# Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues

David R. Martin, *Member*, *IEEE*, Charless C. Fowlkes, and Jitendra Malik, *Member*, *IEEE*

**Abstract**—The goal of this work is to accurately detect and localize boundaries in natural scenes using local image measurements. We formulate features that respond to characteristic changes in brightness, color, and texture associated with natural boundaries. In order to combine the information from these features in an optimal way, we train a classifier using human labeled images as ground truth. The output of this classifier provides the posterior probability of a boundary at each image location and orientation. We present precision-recall curves showing that the resulting detector significantly outperforms existing approaches. Our two main results are 1) that cue combination can be performed adequately with a simple linear model and 2) that a proper, explicit treatment of texture is required to detect boundaries in natural images.

**Index Terms**—Texture, supervised learning, cue combination, natural images, ground truth segmentation data set, boundary detection, boundary localization.

## 1 Introduction

Consider the images and human-marked boundaries shown in Fig. 1. How might we find these boundaries automatically?

We distinguish the problem of boundary detection from what is classically referred to as edge detection. A *boundary* is a contour in the image plane that represents a change in pixel ownership from one object or surface to another. In contrast, an *edge* is most often defined as an abrupt change in some low-level image feature such as brightness or color. Edge detection is thus one low-level technique that is commonly applied toward the goal of boundary detection. Another approach would be to recognize objects in the scene and use that high-level information to infer the boundary locations.

In this paper, we focus on what information is available in a local image patch like those shown in the first column of Fig. 2. Though these patches lack global context, it is clear to a human observer which contain boundaries and which do not. Our goal is to use features extracted from such an image patch to estimate the posterior probability of a boundary passing through the center point. A boundary model based on such local information is likely to be integral to any perceptual organization algorithm that operates on natural images, whether based on grouping pixels into regions [1], [2] or grouping edge fragments into contours [3], [4]. This paper is intentionally agnostic about how a local boundary model might be used in a system for performing a high-level visual task such as recognition.

The most common approach to local boundary detection is to look for discontinuities in image brightness. For example, the Canny detector [5] models boundaries as brightness step edges. The brightness profiles in the second column of Fig. 2 show that this is an inadequate model for boundaries in natural images where texture is a ubiquitous phenomenon. The Canny detector fires wildly inside textured regions where high-contrast edges are present, but no boundary exists. In addition, it is unable to detect the boundary between textured regions when there is only a subtle change in average image brightness.

A partial solution is provided by examining gradients at multiple orientations around a pixel. For example, a boundary detector based on the eigenspectrum of the spatially averaged second moment matrix can distinguish simple edges from the multiple incident edges that may occur inside texture. While this approach will suppress false positives in a limited class of textures, it will also suppress corners and contours bordering textured regions.

The significant problems with simple brightness edge models have lead researchers to develop more complex detectors that look for boundaries defined by changes in texture, e.g., [6], [7]. While these work well on the pure texture-texture boundaries provided by synthetic Brodatz mosaics, they have problems in the vicinity of simple brightness boundaries. Texture descriptors computed over local windows that straddle a boundary have different statistics from windows contained in either of the neighboring regions. This inevitably results in either doubly-detected boundaries or thin halo-like regions along contours (e.g., see images in [6], [8], [9]). Just as a brightness edge model does not detect texture boundaries, a pure texture model does not detect brightness edges effectively.

- *D.R. Martin is with the Computer Science Department, 460 Fulton Hall, Boston College, 140 Commonwealth Ave., Chestnut Hill, MA 02167. E-mail: dmartin@cs.bc.edu.*
- *C.C. Fowlkes and J. Malik are with the Computer Science Division, Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720. E-mail: {fowlkes, malik}@eecs.berkeley.edu.*

*Manuscript received 21 Jan. 2003; revised 25 July 2003; accepted 7 Aug. 2003. Recommended for acceptance by W. Freeman.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118170.*

Fig. 1. Example images and human-marked segment boundaries. Each image shows multiple (4-8) human segmentations. The pixels are darker where more humans marked a boundary. Details of how this ground-truth data was collected are discussed in Section 3.

Clearly, boundaries in natural images can be marked by joint changes in several cues including brightness, color, and texture. Evidence from psychophysics [10] suggests that humans make combined use of multiple cues to improve their detection and localization of boundaries. There has been limited work in computational vision on addressing the difficult problem of cue combination. For example, the authors of [2] associate a measure of texturedness with each point in an image in order to suppress contour processing in textured regions and vice versa. However, their solution is full of ad hoc design decisions and hand chosen parameters.

In this paper, we provide a more principled approach to cue combination by framing the task as a supervised learning problem. A large data set of natural images that have been manually segmented by multiple human subjects [11] provides the ground truth label for each pixel as being on or off-boundary. The task is then to model the probability of a pixel being on-boundary conditioned on some set of local image features. This sort of quantitative approach to learning and evaluating boundary detectors is similar in spirit to the work of Konishi et al. [12] who used the Sowerby data set of English countryside scenes. Our work is distinguished by an explicit treatment of texture, enabling superior performance on a more diverse collection of natural images.

By modeling texture and combining various local cues in a statistically optimal manner, we demonstrate a marked improvement over the state of the art in boundary detection. Fig. 3 shows the performance of our detector compared to the Canny detector, a detector based on the second moment matrix used by Konishi et. al. [12], and the human subjects. The remainder of the paper will present how this improvement was achieved. In Section 2, we describe the local brightness, color, and texture features used as input to our algorithm. In Section 3, we present our training and testing methodology and the data set of 12,000 human segmentations that provide the ground truth data. We apply this methodology in Section 4 to optimize each local feature independently, and in Section 5 to perform cue combination. Section 6 presents a quantitative comparison of our method to existing boundary detection methods. We conclude in Section 7.

## 2 Image Features

Our approach to boundary detection is to look at each pixel for local discontinuities in several feature channels, over a range of orientations and scales. We will consider two brightness features (oriented energy and brightness gradient), one color feature (color gradient), and one texture feature (texture gradient). Each of these features has free parameters that we will calibrate with respect to the training data.

### 2.1 Oriented Energy

In natural images, brightness edges are more than simple steps. Phenomena such as specularities, mutual illumination, and shading result in composite intensity profiles consisting of steps, peaks, and roofs. The oriented energy (OE) approach [13] can be used to detect and localize these composite edges [14]. OE is defined as:

$$\mathrm{OE}_{\theta,\sigma} = (I * f^e_{\theta,\sigma})^2 + (I * f^o_{\theta,\sigma})^2, \tag{1}$$

where $f^e_{\theta,\sigma}$ and $f^o_{\theta,\sigma}$ are a quadrature pair of even and odd-symmetric filters at orientation $\theta$ and scale $\sigma$. Our even-symmetric filter is a Gaussian second-derivative, and the corresponding odd-symmetric filter is its Hilbert transform. $\mathrm{OE}_{\theta,\sigma}$ has maximum response for contours at orientation $\theta$. The filters are elongated by a ratio of 3:1 along the putative boundary direction.

### 2.2 Gradient-Based Features

We include the oriented energy feature in our analysis because it is the standard means of detecting brightness edges in images. For more complex features, we introduce a gradient-based paradigm that we use for detecting local changes in color and texture, as well as brightness. At a location $(x, y)$ in the image, draw a circle of radius $r$, and divide it along the diameter at orientation $\theta$. The gradient function $G(x, y, \theta, r)$ compares the contents of the two resulting disc halves. A large difference between the disc halves indicates a discontinuity in the image along the disc's diameter.

How shall we describe and compare the two half-disc regions for each cue? Successful approaches to this problem have commonly made use color and texture features based on the empirical distribution of pixel values averaged over

