), of the Euclidean distance E_{ij} between them in CIE-Lab. This function also has the advantage of being roughly linear for small distances, which is why the choice of a color space is still relevant.
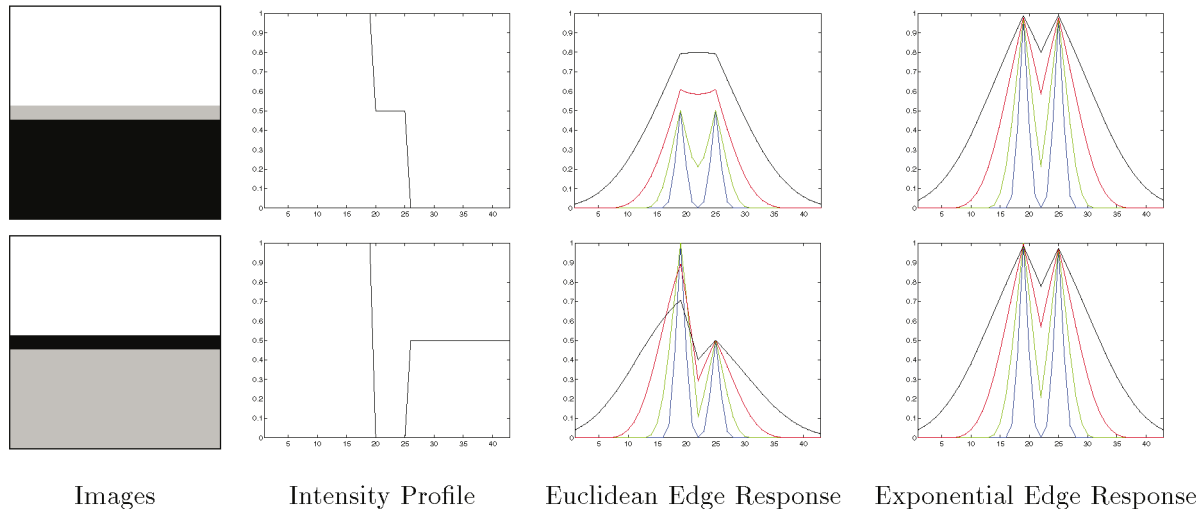
Fig. 3. Comparison of Euclidean and exponential distance measures. The blue, green, red, and black functions are responses of the proposed edge detector for σ = 1, 2, 4, and 8, respectively.

Fig. 3 compares this distance measure with the Euclidean distance on two test images, each with two salient edges, but the ordering of black and gray has been inverted. If we treat the distance from white to black as twice that from gray to either color and examine the edge strength as scale varies, the results are unintuitive. The two edges will be less distinguishable as scale increases, but a detector's response to each ought to be equal and constant. However, the plots show the Euclidean edge response for the top image becoming stronger as scale increases, while the response to one edge in the second example weakens as scale increases. The exponential distance measure returns a value of one at all edges regardless of the scale chosen or the ordering of colors.

Theoretical justification for this measure can be found in the work of Shepard [38], who proposed that an exponential law governs not only color similarity, but also similarity in other perceptual phenomena such as size, shape, pitch, and phonemes. Only γ and the underlying norm (Euclidean, Manhattan, etc.) for the physical space changes.

3.3   Distance between Signatures

Measuring distance between two color signatures is a subproblem of measuring distance between probability density functions. Results from ergodic theory, probability theory, and information theory have combined to produce many different distance measures. A primitive distance measure that fits this category was developed by Levenshtein [39] for two binary strings that need not have the same length.

The constraints of computational efficiency and matching human perceptual similarity have prevented many of these measures from being applied to vision. Within vision, reliance on histograms has tended to produce algorithms that do not fully take into account the distances between bins as well as the amounts of mass within each bin. For example, the histogram intersection method [40] is less useful in cases where a bin partition splits a cluster of similar pixels. Cross-bin measures have been developed (e.g., [41]) but also fail to measure similarity accurately [34]. The Hausdorff distance has led to a family of distance measures based on local correspondence between points [42], but often a global correspondence is needed.

The Earth Mover's Distance (EMD) overcomes these limitations by formulating the distance measurement as an instance of the transportation problem. If one signature is represented as piles of dirt and the other as a set of holes, the minimum amount of work needed to move the dirt into the holes is the EMD. It was first used by Rubner et al. [43] to compute a perceptual distance between images by comparing their color signatures. However, we note that in the vision community a one-dimensional form of the EMD known as the *line feature distance* was used by Shen and Wong [44]. It was later renamed the *match distance* and extended to multidimensional histograms by Werman et al. [45].

Similar measures in ergodic and probability theory include $\bar{d}$ [46], $\bar{\rho}$ [47], and Wasserstein[1] [48]. The key feature that separates the EMD from these three measures is that it can compute distances between signatures of unequal total mass, finding the subset of the larger that corresponds best to the smaller. This property is important because, when detecting corners, the signatures naturally have unequal total mass.

We have chosen it for our work for a number of reasons. Because the EMD can handle signatures with arbitrary numbers of clusters, it can adapt to arbitrary complexity. Robustness stems from the fact that perturbations in cluster centers or even small changes in the number of clusters do not have disproportionate effects on the result. Finally, previous work in color and texture [34], [49] shows it to be an effective tool for matching human perception under a wide range of conditions.

4   EDGE DETECTION

Here, we present a method of estimating the strength and orientation of an edge hypothesized to split a window. To summarize, we divide the window in half with a line segment, compute a color signature for each half, and find the EMD between them. Repeating this process using line segments with different orientations reveals the true strength and orientation of the edge.

---

1. This Russian transliteration is the accepted one in the literature, but the cited article uses "Vasharshtein."
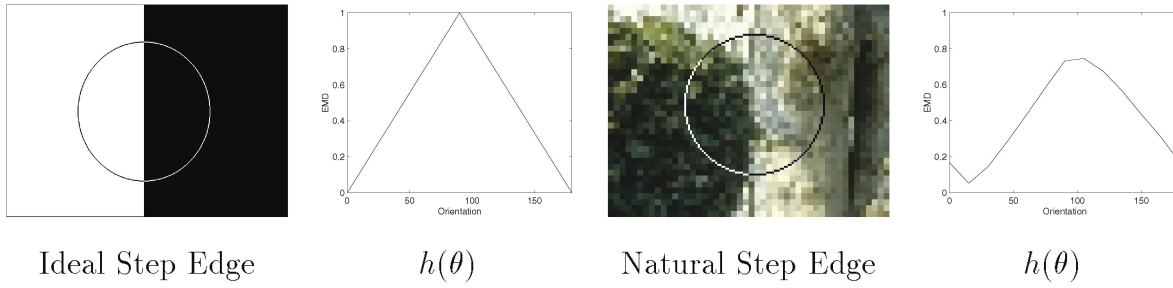
Fig. 4. Applying the EMD to a circular neighborhood over a range of diameter orientations produces a function from which edge information is extracted.

The window is circular so that all orientations of the line segment (now a diameter) receive equal treatment. The radius of the circle is 3σ, where σ is a scale parameter. Using Orchard and Bouman's algorithm, we perform vector quantization on all pixels whose intersection with this circle is nonzero. To form two color signatures, we map each pixel in a semicircle to its nearest color cluster and add its weight to the corresponding point mass.

4.1 Choosing Pixel Weights

Most edge detectors weight some pixels more than others. The weight given to each pixel in this method is the product of three factors: the area of a pixel (which we model as a unit square) that falls inside a semicircle, its relative importance to the computation, and a normalization factor so that the total mass of each signature is one. We model a pixel's importance as a Rayleigh distribution on the distance from a pixel's center to the circle's center. Rayleigh distributions are expressed in polar coordinates as:

$$f(r, \theta) = r \exp\{-r^2/2\sigma^2\}.$$

This function in one dimension is similar in shape to the weighting functions used by Canny and by Deriche [50]. We depart significantly from their formulations in 2D by revolving the 1D curve into an isotropic function.

Isotropy is needed for computational efficiency; without it the weight of each pixel would change as the orientation of the diameter varies. Because we do not assume that the image gradient is normal to the edge's orientation, we must incur the penalty of using more than two orientations. Also, some windows may have two edges going through the circle's center, and we would like to detect both.

We increase efficiency further by noting that a set of diameters divides the circle into wedges. Maintaining the color signature of each wedge separately makes it easy to update the signatures for each new orientation. We choose our wedges to be $15°$ wide.

4.2 Extracting Edge Information

The result of computing the EMD over a set of orientations at each window is a function $h(\theta), 0 \leq \theta < 180$. Fig. 4 shows two applications of the algorithm on two windows, the first straddling an ideal step edge and the second taken from an image containing a bush and a tree. The strength of each edge is the maximum value of $h(\theta)$, which never exceeds one, and the orientation of the edge is the argument that produced the strength.