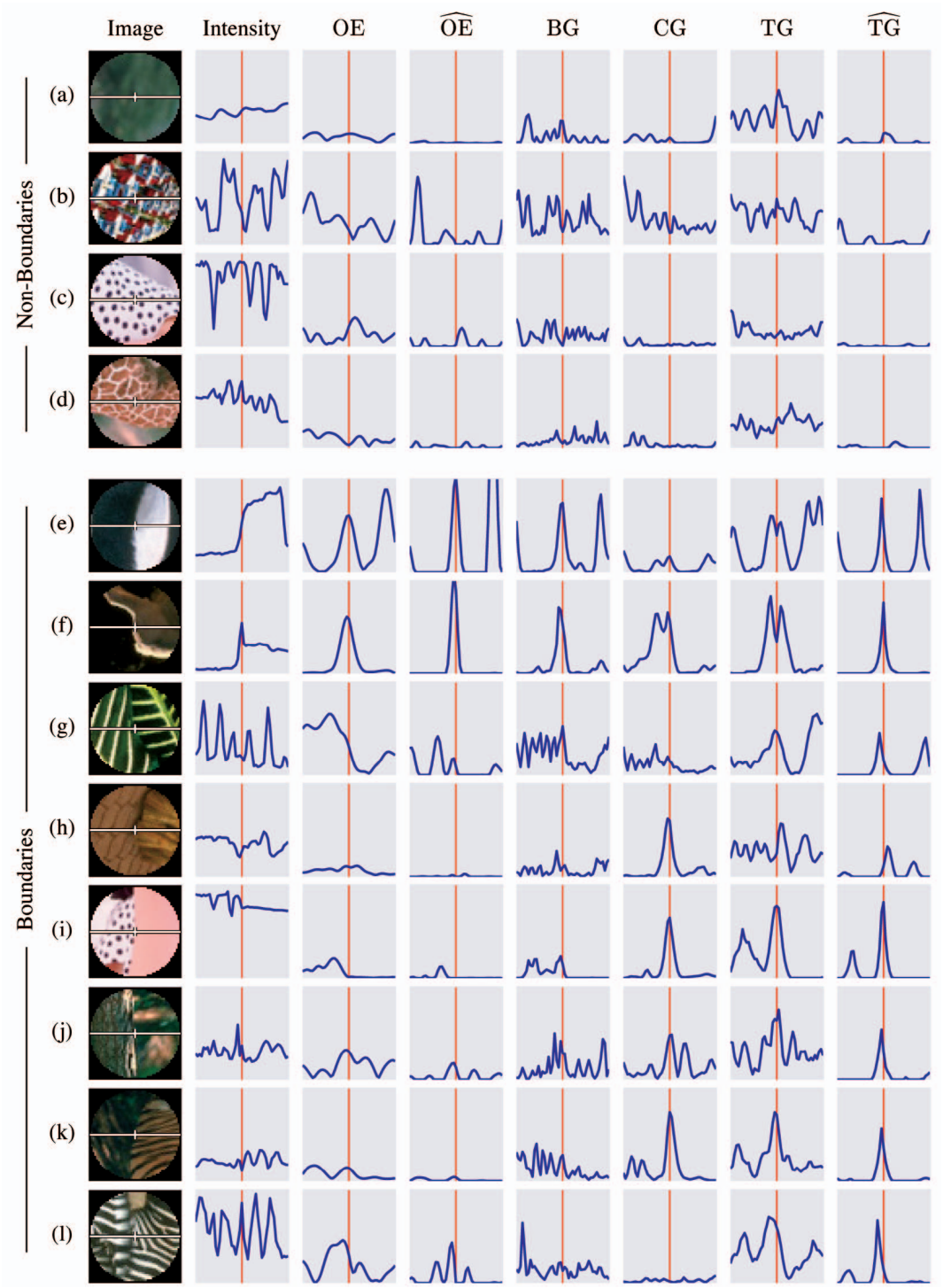


Fig. 2. Local Image Features. In each row, the first panel shows an image patch. The following panels show feature profiles along the patch's horizontal diameter. The features are raw image intensity, oriented energy $\mathrm{OE}$, brightness gradient $\mathrm{BG}$, color gradient $\mathrm{CG}$, raw texture gradient $\mathrm{TG}$, and localized texture gradient $\widehat{\mathrm{TG}}$. The vertical line in each profile marks the patch center. The scale of each feature has been chosen to maximize performance on the set of training images—2 percent of the image diagonal (5.7 pixels) for OE, CG, and TG, and 1 percent of the image diagonal (3 pixels) for BG. The challenge is to combine these features in order to detect and localize boundaries.

some neighborhood. Distributions of color in perceptual color spaces have been used successfully as region descriptors in the QBIC [15] and Blobworld [8] image retrieval systems. In addition, the compass operator of Ruzon and Tomasi [16], [17] uses color histogram comparisons to find corners and edges in color images. For texture analysis, there is an emerging consensus that an image should first be convolved with a bank of filters tuned to various orientations and spatial frequencies [18], [19]. The empirical distribution of filter responses has been demonstrated to be a powerful feature in both texture synthesis [20] and texture discrimination [21].

For brightness and color gradient features, we bin kernel density estimates of the distributions of pixel luminance

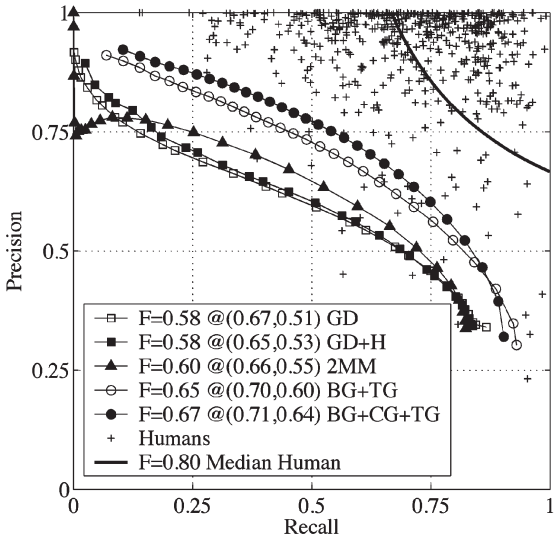


Fig. 3. Two Decades of Boundary Detection. The performance of our boundary detector compared to classical boundary detection methods and to the human subjects' performance. A precision-recall curve is shown for each of five boundary detectors: 1) Gaussian derivative (GD), 2) Gaussian derivative with hysteresis thresholding (GD+H), the Canny detector, 3) A detector based on the second moment matrix (2MM), 4) our gray-scale detector that combines brightness and texture (BG+TG), and 5) our color detector that combines brightness, color, and texture (BG+CG+TG). Each detector is represented by its *precision-recall* curve, which measures the trade off between accuracy and noise as the detector's threshold varies. Shown in the caption is each curve's F-measure, valued from zero to one. The F-measure is a summary statistic for a precision-recall curve. The points marked by a "+" on the plot show the precision and recall of each ground truth human segmentation when compared to the other humans. The median F-measure for the human subjects is 0.80. The solid curve shows the F=0.80 curve, representing the frontier of human performance for this task.

and chrominance in each disc half. The binning was done by sampling each Gaussian kernel out to $2\sigma$ at a rate ensuring at least two samples per bin. For the texture gradient, we compute histograms of vector quantized filter outputs in each disc half. In all three cases, the half-disc regions are described by histograms, which we compare with the $\chi^2$ histogram difference operator [22]:

$$\chi^2(g,h) = \frac{1}{2}\sum \frac{(g_i - h_i)^2}{g_i + h_i}. \quad (2)$$

The brightness, color, and texture gradient features therefore encode, respectively, changes in the local distributions of luminance, chrominance, and filter responses.

Each gradient computation shares the step of computing a histogram difference at eight orientations and three half-octave scales at each pixel.[1] In the following sections, we discuss in detail the possible design choices for representing and comparing color, brightness, and texture.

### 2.2.1 Brightness and Color Gradients

There are two common approaches to characterizing the difference between the color distributions of sets of pixels. The first is based on density estimation using histograms. Both QBIC and Blobworld use fully three-dimensional color histograms as region features and compare histograms using a similarity measure such as $L^1$ norm, $\chi^2$ difference, or some quadratic form. Blobworld smooths the histograms to prevent the aliasing of similar colors, while QBIC models the perceptual distance between bins explicitly.[2] A second common approach avoids quantization artifacts by using the *Mallows* [23] or *Earth Mover's distance* (EMD) [24] to compare color distributions. In addition, the EMD explicitly accounts for the "ground distance" between points in the color space. This is a desirable property for data living in a perceptual color space where nearby points appear perceptually similar. However, once colors in such a space are further apart than some degree of separation, they tend to appear "equally distant" to a human observer. Ruzon and Tomasi use an attenuated EMD to model this perceptual roll-off, but the EMD remains computationally expensive. For one-dimensional data, efficient computation is possible using sorting. In higher dimensions, however, one must explicitly solve an assignment problem, resulting in a considerable increase in computational complexity.

We would like a way to model the color distribution accurately with respect to human perception, while retaining computationally feasibility. Our approach is based on binning kernel density estimates of the color distribution in CIELAB using a Gaussian kernel, and comparing histograms with the $\chi^2$ difference. The $\chi^2$ histogram difference does not make use of the perceptual distance between bin centers. Therefore, without smoothing, perceptually similar colors can produce disproportionately large $\chi^2$ differences. Because the distance between points in CIELAB space is perceptually meaningful in a local neighborhood, binning a kernel density estimate whose kernel bandwidth $\sigma$ matches the scale of this neighborhood means that perceptually similar colors will have similar histogram contributions. Beyond this scale, where color differences are perceptually incommensurate, $\chi^2$ will regard them as equally different. We believe this combination of a kernel density estimate in CIELAB with the $\chi^2$ histogram difference is a good match to the structure of human color perception.

For the brightness gradient we compute histograms of L* values. The color gradient presents additional challenges for density estimation because the pixel values are in the 2D space (a* and b*). When using 2D kernels and 2D histograms one typically reduces both the number of kernel samples and the number of bins in order to keep the computational cost reasonable. However, this compromises the quality of the density estimate.

Rather than compute the joint gradient $\mathrm{CG}^{ab}$, we compute marginal color gradients for a* and b* and take the full color gradient to be the sum of the corresponding marginal gradients: $\mathrm{CG}^{a+b} = \mathrm{CG}^{a} + \mathrm{CG}^{b}$. This is motivated by the fact that the a* and b* channels correspond to the perceptually orthogonal red-green and yellow-blue color

1. A naive implementation would involve much redundant computation. Appendix A presents efficient algorithms for computing the gradient features.

2. The quadratic form distance function used in QBIC is $d(g,h) = (g-h)^T A(g-h)$, where $g$ and $h$ are the histograms to compare, and $A$ is a matrix giving the similarity $A_{ij}$ between two bins $i$ and $j$. The QBIC authors indicate that this measure is superior for their task. We will not consider this histogram similarity function because it is computationally expensive, difficult to define $A$, and similar in spirit to the Earth Mover's distance.

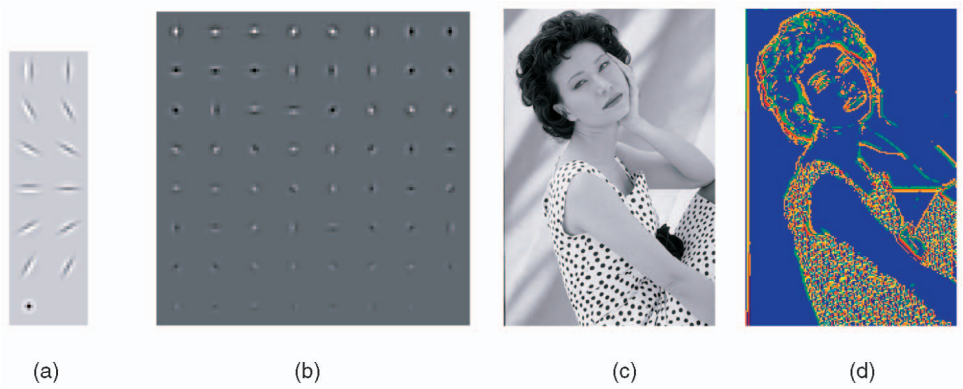


Fig. 4. Computing Textons. (a) Filter Bank: The 13-element filter bank used for computing textons. (b) Universal Textons: Example universal textons computed from the 200 training images, sorted by L1 norm for display purposes. (c) Image and (d) Texton Map: An image and its associated texton map. Texton quality is best with a single scale filter bank containing small filters. Each pixel produces a 13-element response to the filter bank, and these responses are clustered with k-means. In this example, using 200 images with k = 64 yields 64 universal textons. The textons identify basic structures such as steps, bars, and corners at various levels of contrast. If each pixel in the image shown in (c) is assigned to the nearest texton and each texton is assigned a color, we obtain the texton map shown in (d). The elongated filters have 3:1 aspect, and the longer $\sigma$ was set to 0.7 percent of the image diagonal (about 2 pixels).

opponents found in the human visual system (see Palmer [25]). The comparison of $\mathrm{CG}^{ab}$ to $\mathrm{CG}^{a+b}$ is presented in Section 4.

### 2.2.2 Texture Gradient

In a manner analogous to the brightness and color gradient operators, we formulate a directional operator that measures the degree to which texture of scale $r$ varies at an image location $(x, y)$ in direction $\theta$. We compute the texture dissimilarity in the two halves of a disk of centered on a point and divided in two along a diameter. Oriented texture processing along these lines has been pursued by Rubner and Tomasi [6].

Fig. 4a shows the filter bank that we use for texture processing. It contains six pairs of elongated, oriented filters, as well as a center-surround filter. The oriented filters are in even/odd quadrature pairs, and are the same filters we used to compute oriented energy. The even-symmetric filter is a Gaussian second derivative, and the odd-symmetric filter is its Hilbert transform. The center-surround filter is a difference of Gaussians. The even and odd filter responses are not combined as they are in computing oriented energy. Instead, each filter produces a separate feature. To each pixel, we associate the vector of 13 filter responses centered at the pixel. Note that unlike [2], we do not contrast-normalize the filter responses for texture processing. Our experiments indicate that this type of normalization does not improve performance, as it appears to amplify noise more than signal.

Each disc half contains a set of filter response vectors which we can visualize as a cloud of points in a feature space with dimensionality equal to the number of filters. One can use the empirical distributions of these two point clouds as texture descriptors, and then compare the descriptors to get the value of the texture gradient.

Many questions arise regarding the details of this approach. Should the filter bank contain multiple scales, and what should the scales be? How should we compare the distributions of filter responses? Should we use the Earth Mover's distance, or should we estimate the distributions? If the latter, should we estimate marginal or joint distributions and with fixed or adaptive bins? How should we compare distributions—some $L^p$-norm or the $\chi^2$ difference? Puzicha et al. [21] evaluate a wide range of texture descriptors in this framework and examine many of these questions. We choose the approach developed in [2], which is based on the idea of *textons*.

The texton approach estimates the joint distribution of filter responses using adaptive bins. The filter response vectors are clustered using k-means. Each cluster defines a Voronoi cell in the space of joint filter responses, and the cluster centers define texture primitives. These texture primitives—the *textons*—are simply linear combinations of the filters. Fig. 4b shows example textons for $k = 64$ computed over the 200 images in the training set. After the textons have been identified, each pixel is assigned to the nearest texton. The texture dissimilarities can then be computed by comparing the histograms of texton labels in the two disc halves. Figs. 4c and 4d shows an image and the associated *texton map*, where each pixel has been labeled with the nearest texton. Some questions remain, namely, what images to use to compute the textons, the choice of $k$, the procedure for computing the histograms, and the histogram comparison measure.

For computing textons, we can use a large, diverse collection of images in order to discover a set of *universal textons*. Alternately, one can compute *image-specific textons* by separately clustering filter responses in each test image. The optimal number of textons, $k$, depends on this choice between universal and image-specific as well as the scale $r$ of the texture gradient operator and the size of the image. Experiments exploring both of these issues are presented in Section 4.

To compute the texton histograms, we use hard binning without smoothing. It is possible to do soft binning in the texton framework by considering a pixel's distance to each bin center. However, this type of soft binning is computationally expensive and, in our experiments, it has not proved worthwhile. It seems likely that hard binning is not a problem because adjacent pixels have correlated filter responses due to the spatial extent of the

filters. Consequently, the data is already somewhat smoothed, and pixels in a disc are likely to cover fewer bins ensuring more samples per bin.

Finally, the $\chi^2$ difference is not the only viable measure of histogram difference for this task. Both Puzicha et al. [22] and Levina [26] evaluate various methods for comparing texture distributions, including L1 norm, $\chi^2$ difference, and the Mallows or Earth Mover's distance. The optimal difference measure, however, depends on the task (matching or discrimination) and on the images used (Brodatz patches or natural images). Our experiments show that for local boundary detection in natural images, the $\chi^2$ difference is marginally superior to the L1 norm, and significantly better than the Mallows distance.

### 2.3 Localization

The underlying function of boundary existence that we are trying to learn is tightly peaked around the location of image boundaries marked by humans. In contrast, Fig. 2 shows that the features we have discussed so far don't have this structure. By nature of the fact that they pool information over some support, they produce smooth, spatially extended outputs. Since each pixel is classified independently, spatially extended features are problematic for a classifier, as both on-boundary pixels and nearby off-boundary pixels will have large gradient values.

The texture gradient is particularly prone to this effect due to its large support. In addition, the TG produces multiple detections in the vicinity of brightness edges. The bands of textons present along such edges often produce a larger TG response on each side of the edge than directly on the edge. This double-peak problem is ubiquitous in texture edge detection and segmentation work [6], [8], [9], where it produces double detections of edges and sliver regions along region boundaries. We are aware of no work that directly addresses this phenomenon. Nonmaxima suppression is typically used to narrow extended responses, but multiple detections requires a more general solution. We exploit the symmetric nature of the texture gradient response to both localize the edge accurately and eliminate the double detections.

To make the spatial structure of boundaries available to the classifier, we transform the raw feature signals in order to emphasize local maxima in a manner that simultaneously smooths out multiple detections. Given a feature $f(x)$ defined over spatial coordinate $x$ orthogonal to the edge orientation, consider the derived feature $\hat{f}(x) = f(x)/d(x)$, where $d(x) = -|f'(x)|/f''(x)$ is the first-order approximation of the distance to the nearest maximum of $f(x)$. We use the smoothed and stabilized version

$$\hat{f}(x) = \tilde{f}(x) \cdot \left( \frac{-f''(x)}{|f'(x)| + \epsilon} \right) \tag{3}$$

with $\epsilon$ chosen to optimize the performance of the feature. By incorporating the $1/d(x)$ localization term, $\hat{f}(x)$ will have narrower peaks than the raw $f(x)$. $\tilde{f}(x)$ is a smoothed estimate of the underlying gradient signal that eliminates the double peaks.

To robustly estimate the directional derivatives and the smoothed signal, we fit a cylindrical parabola over a 2D circular window of radius $r$ centered at each pixel.[3] The axis of the parabolic cylinder is constrained to lay parallel to the image plane and encodes the edge location and orientation; the height encodes the edge intensity; and the curvature of the parabola encodes localization uncertainty. We project the data points inside the circular fit window onto the plane orthogonal to both the image plane and the edge orientation, so that the fit is performed on a 1D function. The least squares parabolic fit $ax^2 + bx + c$ provides directly the signal derivatives as $f''(x) = 2a$ and $f'(x) = b$, as well as $\tilde{f}(x) = c$. Thus, the localization function becomes $\hat{f} = -(2c^+a^+)/(|b| + \epsilon)$, where $c$ and $a$ are half-wave rectified. This rectification is required to avoid nonsensical sign changes in the signal when $c$ and $a$ are multiplied together.