We do not expect the true orientation of the edge to be one in the sampled range, however. The vertex of the parabola containing the three highest EMD values gives the true strength and orientation. Curve-fitting is necessary because we do not assume that the responses at $0°$ and $90°$ can be combined to form a gradient.

In addition, there are isolated instances where two separate edges can be made out, resulting in two separate maxima in $h(\theta)$ with the same strength. The algorithm can recognize this case and record two orientations, but the strength of each is equal to the maximum of the two. An example is shown on the left of Fig. 5.

The edge on the right of Fig. 4 is somewhat curved instead of straight, resulting in the two highest EMD values being almost equal. The right half of Fig. 5 shows a more dramatic example where the curvature of the edge causes many orientations to have almost the same EMD values. Since parabolic interpolation would be unstable, we pursue a different approach.

The width of the continuous interval whose values are all close to the maximum EMD (meaning greater than 95 percent) is the *uncertainty* of the orientation. The strength
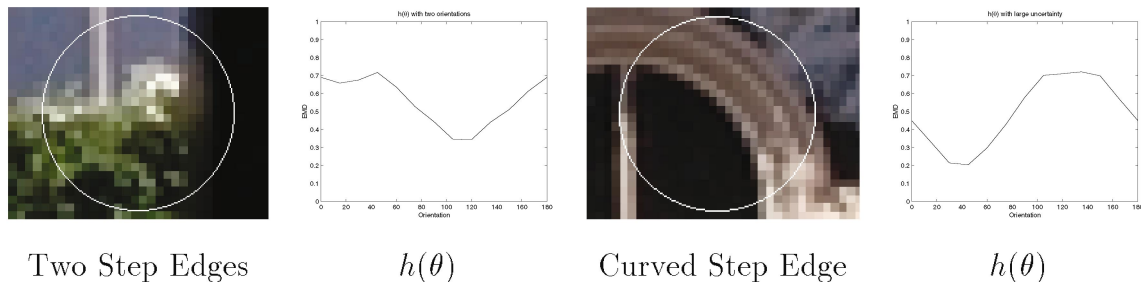

Fig. 5. In rare instances, $h(\theta)$ will have two peaks ($0°$ and $42.7°$) with the same strength, indicating two separate edges. Also, uncertainty can be detected and measured in cases where curvature causes many orientations at a point to have the same EMD.

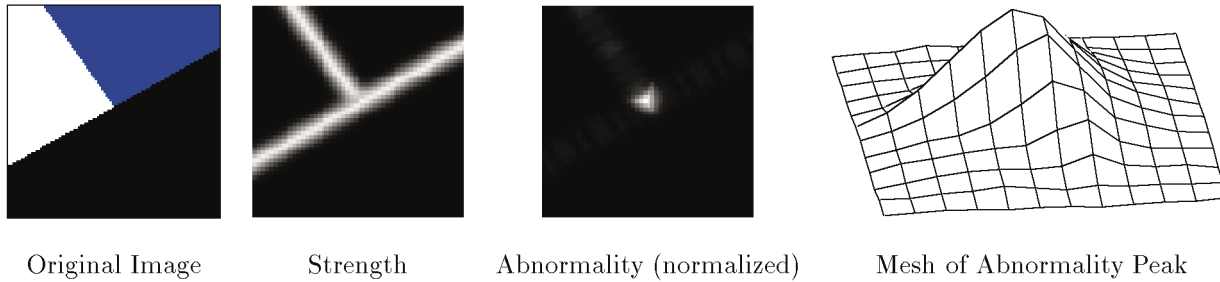| Original Image | Strength | Abnormality (normalized) | Mesh of Abnormality Peak |

Fig. 6. Whereas the strength at each point represents the maximum EMD value over all orientations, the abnormality represents the minimum, which is high near junctions. The shape of the mesh resembles a triangular pyramid.

is the maximum sampled EMD, and the orientation is the midpoint of the interval.

5 JUNCTION DETECTION

A junction is a point where three or more image regions meet, but the EMD, like other distance measures, cannot handle three color signatures at a time. It is certainly possible to split a circle into three or more neighborhoods and find the EMD between each pair, but this is an order of magnitude more expensive, and the added problem of determining the number of image regions meeting at a point is cumbersome. Instead, we opt for a less direct but simpler approach.

One of the main assumptions of gradient-based edge detectors is that the edge exists along an isocontour of intensity or color, which is not always true. For an ideal step edge, this is the case; at an orientation normal to an edge, the two color signatures are identical and the EMD is exactly zero.

In the natural image examples we have shown so far, the minimum EMD over all orientations is never exactly zero due to inhomogeneities in the spatial distribution of colors. The closer the minimum is to zero, the more the image data matches the model of an ideal step edge regardless of the strength. For this reason, the minimum value of $h(\theta)$ is called the *abnormality*.

Although abnormality can provide a quantitative measure of confidence in our edge computation, it serves a much more interesting purpose. When abnormality is high, it indicates a complete lack of symmetry in the image data that usually corresponds to a junction. The notion is similar to the original idea of Moravec [51], who developed an "interest operator" by looking for neighborhoods where the sum of squared differences of adjacent pixel intensities is high in eight directions. Abnormality naturally extends this idea to all directions.

The synthetic image in Fig. 6 illustrates this concept. Points near the edges have high strength, but only those near the junction have high abnormality as well. The shape of the mesh provides a clue as to how to interpret the image content in this neighborhood.

We propose the following algorithm for reconstructing junctions containing three regions: For each local maximum above a threshold, examine a small neighborhood (with size proportional to σ) centered at the maximum. Fit a triangular pyramid to the abnormality values and the sides of the base will be almost normal to the edges that form the junction. For "T-junctions," the error in this approximation must be more than for "Y-junctions."

Abnormality is an attractive feature of this edge detector because junctions are among the most difficult places for any edge detector to function properly. The effects of nearby junctions on the location of abnormality peaks in natural images, however, is more complicated, and we have not fully investigated its properties.

6 CORNER DETECTION

A corner can be informally defined as a junction with only two regions, with one region subtending a smaller angle than the other. We cannot detect corners using abnormality because the EMD of the corner point at the orientation the bisects the corner is usually zero. Running the edge detector at a corner point yields an orientation normal to the bisector of the corner with high uncertainty; the strength is not even a local maximum. Finding corners, therefore, requires a separate approach.

Most corner detectors in the literature bear little or no relation to edge detectors since edge detectors usually assume a window that is split in half by an edge. The SUSAN operator [52] is a noteworthy exception. Also, corner detectors have been applied to color images very rarely, if ever. This section presents a method based on our edge detector that works directly on color images.

Under the proposed framework of color distributions, the only difference between detecting edges and detecting corners is that we no longer have half the wedges of the circle in each signature. An extra parameter β is introduced to measure the angle subtended by the corner. The range of θ must now be $[0, 360)$ since the two radii that split the circle no longer have odd symmetry. We refer to the two sides as "clockwise" and "counterclockwise," and θ refers to the orientation of the clockwise side. Fig. 7 illustrates the parameters.
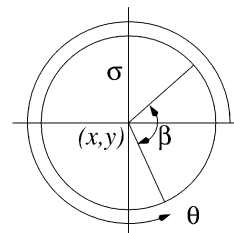

Fig. 7. Parameters of the corner detector. (x, y) marks the center of the circle, σ is the scale parameter, the orientation of the clockwise side of the corner is θ, and β is the angle subtended by the corner.
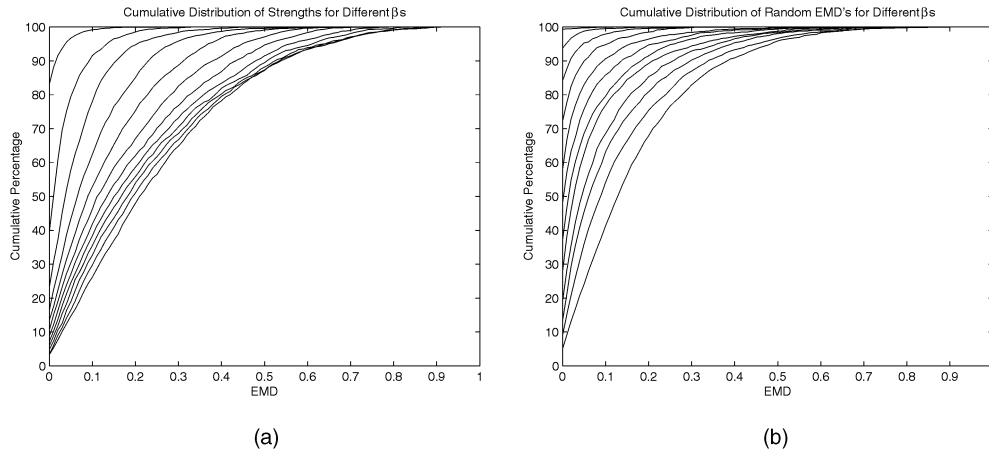
Fig. 8. Cumulative distributions for the response of the corner detector. Each curve represents a different value of β, with $\beta = 15°$ closest to the upper left and $\beta = 180°$ closest to the lower right. (a) Distribution of maximum EMDs (strengths) at each point. (b) Distribution of randomly chosen EMDs at each point. The significant differences in the curves precludes us from comparing corner responses that differ in β.

Otherwise, the algorithm for computing strength and orientation at each point is similar to that for edge detection. A circular neighborhood is quantized and two color signatures are formed. The EMD measures the perceptual distance between signatures. We believe the resulting algorithm to be the first corner detector that can handle either color or most arbitrary textures.

There are two practical difficulties we must address: the dissimilarity between two signatures with unequal amounts of mass, and the process for deciding which points are corners. These issues are covered in the next two sections.

6.1 Computing Distance between Signatures of Unequal Mass

When β is less than $180°$, the two color signatures have unequal amounts of mass. There are two possible methods for accounting for this difference. The first is to normalize the larger signature so that its mass equals that of the smaller; a technique called *normalized matching*. The second method, *partial matching*, matches the smaller signature to the subset of a larger signature that minimizes the total work. Regardless of the method, the mass of the smaller signature is set to one so that the EMD continues to lie in the range $[0, 1]$.

Both types of matching can fail under certain circumstances. If a small corner (e.g., $\beta = 30°$) is solid red, for instance, and the outside region has enough red pixels scattered throughout, partial matching will find equal amounts of red and return zero distance, thereby missing the corner. Normalized matching will maintain the percentage difference and return a much higher value.

Normalized matching, however, often performs worse than partial matching on typical corners. Pixels along an edge often have colors that fall between the colors of the regions involved due to antialiasing. Consequently, the EMD is maximized when such pixels are placed outside the corner where they have less effect on the computation. The result is that the size of a corner is often underestimated using normalized matching.

We have chosen partial matching for our experiments. We may miss a few small corners, but experience shows that such corners rarely appear in natural images. For the corners that are found, it is better to describe them accurately.

6.2 Finding Corners

The result of applying the corner detector to an image is a four-dimensional (x, y, θ, β) array of EMD values, and corners are relative maxima above a minimum strength in this array. However, there are some complicating factors since the conditions for corners are more restrictive than for edges.

First, a corner is a response to a phenomenon that takes place over a relatively large portion of the image, so checking only the nearest neighbors in the array will produce too many corners. It is possible to get many maxima all responding to the same corner phenomenon but with noticeably different parameters.

Second, we cannot directly compare EMD responses from parameter values that differ in β. Changing the size of the corner also changes the statistics of the EMDs that are generated. Fig. 8 shows the result of an experiment that evaluated the corner detector for different values of β at 2,000 randomly selected image points over a set of natural images. Fig. 8a shows the cumulative distribution of the strengths at each point, and Fig. 8b shows the cumulative distribution of EMDs from randomly chosen orientations at each point. Comparing EMD values across β would cause a bias against more acute corners. Even the thresholds for corner strength must vary with β.

Therefore, we include edge information to winnow the set of maxima over x, y, and θ for every value of β to the actual corners. If multiple responses to a corner still remain, we select one according to a heuristic. These steps are explained in more detail below.

6.2.1 Testing Corner Candidates

Fig. 9 shows how false positives are distinguished from true corners. A true corner exists because two edges meet at a point; in which case, we expect the orientations of the edges at the endpoints of the corner to match those predicted by the corner detector. Furthermore, the response of an edge detector near a corner is rounded off (see Deriche and Giraudon [53]), and so we expect a weak edge response between the two sides of the corner. On the other hand, a false positive is sometimes generated when inhomogeneities on either side of an edge cause a spurious response. In this case, there will be no supporting edge information in the three places we have predicted.
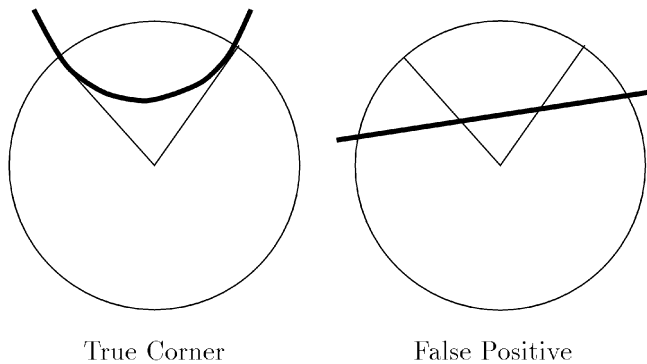
Fig. 9. True corners are aligned with edges at the endpoints. The edge is rounded off in the middle, where the response weakens. False positives occur due to inhomogeneities on either side of an edge.

For each corner, we compute θ_C and θ_{CC}, the difference between the orientation of the clockwise and counter-clockwise sides of the corner, respectively, and the edge orientation found at the endpoint of each side. The fit to the model is expressed as

$$P = \cos\theta_C + \cos\theta_{CC},$$

where P lies between 0 and 2. We threshold P at 1.97.

Also, where the edge crosses the corner's axis of symmetry, the projection of the edge response vector onto the line normal to this axis must be weaker than the corner response. We check responses on a small interval along this axis centered at a point $3\sigma(\sec\frac{\beta}{2} - \tan\frac{\beta}{2})$ pixels away from the corner. This quantity is the distance from the corner point to the circumference of an imaginary circle tangent to the sides of the corner at its endpoints.

Fig. 10 shows the initial corner candidates found by thresholding the relative maxima. Most of these responses are false positives, and this procedure greatly reduces their number.

6.2.2 Pruning Multiple Responses

Ideally, only one response to each corner phenomenon remains after this step, but Fig. 10 shows that this is not always the case. We offer a heuristic to choose one when this happens, though perhaps multiple responses could be combined in some way.

First, we must decide when two corner candidates are responding to the same actual corner. In Fig. 10, this is trivial, but the general question admits no equally general solution. We define two corners as being "close enough" if the corner points are within $9\sigma/4$ pixels of each other and one of two conditions is true: 1) either the two clockwise or the two counterclockwise orientations differ by no more than $10°$, or 2) the sum of these differences is no more than $40°$. These conditions group "nested" corners while preserving multiple corners near junctions.

An ambiguity arises when corner X is close to corners Y and Z, but Y and Z are not close to each other. If our notion of "closeness" is global, then the order in which we examine corners affects the final output. Since this is unacceptable, we compute the transitive closure of "closeness," that is, X, Y, and Z will all become part of the same set. It is theoretically possible that corners in distant parts of the image could become part of the same set; in practice, however, the application of the edge model removes enough candidates to prevent this.

Once we have computed the transitive closure, we select the member of each set that maximizes the expression $2C + P + E$, where C is the corner strength, P is the degree of orientation match described earlier, and E is the sum of the edge strengths at the endpoints of the two sides of the corner. C is doubled so that each term contributes equally.
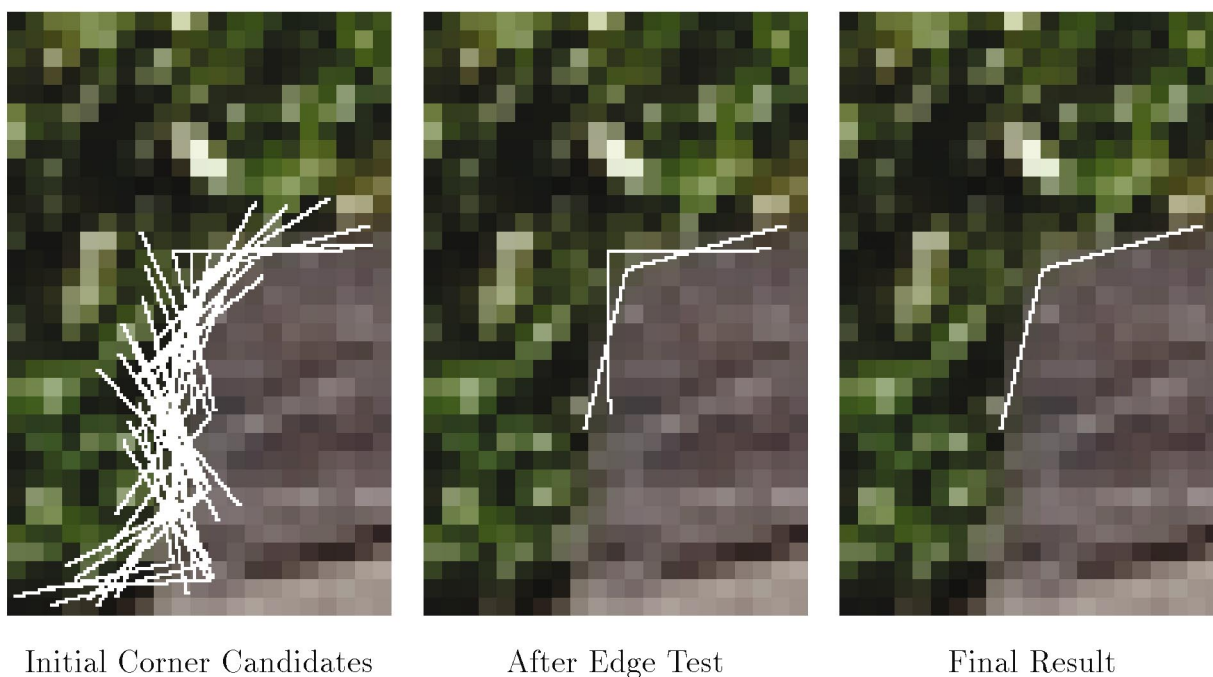