The last two columns of Fig. 2 show the result of applying this transformation to the texture gradient. The effect is to reduce noise, tightly localize the boundaries, and coalesce double detections. We found that the localization procedure does not improve the brightness and color gradient features so our final feature set consists of $\{\widehat{OE}, BG, CG, \widehat{TG}\}$, each at eight orientations and three half-octave scales.

## 3 Evaluation Methodology

Our system will ultimately combine the cues of the previous section into a single function $P_b(x, y, \theta)$ that gives the posterior probability of a boundary at each pixel $(x, y)$ and orientation $\theta$. In order to optimize the parameters of this system and compare it to other systems, we need a methodology for judging the quality of a boundary detector. We formulate boundary-detection as a classification problem of discriminating nonboundary from boundary pixels and apply the *precision-recall* framework using human-marked boundaries from the Berkeley Segmentation Dataset [11] as ground truth.

The segmentation data set contains 5-10 segmentations for each of 1,000 images. The instructions to subjects were brief:

> You will be presented a photographic image. Divide the image into some number of segments, where the segments represent "things" or "parts of things" in the scene. The number of segments is up to you, as it depends on the image. Something between 2 and 30 is likely to be appropriate. It is important that all of the segments have approximately equal importance.

Fig. 1 demonstrates the high degree of consistency between different human subjects. Additional details on the data set construction may be found in [11]. In addition, the data set can be downloaded from the Internet [27] along with code for running our boundary detection and segmentation benchmark. We use 200 images and associated segmentations as the training data, and the next 100 images and associated segmentations as the test data set.

Our evaluation measure—the precision-recall curve—is a parametric curve that captures the trade off between

3. Windowed parabolic fitting is known as second-order Savitsky-Golay filtering, or LOESS smoothing. We also considered Gaussian derivative filters $\{G_r, G'_r, G''_r\}$ to estimate $\{f_r, f'_r, f''_r\}$ with similar results.

accuracy and noise as the detector threshold varies. *Precision* is the fraction of detections that are true positives rather than false positives, while *recall* is the fraction of true positives that are detected rather than missed. In probabilistic terms, precision is the probability that the detector's signal is valid, and recall is the probability that the ground truth data was detected.

Precision-recall curves are a standard evaluation technique in the information retrieval community [28] and were first used for evaluating edge detectors by Abdou and Pratt [29]. A similar approach was taken by Bowyer et al. [30] for boundary detector evaluation with receiver operating characteristic (ROC) curves. The axes for an ROC curve are *fallout* and *recall*. Recall, or *hit rate*, is the same as above. Fallout, or *false alarm rate*, is the probability that a true negative was labeled a false positive.

Although ROC and PR curves qualitatively show the same trade off between misses and false positives, ROC curves are not appropriate for quantifying boundary detection. Fallout is not a meaningful quantity for a boundary detector since it depends on the size of pixels. If we increase the image resolution by a factor of $n$, the number of pixels grows as $n^2$. Since boundaries are 1D (or at least have a fractal dimension less than 2) the number of true negatives will grow as $n^2$ while the number true positives will grow as slow as $n$. Thus, the fallout will decline by as much as $1/n$. Precision does not have this problem since it is normalized by the number of positives rather than the number of true negatives.

Other methods of evaluating boundary detectors in a quantitative framework exist, such as the Chernoff information used by Konishi et al. [12]. Though the information theoretic approach approach can lead to a useful method for ranking algorithms relative to one another, it does not produce an intuitive performance measure.

The precision and recall measures are particularly meaningful in the context of boundary detection when we consider applications that make use of boundary maps, such as stereo or object recognition. It is reasonable to characterize higher level processing in terms of how much true signal is required to succeed R (recall), and how much noise can be tolerated P (precision). A particular application can define a relative cost $\alpha$ between these quantities, which focuses attention at a specific point on the precision-recall curve. The *F-measure* [28], defined as

$$F = PR/(\alpha R + (1-\alpha)P) \tag{4}$$

captures this trade off as the weighted harmonic mean of $P$ and $R$. The location of the maximum F-measure along the curve provides the optimal detector threshold for the application given $\alpha$, which we set to 0.5 in our experiments.

Precision and recall are appealing measures, but to compute them we must determine which true positives are correctly detected, and which detections are false. Each point on the precision-recall curve is computed from the detector's output at a particular threshold. In addition, we have binary boundary maps as ground truth from the human subjects. For the moment, let us consider how to compute the precision and recall of a single thresholded machine boundary map given a single human boundary map. One could simply correspond coincident boundary pixels and declare all unmatched pixels either false positives or misses. However, this approach would not tolerate any localization error and would consequently overpenalize algorithms that generate usable, though slightly mislocalized boundaries. From Fig. 1, it is clear that the assignment of machine boundary pixels to ground truth boundaries must tolerate localization errors since even the ground truth data contains boundary localization errors.

The approach of [31] is to add a modicum of slop to the rigid correspondence procedure described above in order to permit small localization errors at the cost of permitting multiple detections. However, an *explicit* correspondence of machine and human boundary pixels is the only way to robustly count the hits, misses, and false positives that we need to compute precision and recall. In particular, it is important to compute the correspondence explicitly in order to penalize multiple detections. Single detection is one of the three goals of boundary detection formalized by Canny [5], the other two being a high detection rate and good localization.

The correspondence computation is detailed in Appendix B, which provides us the means of computing the precision and recall for a single human segmentation while permitting a controlled amount of localization error. The segmentation data set, however, provides multiple human segmentations for each image, so that the ground truth is defined by a collection of 5 to 10 human segmentations. Simply unioning the humans' boundary maps is not effective because of the localization errors present in the data set itself. A proper approach to combining the human boundary maps would likely require additional correspondences, or even estimating models of the humans' detection and localization error processes along with the hidden true signal.

Fortunately, we are able to finesse these issues in the following manner. First, we correspond the machine boundary map separately with each human map in turn. Only those machine boundary pixels that match no human boundary are counted as false positives. The hit rate is simply averaged over the different humans, so that to achieve perfect recall the machine boundary map must explain all of the human data. Our intention is that this approach to estimating precision and recall matches as closely as possible the intuitions one would have if scoring the outputs visually. In particular, all three desirable properties of a boundary detector—detection, localization, single detection—are encouraged by the method and visible in the results.

In summary, we have a method for describing the quality of a boundary detector that produces soft boundary maps of the form $P_b(x, y, \theta)$ or $P_b(x, y)$. For the latter, we take the maximum over $\theta$. Given the soft boundary image $P_b(x, y)$, we produce a precision-recall curve. Each point on the curve is computed independently by first thresholding $P_b$ to produce a binary boundary map and then matching this machine boundary map against each of the human boundary maps in the ground truth segmentation data set. The precision-recall curve is a rich descriptor of performance. When a single performance measure is required or is sufficient, precision and recall can be combined with the F-measure. The F-measure curve is usually unimodal, so the maximal F-measure may be reported as a summary of the detector's performance. We now turn to applying this