Fig. 10. Detecting corners from an initial set of candidates involves applying an edge model followed by pruning multiple responses if they exist.
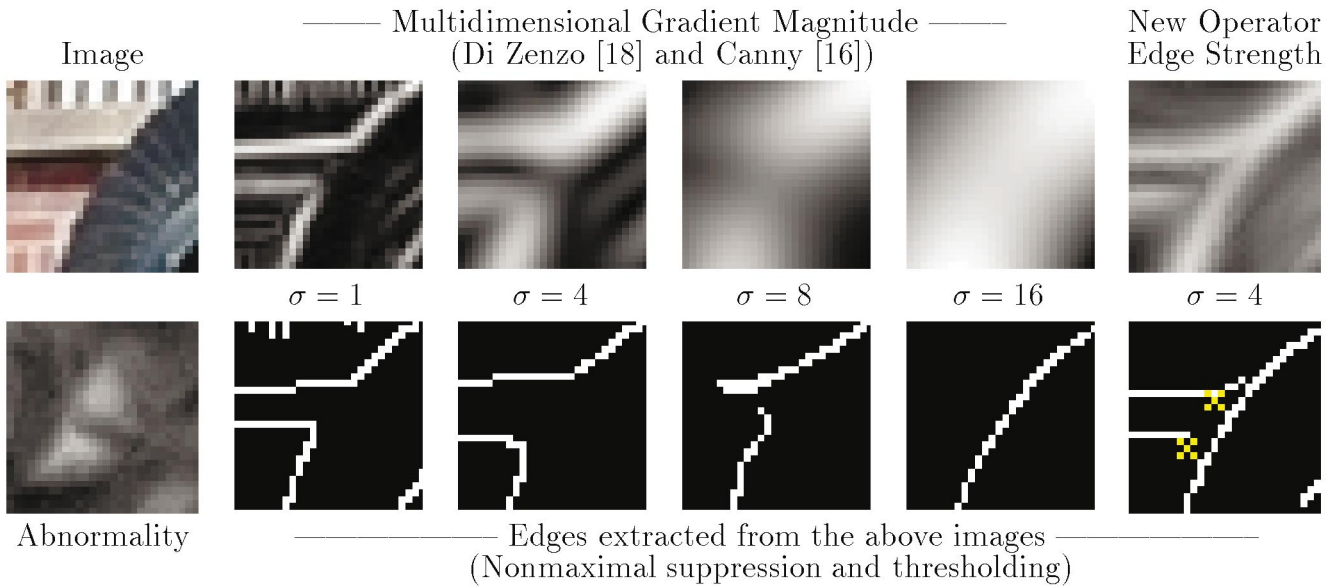
Fig. 11. A region containing high- and low-contrast edges. The gradient magnitude is insufficient to find all the edges at any one scale and would require further processing, while the new operator does find all the edges at one scale. Two junctions are located in the abnormality image, but their peaks (yellow crosses) are not well localized.

The final corner of our example is shown in the third column of Fig. 10.

7 RESULTS

Here, we present results on both small image patches to help analyze the advantages of these algorithms over traditional methods and on larger images to provide intuition about the operator's performance.

7.1 Edges and Junctions

We compare the proposed edge detector to a multidimensional gradient method formed by combining Di Zenzo's gradient in CIE-Lab with Canny's approach into a single operator. We have chosen this operator because the vision community treats Canny's detector as a de facto standard of edge detection due to its simplicity and ease of implementation. For both operators, the actual edges are extracted by using nonmaximal suppression and hysteresis thresholding. When our operator produces two or more responses at a point, only one needs to be a maximum.

Fig. 11 contains four regions, two light and two dark. The edge separating the two dark regions has relatively low contrast. One would expect that the low-contrast edge can be extracted by using a low threshold, but this turns out not to be the case.

At small scales, Canny does not respond to the low-contrast edge. Nonmaximal responses to the high-contrast edge suppress response to the low-contrast edge. If σ is increased past eight, the low-contrast edge is found, but only at the cost of destroying the other edges.

Our method, on the other hand, finds all the relevant edges at one scale. The use of a saturating distance measure prevents the bright regions from having such a marked effect on the computation, allowing the low-contrast edge to be found. In addition, the abnormality located two distinct junctions in the image. However, the peaks are not perfectly

localized, likely due to the fact that these two junctions are close to each other relative to σ.

Fig. 12 shows two regions, one with strong edges and pixels that are the same color as the other region. The edge between them is salient over a wide range of scales. However, the multidimensional Canny operator connects the true boundary to an intratexture edge at $\sigma = 8$. Increasing σ further corrects the mistake, but the edge at $\sigma = 16$ is curved. These differences are due more to the use of distributions to model different colors accurately than to the distance measure used.

When the output on larger image regions is compared, the differences are often striking. Fig. 13 shows a statue with edges extracted by both operators at $\sigma = 4$. Smaller scales pick up too many edges in the ivy and bricks to be useful. The differences between the two are most noticeable in the helmet, shield, right arm, and legs of the statue. Admittedly, our operator's edges do not fully segment the statue from the background, but the improvement is definite.

Another practical advantage of this method is evident at occlusions since they are junctions where more than two colors exist. In Fig. 14, for instance, understanding occlusions properly is critical for any hope of reconstructing the snake.

As in the statue image, a small σ results in too many details being detected to be of use. Fig. 15 shows the edges detected by both operators at $\sigma = 4$. Finding the snake by using either of these images as a guide is bound to be an arduous task; nonetheless, the proposed method has done a better job of keeping the occluding edges intact. In these edge maps, thresholding has been omitted, and the darkness of each pixel is proportional to its strength. Canny's output is lighter on average because the gradient magnitude occupies a much wider range and a few points have a very large magnitude. Because the proposed operator uses a saturating distance measure, more of the edges have large strengths relative to the maximum value.

Closer examination of some of the occlusions in this image (see Fig. 16) provides better insight into the comparative behavior of the two operators. In the first

Edges found using the multidimensional Canny edge detector

Image $\sigma = 2$ $\sigma = 4$ $\sigma = 8$ $\sigma = 16$
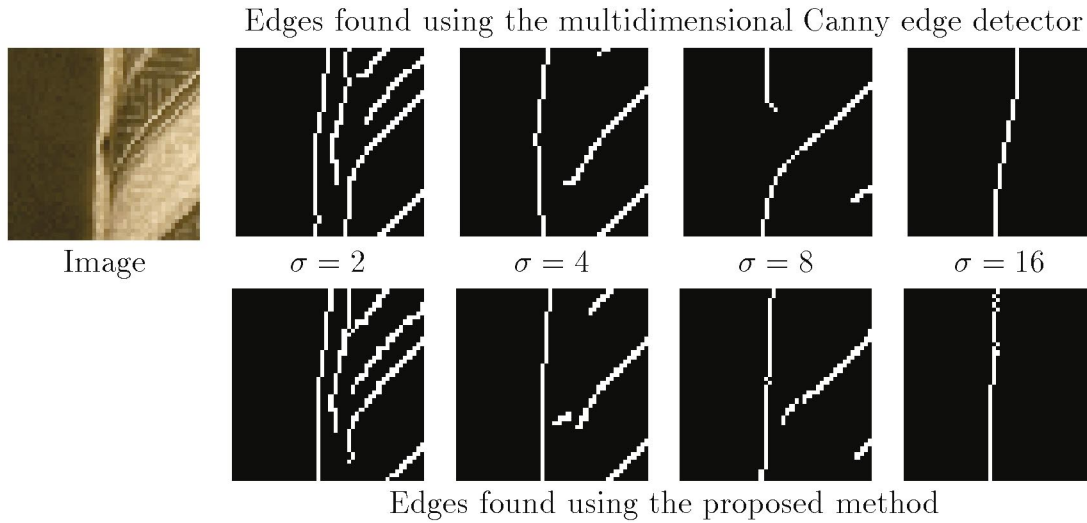
Edges found using the proposed method

Fig. 12. The main edge in this image is salient at a large range of scales even though one region contains strong edges. Canny's edge detector makes an error before $\sigma = 8$, and the lone edge at $\sigma = 16$ is badly localized. The proposed method creates more stable edges as scale changes.

row, both operators find the three edges visible in the image, but the new operator's edges are better localized and come closer to the edge of the blade of grass. In the second row, the edges found by Canny mix boundaries of the snake and grass together, while the new operator is better able to resolve edges of the snake on both sides of the grass. The final row again shows Canny combining snake and grass edges together so that neither will be easy to identify; our method gives a much better chance of being able to distinguish occluding edges from occluded edges.

The one area where the proposed detector falls short is in running time. Canny is separable (its 2D convolution mask can be split into two 1D masks), computes weighted averages, and finds the Euclidean distance between these averages at two orientations. In contrast,

the proposed method is nonseparable, performs vector quantization, and finds the EMD between color signatures at 12 orientations. As a result, Canny's operator can be run on a 768×512 image at practically any reasonable scale in a fraction of a second on an SGI Octane, while the proposed method requires almost 3.5 minutes at $\sigma = 1$, 14 minutes at $\sigma = 4$, and 33 minutes at $\sigma = 8$.

7.2 Corners

All corner results were computed with the same set of thresholds. The lengths of the sides of the corners drawn in the images are equal to the radius of the operator. Also, we have restricted β to the range $[30, 150]$ because $15°$ and $165°$ corners are unreliable; the small corners do not contain enough pixels to get an accurate representation, and the
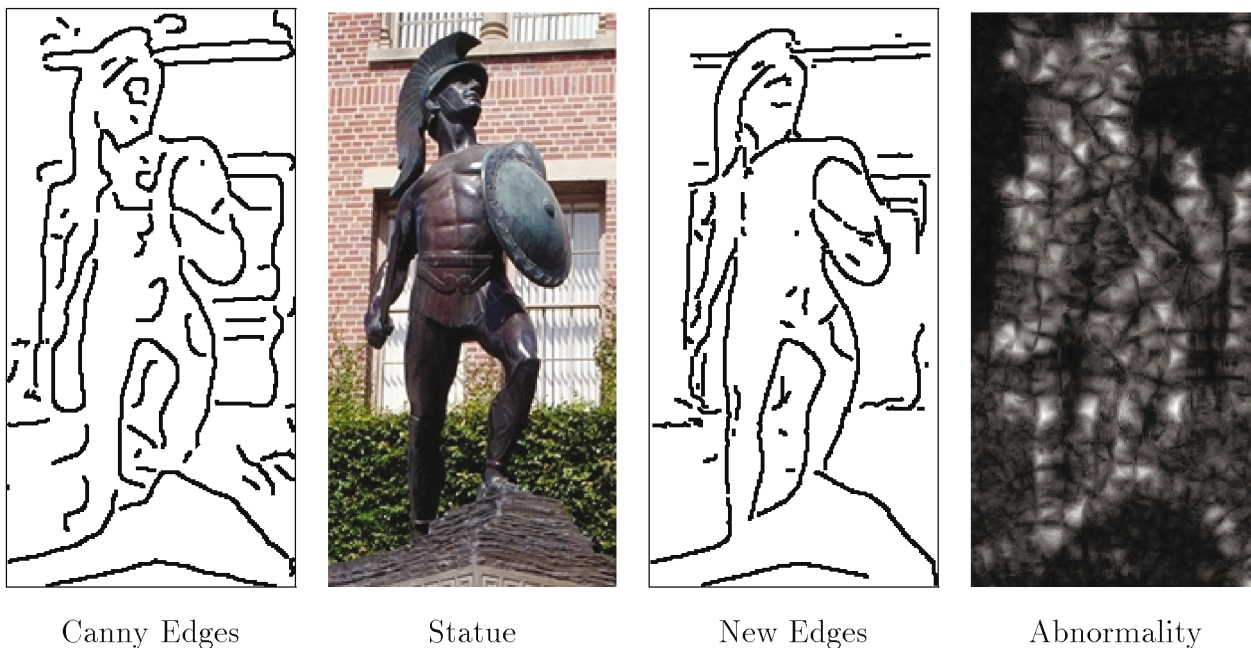

Canny Edges   Statue   New Edges   Abnormality

Fig. 13. Comparison of edges on a 320×160 image. Note the differences in the helmet, shield, right arm, and legs. The peaks in the abnormality, normalized for display, indicate potential junctions. Both algorithms were run using $\sigma = 4$.

Fig. 14. Distinguishing edges of the snake from edges of the grass at points where the two meet is crucial for future identification of the snake.

large corners are nearly impossible to distinguish from edges. However, this appears to be wider than the range of most other detectors.

Fig. 17a shows one fabric occluding another. Although each contains texture that varies greatly in color and has regions in partial shadow, the corner is correctly detected. Fig. 17b is more complicated because three textures are involved: trees, illuminated rock, and rock in deep shadow. Five corners separate the regions and serendipitously form most of the boundary of the illuminated rock.

Fig. 18 illustrates a conceptual difficulty with corner detection. The left image shows corners, including those as large as $165°$, and the other shows edges extracted using the edge detector. There is a high degree of overlap between the features as they are drawn on this image since most of the

corners have high values of β. It begs the question of whether corners are relevant in images that do not consist mostly of man-made objects.

Even more disconcerting is the fact that we have developed two separate algorithms for detecting features from what is arguably the same data. Edge information is simply that obtained by setting β to $180°$, while corners cannot be reliably detected without edge information. The two features are complementary, and both are needed for accurate boundary representation.

Fig. 19 illustrates this last point. The edges found outline the boundary of this rock but, because the rock is convex in shape, the edges are sometimes a few pixels inside the rock. The two corners, however, are well localized, and combining
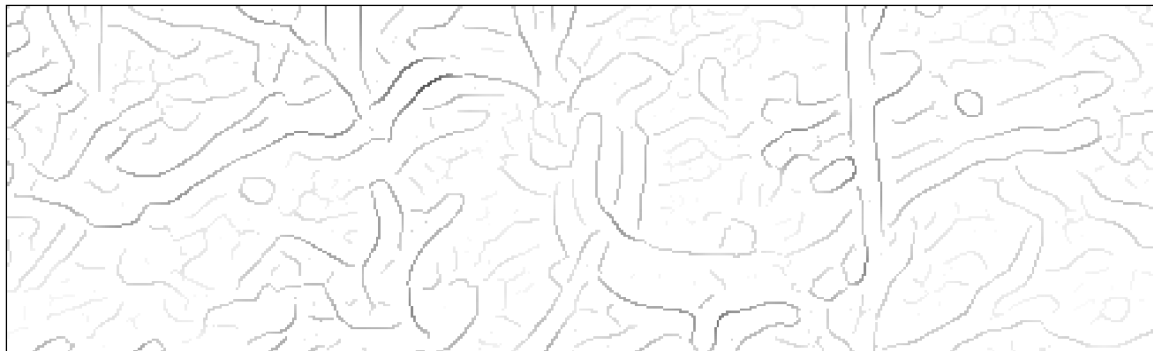
Canny edges, $\sigma = 4$
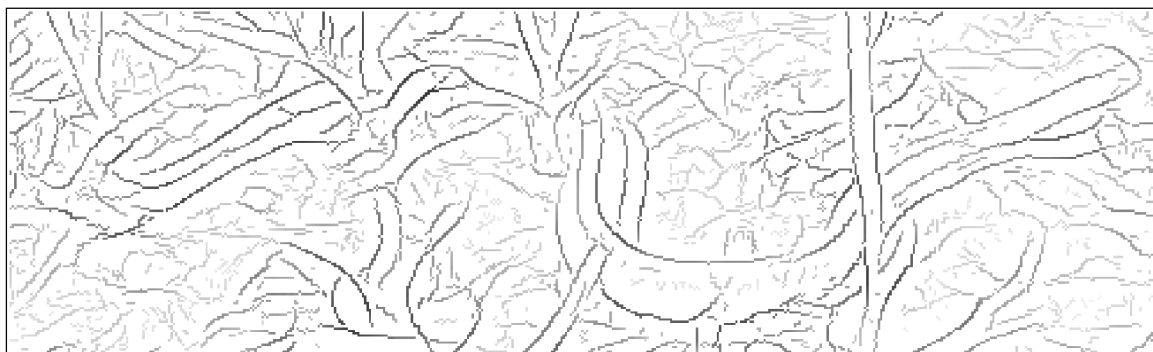
New edges, $\sigma = 4$

Fig. 15. The proposed detector better separates snake edges from grass edges. Darker edges have higher strengths relative to the maximum of each operator's range.