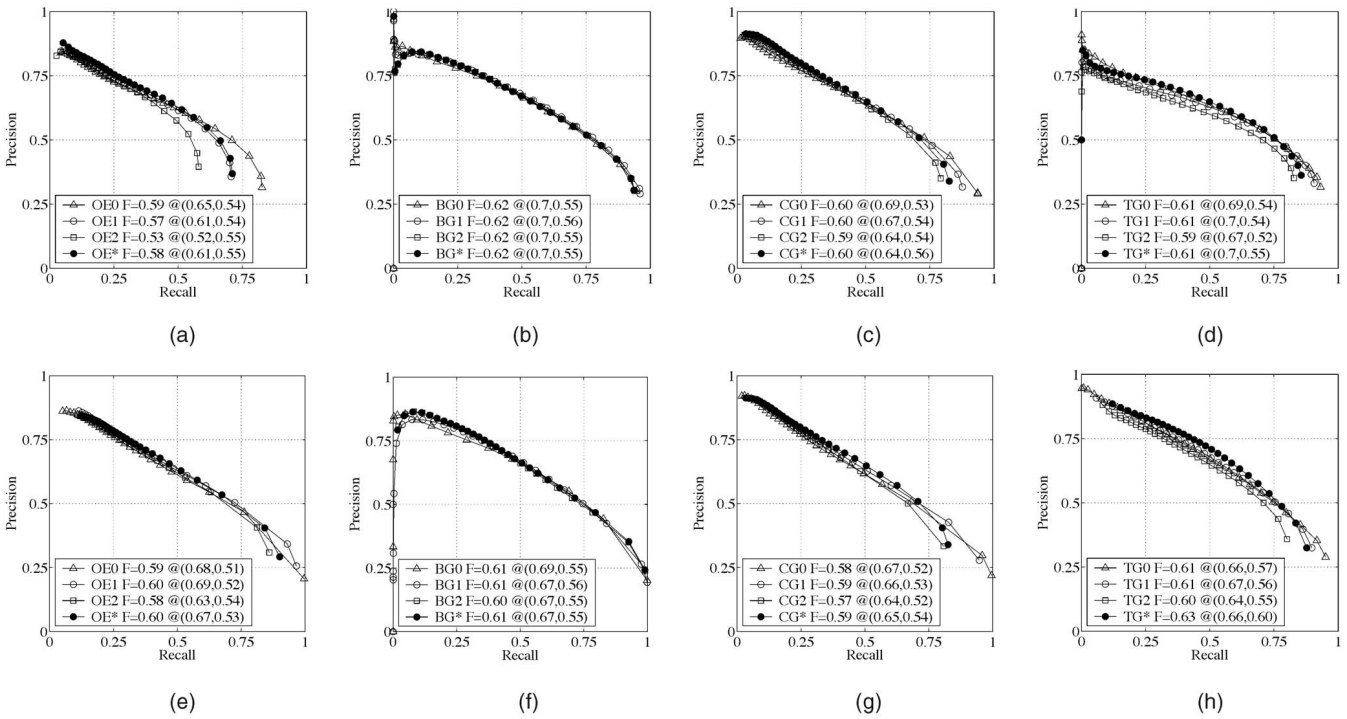


Fig. 5. (a) Raw OE, (b) Raw BG, (c) Raw CG, (d) Raw TG, (e) Localized OE, (f) Localized BG, (g) Localized CG, (h) Localized TG. Performance of raw and localized features (top and bottom rows, respectively). The precision and recall axes are defined in Section 3. Curves toward the top (lower noise) and right (more recovered signal) are better. Each curve is parameterized by $P_b$ and is scored by its maximal F-measure, the value and location of which are shown in the legend. Each panel in this figure shows four curves: one curve for each of three half-octave spaced scales of the feature, along with one curve for the combination of the three scales. The three scales are labeled smallest to largest as 0,1,2, and the combination of scales is indicated by a "*". The starting scales for OE, BG, CG, and TG are 1.4 percent, 0.75 percent, 1.4 percent, and 1.4 percent of the image diagonal, respectively. With the exception of Fig. 10, we use the logistic regression to model $P_b$. In this figure, we see that the localization procedure is marginally helpful for OE, unnecessary for BG and CG, and extremely helpful for TG. The performance gain for TG is due to the elimination of double-detections along with good localization, as is evident from Fig. 2. In addition, TG is the only feature for which there is benefit from combining scales. Note that each feature's $\epsilon$ and scale parameters were optimized against the training set using the precision-recall methodology.

evaluation methodology to optimizing our boundary detector, and comparing our approach to the standard methods.

## 4 Cue Optimization

Before combining the brightness, color, and texture cues into a single detector, we first optimize each cue individually. By applying coordinate ascent on each cue's parameters with high precision and recall as the objective, we can optimize each cue with respect to the ground truth data set so that no change in any single parameter improves performance. For space considerations, we do not present the complete set of experiments, rather only those that afford interesting observations.

Each of the four cues—oriented energy (OE), brightness gradient (BG), color gradient (CG), and texture gradient (TG)—has a scale parameter. In the case of OE, the scale $\sigma$ is the bandwidth of the quadrature filter pair. For the others, the scale $r$ is the radius of the disc. We determined the optimal one octave range for each cue. In units of percentage of the image diagonal, the ranges are 1.4 to 2.8 percent for OE, CG, and TG, and 0.75 to 1.5 percent for BG. These scales are optimal, independent of whether or not we use the localization procedure of Section 2.3. The middle scale always performs best, except in the case of raw OE where the largest scale is superior.

Fig. 5 shows the precision-recall (PR) curves for each cue at the optimal scales both with and without localization applied. In addition, each plot shows the PR curve for the combination of the three scales. Each curve is generated from a $P_b(x, y)$ function that is obtained by fitting a logistic model to the training data set. We evaluate the $P_b$ function on the test set to produce the $P_b(x, y)$ images from which the curve is generated. The $\epsilon$ for each cue's localization function was optimized separately to 0.01 for TG and 0.1 for all other cues. The figure shows that localization is not required for BG and CG, but helpful for both OE and TG. The localization function has two potential benefits. It narrows peaks in the signal, and it merges multiple detections. From Fig. 2, we see that the scale of OE is rather large so that localization is effective at narrowing the wide response. TG suffers from both multiple detections and a wide response, both of which are ameliorated by the localization procedure.

Fig. 6 shows our optimization of the kernel size used in the density estimation computations for BG and CG. For these features, we compare the distributions of pixel values in two half discs, whether those values are brightness (L*) or color (a*b*). First consider the color gradient $CG^{a+b}$ computed over the marginal distributions of a* and b*. With a disc radius ranging from four to eight pixels, kernels are critical in obtaining low-variance estimates of the distributions. In the figure, we vary the Gaussian kernel's sigma

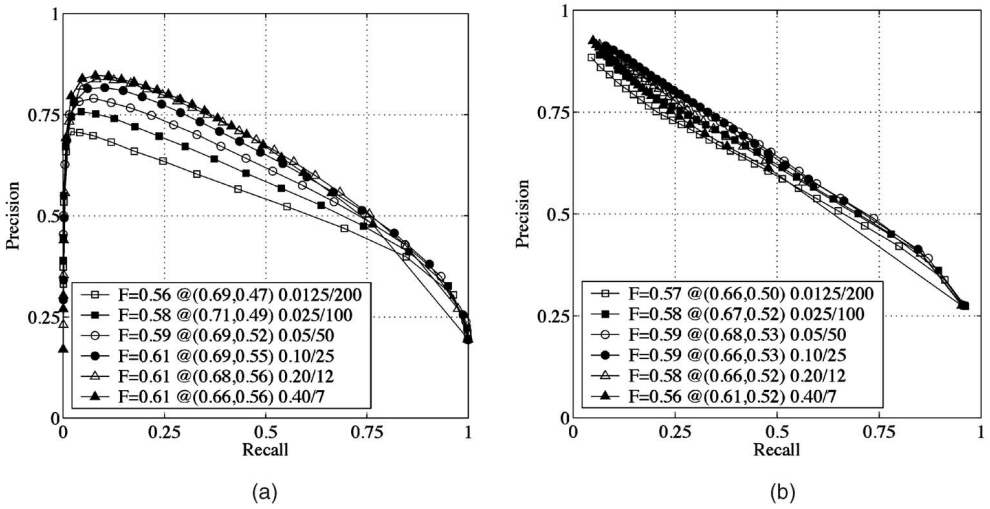


Fig. 6. (a) Brightness gradient. (b) Color gradient. Kernel bandwidth for BG and CG kernel density estimates. Both BG and CG operate by comparing the distributions of 1976 CIE L*a*b* pixel values in each half of a disc. We estimate the 1D distributions of L*, a*, and b* with histograms, but smoothing is required due to the small size of the discs. Each curve is labeled with $\sigma$ and bin count. The accessible ranges of L*, a*, and b* are scaled to $[0, 1]$. The kernel was clipped at $2\sigma$ and sampled at 23 points. The bin count was adjusted so that there would be no fewer than two samples per bin. The best values are $\sigma = 0.2$ for BG (12 bins) and $\sigma = 0.1$ for CG (25 bins).

from 1.25 to 40 percent of the diameter of the domain. In addition, the number of bins was varied inversely in order to keep the number of samples per bin constant, and above a minimum of two per bin. The kernel was clipped at $2\sigma$ and sampled at 23 points. The dominant PR curve on each plot indicates that the optimal parameter for BG is $\sigma = 0.2$ (with 12 bins) and $\sigma = 0.1$ for CG (with 25 bins).

The experiments in Fig. 6 used the separated version of the color gradient $\mathrm{CG}^{a+b}$ rather than the joint version $\mathrm{CG}^{ab}$. Fig. 7 shows the comparison between these two methods of computing the color gradient. Whether using a single scale of CG or multiple scales, the difference between $\mathrm{CG}^{a+b}$ and $\mathrm{CG}^{ab}$ is minimal. The joint approach is far more expensive computationally due to the additional dimension in the kernels and histograms. The number of bins in each dimension was kept constant at 25 for the comparison, so the computational costs differed by 25x, requiring tens of minutes for $\mathrm{CG}^{ab}$. If computational expense is kept constant, then the marginal method is superior because of the higher resolution afforded in the density estimate. In all cases, the marginal approach to computing the color gradient is preferable.

The texture gradient cue also has some additional parameters beyond $r$ and $\epsilon$ to tune, related to the texture representation and comparison. The purpose of TG is to quantify the difference in the distribution of filter responses in the two disc halves. Many design options abound as discussed in Section 2.2.2. For filters, we use the same even and odd-symmetric filters that use for oriented energy—a second derivative Gaussian and its Hilbert transform—at six orientations along with a center-surround DOG. We experimented with multiscale filter banks, but found agreement with Levina [26] that a single-scale filter bank at the smallest scale was preferable. Fig. 4a shows the filter bank we used for texture estimation. As for distribution estimation issues, we follow the *texton* approach of Malik et al. [2] which estimates the joint distribution with adaptive bins by clustering the filter responses with k-means, and compares histograms using the $\chi^2$ measure. We verified that none of $L^1$, $L^2$, or $L^\infty$ norm performs better. In addition, we determined that the Mallows distance computed on the marginal raw filter outputs performed poorly. The Mallows distance on the joint distribution is computationally infeasible, requiring the solution to a large assignment problem.