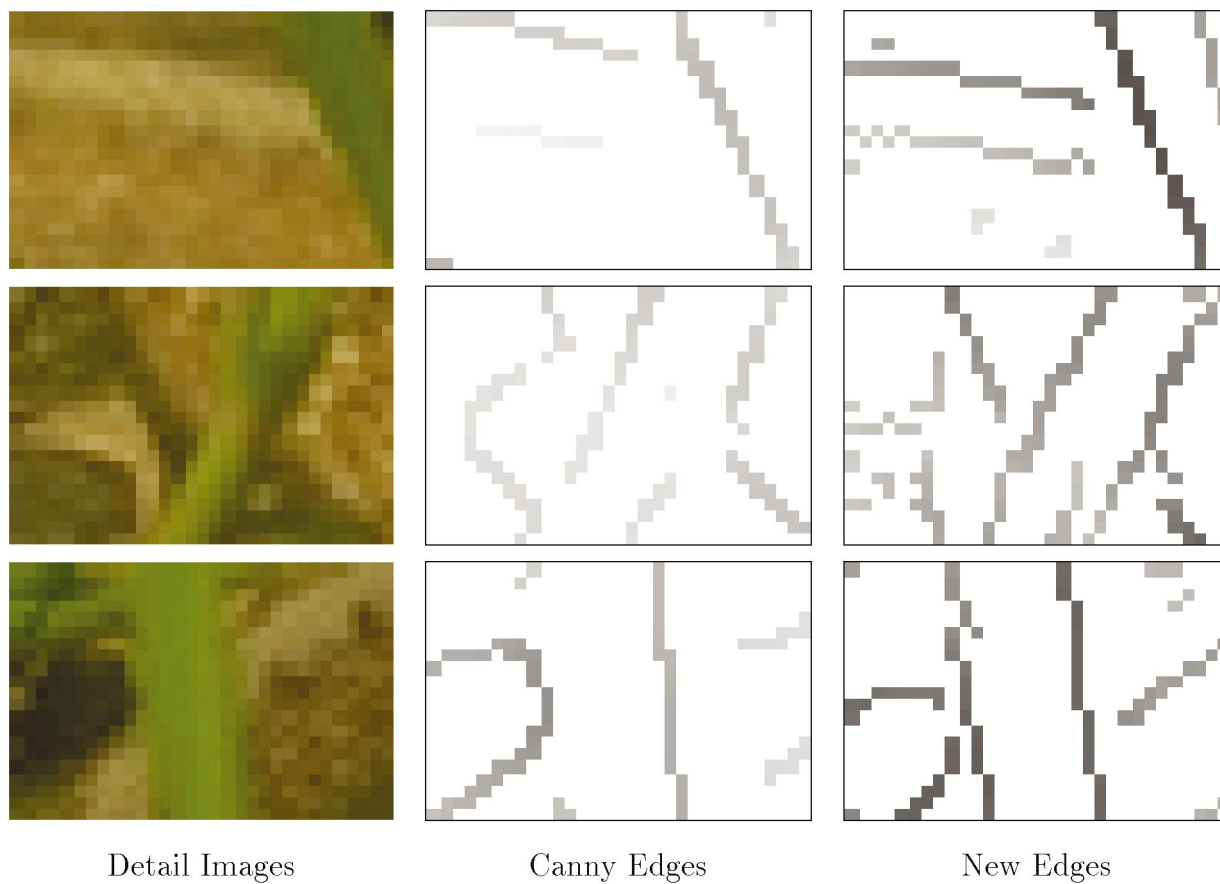Detail Images   Canny Edges   New Edges

Fig. 16. Detail of edge maps in Fig. 15. Because the proposed method accurately models neighborhoods with three or more colors, occlusions are easier to reconstruct.
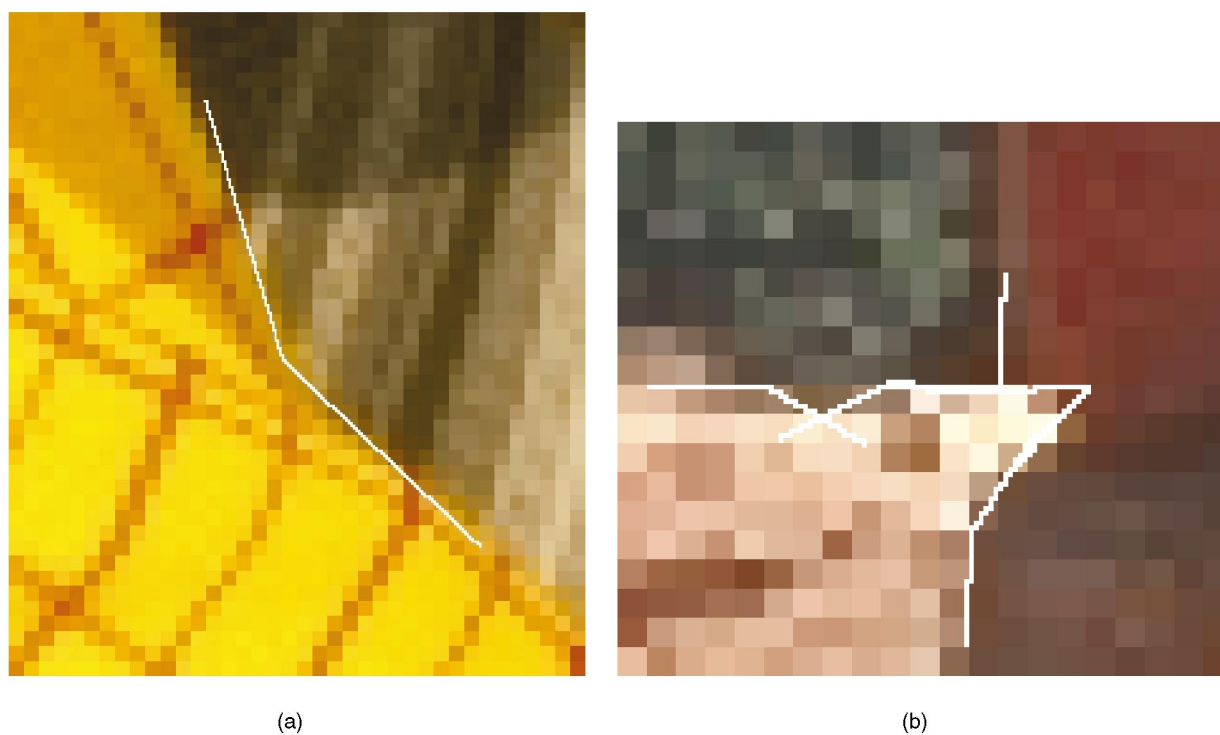

(a)   (b)

Fig. 17. Basic corner detection results. (a) Fabrics. (b) Rocks and trees.
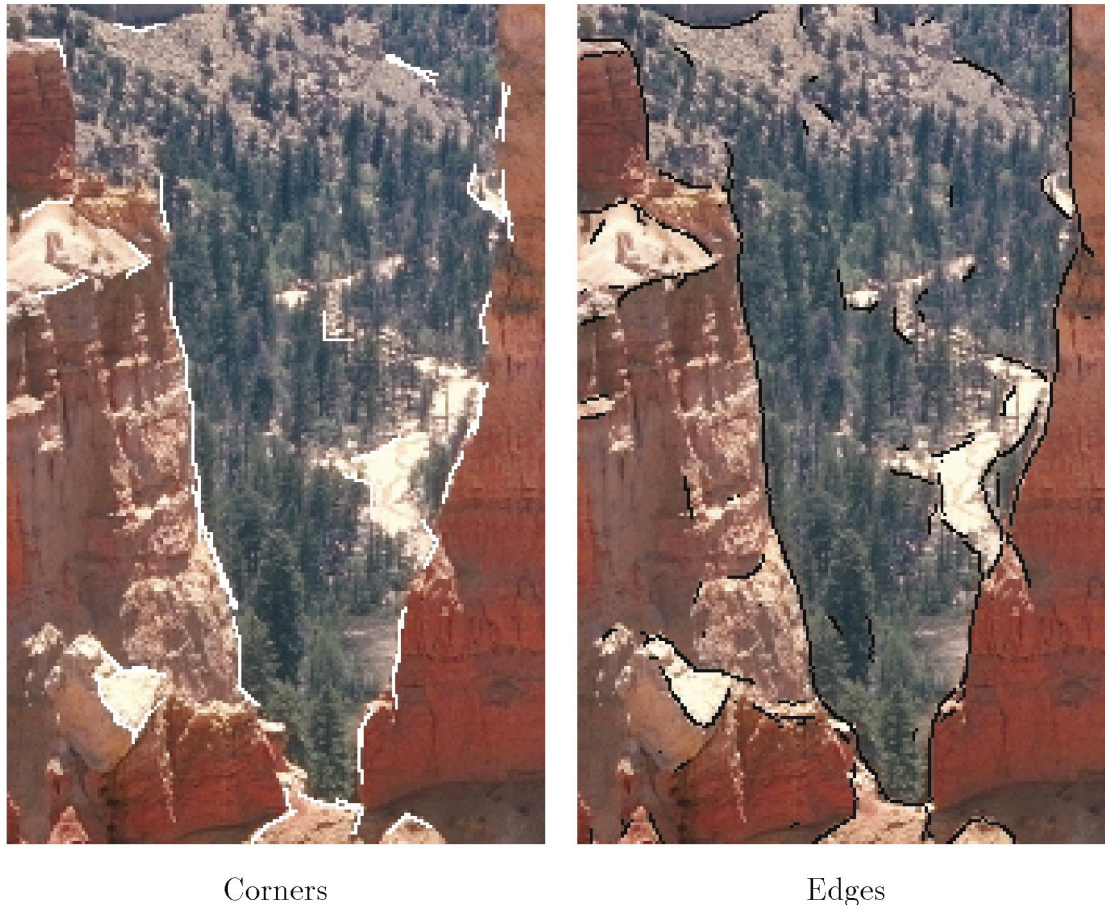
Corners   Edges

Fig. 18. Corner detector and edge detector applied to an image of a canyon. Note the high amount of overlap between the two sets of results. $165°$ corners have been included.

this information with the edges would produce a more accurate boundary.

One way this could be done is by an energy minimization method. The corners can be thought of as basins of attraction that perturb the edges. A better method would be to use the entire 4D array of data to extract a more accurate boundary directly. Following an edge would require decreasing β from $180°$ to the value at the endpoint of the corner and then changing direction and increasing β back to $180°$.

8 CONCLUSIONS

A distribution of values obviously contains more information than a single value, but, if a single value was sufficient for feature detection, the extra information would be redundant. The improvement in the quality of edges detected, the potential for reconstructing junctions, and the ability to find corners in textured, color images at large scales all give testimonial to the added power of distributions in feature detection. Being able to find all three features from one array of values is encouraging. We believe this work to be most applicable to figure-ground separation, a task other operators can carry out only in simple situations. Finding corners with the help of edges and finding occluding edges through the reconstruction of junctions brings us closer to this ultimate goal than other methods.

Among the secondary principles established by this work, three stand out. The first is that it is best to treat color as a vector instead of three components; only then can we be reasonably certain that an algorithm's representation of color is similar to that of a person. Second, using a saturating distance measure is more in line with the notion that all pairs of dissimilar colors are equally different. Finally, the Earth Mover's Distance, previously shown to be applicable to the color distributions of entire images, has been found to be useful for local neighborhoods. It is what allows us to combine distributions and the saturating distance measure.

The limitations of the algorithm are more practical and deal with the high computation cost. Quantizing each image window and computing the EMD many times is much slower than the corresponding operations in other detectors, though these operations can be done largely in parallel. Experiments designed to reduce the running time had the disadvantage of producing less accurate results. For instance, quantizing the entire image once creates too many clusters in some places and too few in others, and it causes false edges in smoothly varying regions. Replacing the EMD with a faster distance measure also means limiting our representation to color histograms, or giving up the saturating distance measure, or both.

In addition, we have noticed that the colors produced through vector quantization do not always match mental models of color. As was mentioned earlier, CIE-Lab was designed to measure distances between large, uniform color patches, not individual pixels. Other representations derived from CIE-Lab [54] may better take these effects

Fig. 19. Corner and edge information is complementary. Is it possible to extract a boundary that contains both at the same time?

into account. A separate difficulty is that the transformation from RGB to CIE-Lab or other perceptual color spaces has nonuniform effects on noise. Experiments showed that using gray-scale images with this edge detector works better in RGB than in CIE-Lab [7]. A recent alternative proposed in [55] clustered in RGB but computed distances using CIE-Lab.

We have identified two key areas that would have the most impact on feature detection as described here. The first is stability in producing color clusters, which would greatly decrease the running time. The second is the generalization of distributions beyond what we have proposed.

Each pixel is part of approximately $9\pi\sigma^2$ circular windows. A "stable" pixel is one that, regardless of where it appears relative to the center of a window, is always represented by the same color cluster after vector quantization. Unfortunately, pixels near an edge are usually "unstable" and are mapped to different colors at different times. The result is edges that appear noisy compared to Canny's since the mean color of a neighborhood changes smoothly as the operator window translates.

Two overlapping windows can be considered "stable" if their perceptual distance is due only to those pixels not shared by both. Currently, this is not guaranteed because each window is quantized separately. An algorithm that could guarantee stability between windows in both the horizontal and vertical directions while producing perceptually accurate color clusters would likely improve the results and run much faster.

Finally, we note that the next obvious generalization in the representation of an image neighborhood would include spatial information about each cluster or each vector, for example by including the x- and y-coordinates in the vector itself. Neither color nor texture alone has proven adequate to detect all features, but the combination of the two has the potential to take feature detection even farther.

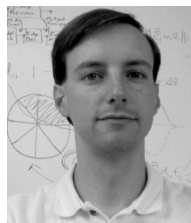**Note.** We have made the code for these detectors available at http://robotics.stanford.edu/~ruzon/compass/.

REFERENCES

[1] V.S. Nalwa and T.O. Binford, "On Detecting Edges," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 699-714, Nov. 1986.

[2] Y.G. Leclerc and S.W. Zucker, "The Local Structure of Image Discontinuities in One Dimension," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 3, pp. 341-355, May 1987.

[3] S.-J. Wang and T.O. Binford, "Generic, Model-Based Estimation and Detection of Discontinuities in Image Surfaces," *Proc. Image Understanding Workshop,* vol. II, pp. 113-116, Nov. 1994.

[4] T.O. Binford, P.-C. Chiang, "Generic, Model-Based Edge Estimation in the Image Surface," *Proc. Image Understanding Workshop,* vol. II, pp. 1237-1246, May 1997.

[5] M. Ruzon and C. Tomasi, "Color Edge Detection with the Compass Operator," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 160-166, June 1999.

[6] M. Ruzon and C. Tomasi, "Corner Detection in Textured Color Images," *IEEE Int'l Conf. Computer Vision,* vol. II, pp. 1039-1045, Sept. 1999.

[7] M. Ruzon, "Early Vision Using Distributions," PhD thesis, Computer Science Dept., Stanford Univ., Stanford, Calif., Apr. 2000.

[8] R. Nevatia, "A Color Edge Detector and Its Use in Scene Segmentation," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 7, no. 11, pp. 820-826, Nov. 1977.

[9] M. Hueckel, "An Operator which Locates Edges in Digitized Pictures," *J. ACM,* vol. 18, no. 1, pp. 113-125, Jan. 1971.

[10] A. Shiozaki, "Edge Extraction Using Entropy Operator," *Computer Vision, Graphics, and Image Processing,* vol. 36, no. 1, pp. 1-9, Oct. 1986.

[11] M.E. Malowany and A.S. Malowany, "Color-Edge Detectors for a VLSI Convolver," *Proc. Int'l Soc. for Optical Eng.,* vol. 1199, pp. 1116-1126, 1989.

[12] T. Carron and P. Lambert, "Color Edge Detector Using Jointly Hue, Saturation, and Intensity," *IEEE Int'l Conf. Image Processing,* vol. 3, pp. 977-981, Nov. 1994.

[13] T. Carron and P. Lambert, "Fuzzy Color Edge Extraction by Inference Rules: Quantitative Study and Evaluation of Performances," *IEEE Int'l Conf. Image Processing,* vol. 2, pp. 181-184, Oct. 1995.