After settling on the approach of comparing texton histograms with the $\chi^2$ difference, we must choose between image-specific and universal textons as well as the number of textons (the $k$ parameter for k-means). For image-specific

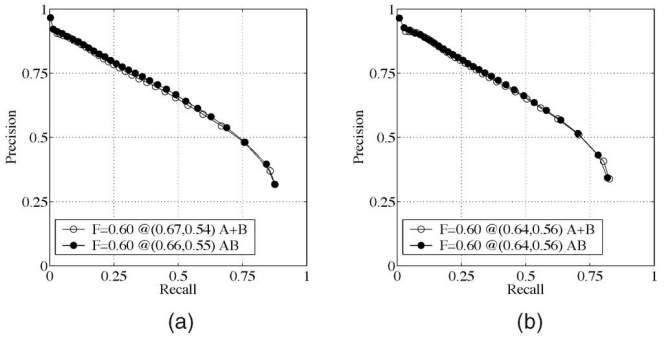


Fig. 7. Marginal versus Joint Estimates of CG. (a) CG middle scale: shows the middle scale of the color gradient and (b) CG combined scales: shows the three scales combined. Our inclination in estimating pixel color distributions was to estimate the 2D joint distribution of a* and b*. However, the 2D kernel density estimation proved to be computationally expensive. Since the a* and b* axes in the 1976 CIE L*a*b* color space were designed to mimic the blue-yellow green-red color opponency found the human visual cortex, one might expect the joint color distribution to contain little *perceptual* information not present in the marginal distributions of a* and b*. The curves labeled "AB" show the color gradient computed using the joint histogram ($\mathrm{CG}^{ab}$); the curves labeled "A+B" show the color gradient computed computed as $(\mathrm{CG}^{a} + \mathrm{CG}^{b})$. The number of bins in each dimension is 25 for both experiments, so that the $\mathrm{CG}^{ab}$ computation requires 25x more bins and 25x the compute time. The cue quality is virtually identical and, so, we adopt the marginal CG approach.

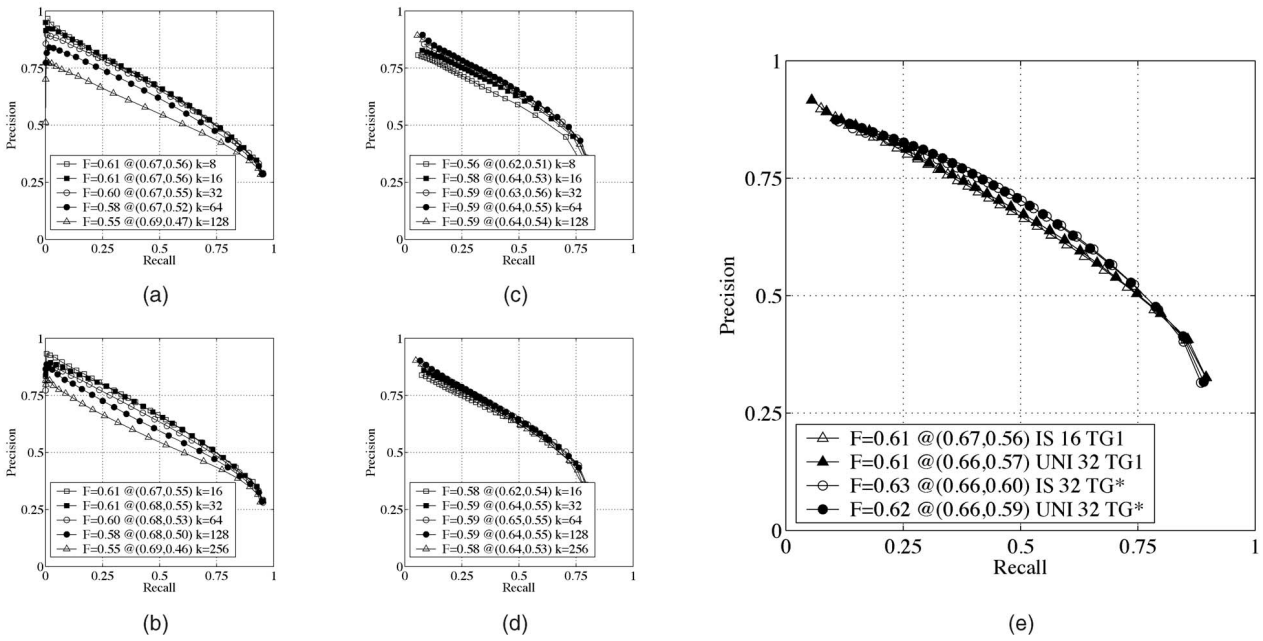


Fig. 8. (a) Image specific TG0, (b) Universal TG0, (c) Image specific TG2, (d) Universal TG2, (e) Image specific vs. universal. Image specific versus Universal textons. We can compute textons on a per-image basis or universally on a canonical image set. (a) and (c) show the performance of the small and large scales of TG for 8-128 image specific textons; (b) and (d) show the performance of the same TG scales for 16-256 universal textons; (e) shows the performance of image specific versus universal textons for the middle TG scale along with the combined TG scales. The optimal number of universal textons is double the number for image specific textons. In addition, smaller scales of TG require fewer textons. The scaling is roughly linear in the area of the TG disc, so that one scales the number of textons to keep the number samples/bin constant. Results are insensitive to within a factor of two of the optimal number. From (e), we see that the choice between image-specific and universal textons is not critical. In our experiments, we use image-specific textons with k={12,24,48}. The choice for us is unimportant, though for other applications such as object recognition one would likely prefer the measure of texture provided by universal textons, which can be compared across images.

textons, we recompute the adaptive texton bins for each test image separately. For universal textons, we compute a standard set of textons from the 200 training images. The computational cost of each approach is approximately equal since the per-image k-means problems are small and one can use fewer textons in the image-specific case.

Fig. 8 shows experiments covering both texton questions. One can see that the choice between image specific and universal textons is not important for performance. We use image-specific textons for convenience, though universal textons are perhaps more appealing in that they can be used to characterize textures in an image-independent manner. Image-independent descriptions of texture would be useful for image retrieval and object recognition applications. The figure also reveals two scaling rules for the optimal number of textons. First, the optimal number of textons for universal textons is roughly double that required for image specific textons. Second, the optimal number of textons scales linearly with the area of the disc. The former scaling is expected, to avoid overfitting in the image-specific case. The latter scaling rule keeps the number of samples per texton bin constant, which reduces overfitting for the smaller TG scales.

It may be surprising that one gets comparable results using both image-specific and universal textons as the image-specific textons vary between training and testing images. Since the texture gradient is only dependent on having good estimates of the distribution in each half-disc, the identity of individual textons is unimportant. The adaptive binning given by k-means on a per-image basis appears to robustly estimate the distribution of filter response and is well behaved across a wide variety of natural images.

## 5 Cue Combination

After optimizing the performance of each cue, we face the problem of combining the cues into a single detector. We approach the task of cue combination as a supervised learning problem, where we will learn the combination rules from the ground truth data. There is some previous work on learning boundary models. Will et. al. [7] learn texture edge models for synthetic Brodatz mosaics. Meila and Shi [32] present a framework for learning segmentations from labeled examples. Most compelling is the work of Konishi et. al. [12], where edge detectors were trained on human-labeled images.

Fig. 9 presents the first set of cue combination experiments using logistic regression. The first task is to determine whether any of the cues is redundant given the others. Until this point, we have presented four cues, two of which—OE and BG—detect discontinuities in brightness. Fig. 9a shows that BG is a superior cue to OE, whether used in conjunction with the texture gradient alone or with the texture and color gradients together. In addition, since we do not gain anything by using OE and BG in conjunction (not shown), we can safely drop OE from the list of cues.

We have the option of computing each cue at multiple scales. Fig. 5 shows that only the texture gradient contains significant independent information at the different scales. The benefit of using multiple TG scales does not remain when

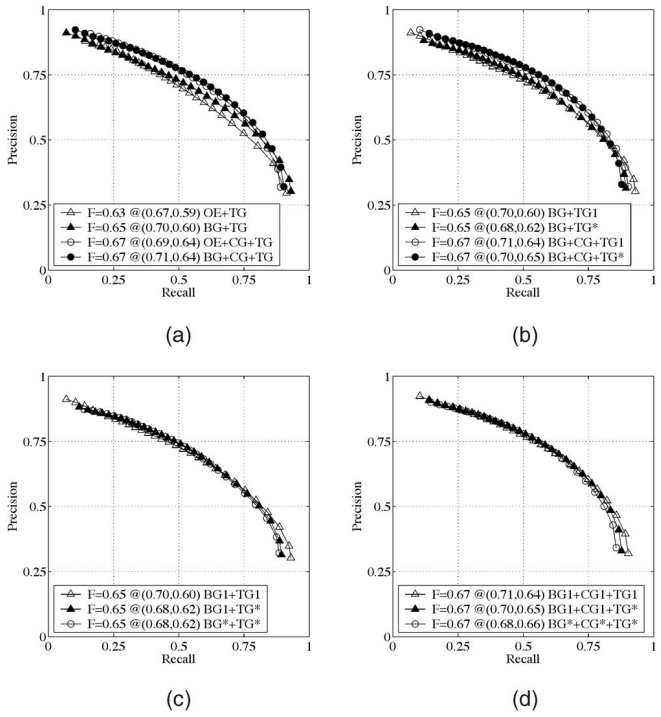


Fig. 9. Cue Combination. After optimizing the parameters of each cue independently, we seek to combine the cues effectively. (a) OE versus BG: shows that whether or not we include CG, we are always better off using BG as our brightness cue instead of OE. Note that though the curve is not shown, using OE and BG together is not beneficial. (b) Multiscale TG: Although we saw in Fig. 5 that we benefit from using multiple scales of TG, the benefit is significantly reduced when BG is included. This is because BG contains some ability to discriminate fine scale textures. (c) Gray-scale model: Our noncolor model of choice is simply the combination of a single scale of BG with a single scale of TG. (d) Color model: Our color model of choice also includes only a single scale of each of the BG, CG, and TG features.

TG is combined with other cues. Fig. 9b shows the effect of using multiple TG scales in conjunction with BG and CG. In both the BG and BG + CG cases, multiple TG scales improve performance only marginally. The Figs. 9c and 9d show the effect of adding multiple BG and CG scales to the model. In neither case do multiple scales improve overall performance. In some cases (see Fig. 9d), performance can degrade as additional scales may introduce more noise than signal.

In order to keep the final system as simple as possible, we will retain only the middle scale of each feature. However, it is surprising that multiscale cues are not beneficial. Part of the reason may be that the segmentation data set itself contains a limited range of scale, as subjects were unlikely to produce segmentations with more than approximately 30 segments. An additional explanation is suggested by Figs. 5h and 9b, where we see that the multiple scales of TG have independent information, but the benefit of multiple TG scales vanishes when BG is used. The brightness gradient operates at small scales, and is capable of low-order texture discrimination. At the smallest scales, there is not enough information for high-order texture analysis anyway, so BG is a good small-scale texture feature. The texture gradient identifies the more complex, larger scale textures.

Until this point, all results were generated with a logistic model. We will show that the logistic model is a good choice by comparing a wide array of classifiers, each trained on the human segmentation data set. With more powerful models, we hoped to discover some interesting cross-cue and cross-scale gating effects. For example, one might discount the simpler boundary detection of BG when TG is low because the brightness edges are likely to correspond to edges interior to textured areas. In addition, the optimal mixing function for the various cues could well be nonlinear, with each cue treated as an expert for a certain class of boundaries. These are the classifiers that we used:

**Density Estimation**. We do density estimation with adaptive bins provided by vector quantization using k-means. Each k-means centroid provides the density estimate of its Voronoi cell as the fraction of on-boundary samples in the cell. We use k=128 bins and average the estimates from 10 runs to reduce variance.

**Classification Trees**. The domain is partitioned hierarchically with top-down axis-parallel splits. When a cell is split, it is split in half along a single dimension. Cells are split greedily so as to maximize the information gained at each step. The effect of this heuristic is to split nodes so that the two classes become separated as much as possible. A 5 percent bound on the error of the density estimate is enforced by splitting cells only when both classes have at least 400 points present.

**Logistic Regression**. This is the simplest of our classifiers and the one perhaps most easily implemented by neurons in the visual cortex. Initialization is random, and convergence is fast and reliable by maximizing the likelihood with about five Newton-Raphson iterations. We also consider two variants: quadratic combinations of features, and boosting using the confidence-rated generalization of AdaBoost by Schapire and Singer [33]. No more than 10 rounds of boosting are required for this problem.

**Hierarchical Mixtures of Experts**. The HME model of Jordan and Jacobs [34] is a mixture model where both the experts at the leaves and the internal nodes that compose the gating network are logistic functions. We consider small binary trees up to a depth of three (eight experts). The model is initialized in a greedy, top-down manner and fit with EM. Two hundred iterations were required for the log likelihood to converge.

**Support Vector Machines**. We use the SVM package `libsvm` [35] to do soft-margin classification using Gaussian kernels. The optimal parameters were $\nu = 0.2$ and $\sigma = 0.2$. In this parameterization of SVMs, $\nu$ provides the expected fraction of support vectors, which is also an estimate of the degree of class overlap in the data. The high degree of class overlap in our problem also explains the need for a relatively large kernel.

We used 200 images for training and algorithm development. The 100 test images were used only to generate the final results for this paper. The authors of [11] show that the segmentations of a single image by the different subjects are highly consistent, so we consider all human-marked

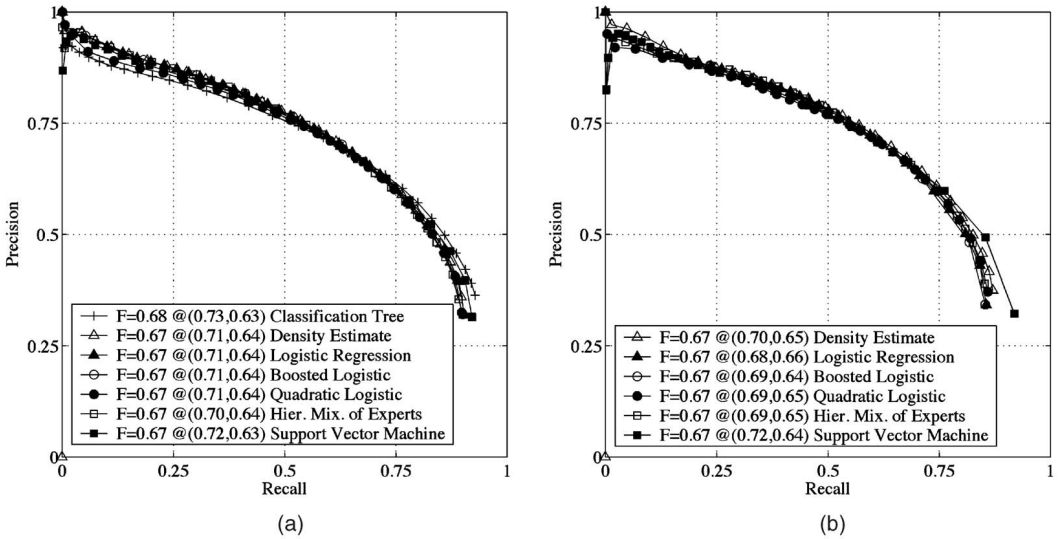


Fig. 10. Choice of Classifier. Until this point, all results have been shown using the logistic regression model. (a) One scale per cue and (b) three scales per cue. This model is appealing because it is compact, robust, stable, interpretable, and quick to both train and evaluate. However, its linear decision boundary precludes any potentially interesting cross-cue gating effects. In this figure, we show the result of applying various more powerful models on (a) one scale of each of BG, CG, and TG, and (b) all three scales of each feature (nine total features). The classification tree model could not be applied in (b) due to the increased number of features. In neither case does the choice of classifier make much difference. In both cases, the logistic regression performs well. The addition of multiple scales does not improve performance. The logistic is still the model of choice.

boundaries valid. For training, we declare an image location $(x, y, \theta)$ to be on-boundary if it is within $\Delta x \leq \sqrt{8}$ pixels and $\Delta\theta = 30$ degrees of any human-marked boundary. The remainder are labeled off-boundary.

This classification task is characterized by relatively low dimension, a large amount of data (100M samples for our 240 x 160-pixel images), poor class separability, and a 10:1 class ratio. The maximum feasible amount of data, uniformly sampled, is given to each classifier. This varies from 50M samples for the density estimation and logistic regression to 20K samples for the SVM and HME. Note that a high degree of class overlap in *any* low-level feature space is inevitable because the human subjects make use of both global constraints and high-level information to resolve locally ambiguous boundaries.

The CPU time required for training and evaluating the models varied by several orders of magnitude. For training, the logistic regression and classification trees required several minutes on a 1GHz Pentium IV, while the density estimation, HME, and SVM models—even with significantly reduced data—required many hours. For evaluation, the logistic regression and classification trees were again the fastest, respectively, taking constant time and time logarithmic in the number of data points. For these, the evaluation time was dominated by the couple of minutes required to compute the image features. The density estimation model evaluation is linear in the value of $k$ used for $k$-means and the number of runs, adding a constant factor of 1,280 to an operation requiring $2f$ operations per pixel, where $f$ is the number of features. The HME is a constant factor of at most 15 slower than the logistic, due to our limit of eight experts. The SVM model is prohibitively slow. Since 25 percent of the training data become support vectors, the SVM required hours to evaluate for a single image.

Fig. 10a shows the performance of the seven classifiers using only the middle scale of BG, CG, and TG. The PR curves all cross approximately at the maximal F-measure point and, so, all the classifiers are equivalent as measured by the F-measure. The classification tree and SVM are able to achieve marginally higher performance in the high recall and low precision regime, but they perform worse in the low recall and high precision area. Overall, the performance of all the classifiers is approximately equal, but other issues affect model choice such as representational compactness, stability, bias, variance, cost of training, and cost of evaluation.