[14] A.R. Weeks and H.R. Myler, "Edge Detection of Color Images Using the HSL Color Space," *Proc. Int'l Soc. for Optical Eng.,* vol. 2424, pp. 291-301, Feb. 1995.

[15] R.A. Salinas, C. Richardson, M.A. Abidi, and R.C. Gonzalez, "Data Fusion: Color Edge Detection and Surface Reconstruction Through Regularization," *IEEE Trans. Industrial Electronics,* vol. 43, no. 3, pp. 355-363, June 1996.

[16] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, Nov. 1986.

[17] G. Robinson, "Color Edge Detection," *Optical Eng.,* vol. 16, no. 5, pp. 479-484, Sept. 1977.

[18] S. Di Zenzo, "A Note on the Gradient of a Multi-Image," *Computer Vision, Graphics, and Image Processing,* vol. 33, no. 1, pp. 116-125, Jan. 1986.

[19] A. Cumani, "Edge Detection in Multispectral Images," *CVGIP: Graphical Models and Image Processing,* vol. 53, no. 1, pp. 40-51, Jan. 1991.

[20] C. Drewniok, "Multispectral Edge-Detection—Some Experiments on Data from Landsat-TM," *Int'l J. Remote Sensing,* vol. 15, no. 18, pp. 3743-3766, Dec. 1994.

[21] E. Saber, A.M. Tekalp, and G. Bozdagi, "Fusion of Color and Edge Information for Improved Segmentation and Edge Linking," *Image and Vision Computing,* vol. 15, no. 10, pp. 769-780, Oct. 1997.

[22] M. Chapron, "A Chromatic Contour Detector Based on Abrupt Change Techniques," *IEEE Int'l Conf. Image Processing,* vol. III, pp. 18-21, Oct. 1997.

[23] A. Moghaddamzadeh, D. Goldman, and N. Bourbakis, "Fuzzy-Like Approach for Smoothing and Edge Detection in Color Images," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 12, no. 6, pp. 801, Sept. 1998.

[24] W.H. Tsang and P.W.M. Tsang, "Suppression of False Edge-Detection Due to Specular Reflection in Color Images," *Pattern Recognition Letters,* vol. 18, no. 2, pp. 165-171, Feb. 1997.

[25] L. Macaire, V. Ultre, and J.G. Postaire, "Determination of Compatibility Coefficients for Colour Edge Detection by Relaxation," *IEEE Int'l Conf. Image Processing,* vol. III, pp. 1045-1048, Sept. 1996.

[26] J. Scharcanski and A.N. Venetsanopoulos, "Edge Detection of Color Images Using Directional Operators," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 7, no. 2, pp. 396-401, Apr. 1997.

[27] R. Machuca and K. Phillips, "Applications of Vector Fields to Image Processing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, no. 3, pp. 316-329, May 1983.

[28] T.L. Huntsberger and M.F. Descalzi, "Color Edge Detection," *Pattern Recognition Letters,* vol. 3, pp. 205-209, 1985.

[29] M. Pietikainen and D. Harwood, "Edge Information in Color Images Based on Histograms of Differences," *Proc. Int'l Conf. Pattern Recognition,* pp. 594-596, 1986.

[30] C.K. Yang and W.H. Tsai, "Reduction of Color Space Dimensionality by Moment-Preserving Thresholding and its Application for Edge-Detection in Color Images," *Pattern Recognition Letters,* vol. 17, no. 5, pp. 481-490, May 1996.

[31] H. Tao and T. Huang, "Color Image Edge Detection Using Cluster Analysis," *IEEE Int'l Conf. Image Processing,* vol. I, pp. 834-837, 1997.

[32] P.M. Djuric and J.K. Fwu, "On the Detection of Edges in Vector Images," *IEEE Trans. Image Processing,* vol. 6, no. 11, pp. 1595-1601, Nov. 1997.

[33] P.E. Trahanias and A.N. Venetsanopoulos, "Vector Order-Statistics Operators as Color Edge Detectors," *IEEE Trans. Systems, Man and Cybernetics,* vol. B-26, no. 1, pp. 135-143, Feb. 1996.

[34] Y. Rubner, "Perceptual Metrics for Image Database Navigation," PhD thesis, Computer Science Dept., Stanford Univ., Stanford, Calif., May 1999.

[35] M.T. Orchard and C.A. Bouman, "Color Quantization of Images," *IEEE Trans. Signal Processing,* vol. 39, no. 12, pp. 2677-2690, Dec. 1991.

[36] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice.* Reading, Mass.: Addison-Wesley, 1990.

[37] G. Wyszecki and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae.* New York: John Wiley and Sons, 1982.

[38] R.N. Shepard, "Toward a Universal Law of Generalization for Psychological Science," *Science,* vol. 237, no. 4820, pp. 1317-23, Sept. 1987.

[39] V.I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Mathematics—Doklady,* vol. 10, pp. 707-710, 1966.

[40] M.J. Swain and D.H. Ballard, "Color indexing," *Int'l J. Computer Vision,* vol. 7, no. 1, pp. 11-32, Nov. 1991.

[41] W. Niblack, R. Barber, W. Equitz, M.D. Flickner, E.H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "Querying Images by Content, Using Color, Texture, and Shape," *Proc. Int'l Soc. for Optical Eng.,* vol. 1908, pp. 173-187, Apr. 1993.

[42] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 850-863, Sept. 1993.

[43] Y. Rubner, C. Tomasi, and L.J. Guibas, "A Metric for Distributions with Applications to Image Databases," *IEEE Int'l Conf. Computer Vision,* pp. 59-66, Jan. 1998.

[44] H.C. Shen and A.K.C. Wong, "Generalized Texture Representation and Metric," *Computer Vision, Graphics, and Image Processing,* vol. 23, no. 2, pp. 187-206, Aug. 1983.

[45] M. Werman, S. Peleg, and A. Rosenfeld, "A Distance Metric for Multidimensional Histograms," *Computer Vision, Graphics, and Image Processing,* vol. 32, no. 3, pp. 328-336, Dec. 1985.

[46] R.M. Dudley, "Distances of Probability Measures and Random Variables," *Annals of Math. Statistics,* vol. 39, no. 5, pp. 1563-1572, 1968.

[47] R.M. Gray, D.L. Neuhoff, and P.C. Shields, "A Generalization of Ornstein's $\bar{d}$ Distance with Applications to Information Theory," *Annals of Probability,* vol. 3, no. 2, pp. 315-328, Apr. 1975.

[48] R.L. Dobrushin, "Prescribing a System of Random Variables by Conditional Distributions," *Theory of Probability and its Applications,* vol. 15, no. 3, pp. 458-486, Sept. 1970.

[49] S. Cohen, "Finding Color and Shape Patterns in Images," PhD thesis, Computer Science Dept., Stanford Univ., Stanford, Calif., May 1999.

[50] R. Deriche, "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector," *Int'l J. Computer Vision,* vol. 1, no. 2, pp. 167-187, 1987.

[51] H. Moravec, "Towards Automatic Visual Obstacle Avoidance," *Proc. Int'l Joint Conf. Artificial Intelligence,* p. 584, 1977.

[52] S.M. Smith and J.M. Brady, "SUSAN—A New Approach to Low-Level Image-Processing," *Int'l J. Computer Vision,* vol. 23, no. 1, pp. 45-78, May 1997.

[53] R. Deriche and G. Giraudon, "A Computational Approach for Corner and Vertex Detection," *Int'l J. Computer Vision,* vol. 10, no. 2, pp. 101-124, Apr. 1993.

[54] X. Zhang and B.A. Wandell, "A Spatial Extension of CIELAB for Digital Color Image Reproduction," *Proc. Soc. Information Display,* pp. 731-734, May 1996.

[55] K.Y. Song, J. Kittler, and M. Petrou, "Defect Detection in Random Color Textures," *Image and Vision Computing,* vol. 14, no. 9, pp. 667-683, Oct. 1996.

Mark A. Ruzon received the BS degree in computer science in 1994 from the University of Southern California, Los Angeles, where he was valedictorian of the School of Engineering. He received the MS degree in 1997 and the PhD degree in 2000 from Stanford University, Stanford, California, both in computer science. His main research interest is in representational issues for computer vision. He now is a research engineer at Quindi Corporation, which produces digital video products.

Carlo Tomasi received the DSc degree in electrical engineering from the University of Padova, Padova, Italy, in 1986 and the PhD degree in computer science from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1991. In 1994, he became an assistant professor of computer science at Stanford University, Stanford, California. His research interests include structure from motion, stereo vision, image databases, immersive television, medical imaging, and modeling humans. He is a member of the IEEE.

▷ **For more information on this or any computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.