The nonparametric models achieve the highest performance, as they are able to make use of the large amount of training data to provide unbiased estimates of the posterior, at the cost of opacity and a large model representation. The plain logistic is stable and quick to train, and produces a compact and intuitive model. In addition, the figure shows that the logistic's bias does not hurt performance. When given sufficient training data and time, all the variants on the logistic—the quadratic logistic, boosted logistic, and HME—provided minor performance gains. However, the many EM iterations required to fit the HME required us to subsample the training data heavily in order to keep training time within reasonable limits.

The support vector machine was a disappointment. Training time is superlinear in the number of samples, so the training data had to be heavily subsampled. The large class overlap produced models with 25 percent of the training samples as support vectors, so that the resulting model was opaque, large, and exceedingly slow to evaluate. In addition, we found the SVM to be brittle with respect to its parameters $\nu$ and $\sigma$. Even at the optimal settings, the training would occasionally produce nonsensical models. Minute variations from the optimal settings would produce infeasible problems. We conclude that the

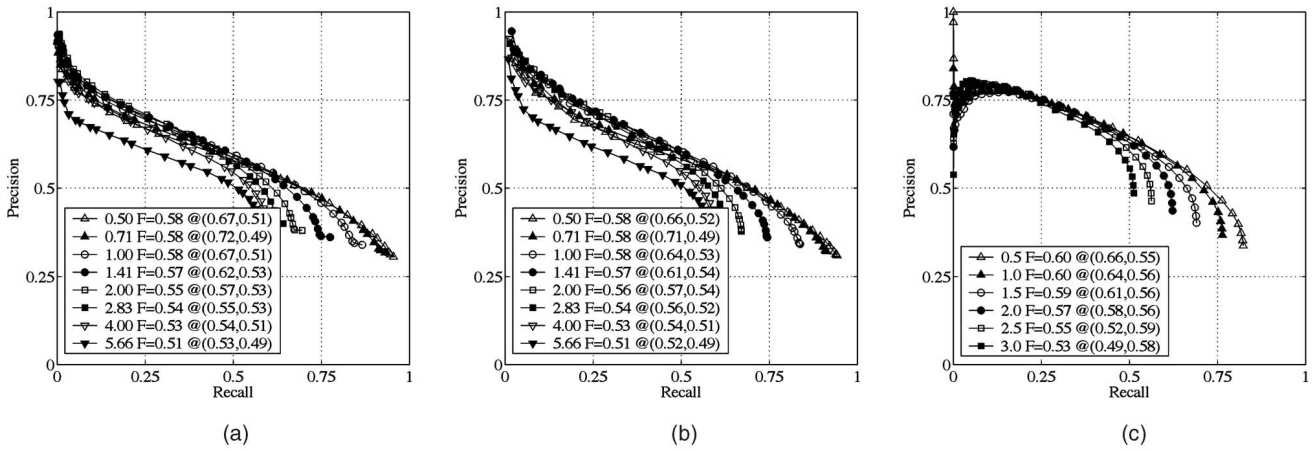


Fig. 11. (a) Gaussian derivative. (b) GD + hysteresis. (c) Second moment matrix. Choosing $\sigma$ for the classical edge operators. The Gaussian derivative (GD) operator (a) without and (b) with hysteresis, and (c) the second moment matrix (2MM) operator, fitted as in Fig. 12. From these experiments, we choose the optimal scales of $\sigma = 1$ for GD regardless of hysteresis, and $\sigma = 0.5$ for 2MM.

SVM is poorly suited to a problem that does not have separable training data.

Fig. 10b shows the performance of each classifier except the classification tree when all three scales are included for each of the three features. The results are much as before, with virtually no difference between the different models. Balancing considerations of performance, model complexity, and computational cost, we favor the logistic model and its variants.

## 6 Results

Having settled on a grayscale boundary model using a single scale each of BG and TG, and a color model that adds a single scale of CG, we seek to compare these models to classical models and the state of the art. The model that we present as a baseline is MATLAB's implementation of the Canny [5] edge detector. We consider the detector both with and without hysteresis. To our knowledge, there is no work proving the benefit of hysteresis thresholding for natural images. We will call the Canny detector without hysteresis "GD," as it is simply a Gaussian derivative filter with nonmaxima suppression. With hysteresis, the operator is called "GD + H." The GD and GD + H detectors each have a single parameter to tune—the $\sigma$ of the Gaussian derivative filters. Figs. 11a and 11b show the PR curves for various choices of $\sigma$. For both cases, $\sigma = 1$ pixel is a good choice. Note that the detector threshold is not a parameter that we need to fit, since it is the parameter of the PR curves.

We also consider a detector derived from the spatially-averaged second moment matrix (2MM). It has long been known that the eigenspectrum of the second moment matrix provides an informative local image descriptor. For example, both eigenvalues being large may indicate a corner or junction. This is the basis of the Plessey or Harris-Stephens [36] corner detector and the Förstner corner detector [37]. One large and one small eigenvalue may indicate a simple boundary. The Nitzberg edge detector [38] used by Konishi et al. [12] is based on the difference between the eigenvalues.

We apply the same training/test methodology to the 2MM detector as we do to our own detectors, using the full eigenspectrum as a feature vector. From the 200 training images, we obtain on and off-boundary labels for pixels and train a logistic model using both eigenvalues of the 2MM as features. Fig. 12 shows the model trained in this manner. Fig. 12a shows the distribution of the training data in feature space. Fig. 12b shows the empirical posterior, and Fig. 12c shows the fitted posterior from the logistic model. To perform nonmaxima suppression on the 2MM output, we calculated the orientation of the operator's response from the leading eigenvector.

The 2MM detector also has two scale parameters. The *inner scale* is the scale at which image derivatives are estimated. We set the inner scale to a minimum value, estimating the derivatives with the typical 3 x 3 [-1,0,1] filters. Fig. 11c shows the optimization over the *outer scale* parameter, which is the scale at which the derivatives are spatially averaged. Only a modest amount of blur is required ($\sigma = 0.5$ pixels). Note that some blur is required, or the second eigenvalue vanishes. Less smoothing is not possible due to pixel resolution.

In Fig. 13, we give a summary comparison of the BG, CG, and TG detectors, along with two combinations: BG + TG for grayscale images, and BG + CG + TG for color images. It is clear that each feature contains a significant amount of independent information. Fig. 3 shows the comparison between the two Gaussian derivative operators (GD and GD + H), the second moment matrix operator (2MM), our grayscale BG + TG operator, and our color BG + CG + TG operator.[4] First, note that hysteresis does impart a marginal improvement to the plain GD operator, though the difference is pronounced only at very low recall rates. The 2MM operator does mark a significant improvement over the Canny detector, except at low recall. The main benefit of the 2MM operator is that it does not fire where both eigenvalues are large—note the opposite signs of the coefficients in the model. As a result, it does not fire where energy at multiple orientations coincide at a pixel, such as at

4. The logistic coefficients for the BG + TG operator are 0.50 for BG and 0.52 for TG with an offset of -2.81. The coefficients for the color model are 0.31 for BG, 0.53 for CG, and 0.44 for TG, with an offset of -3.08. The features are normalized to have unit variance. Feature standard deviations are 0.13 for BG, 0.077 for CG, and 0.063 for TG.

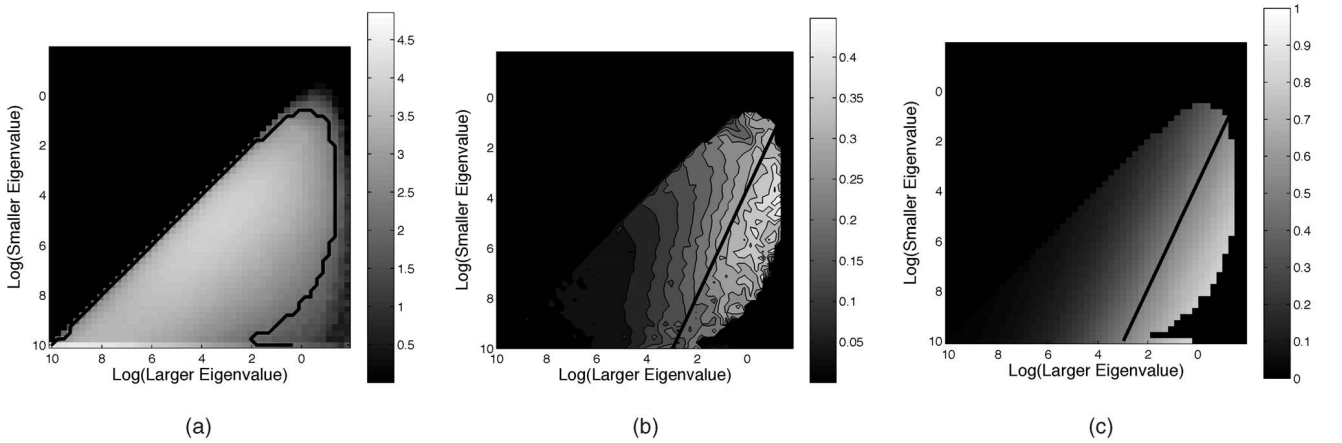


Fig. 12. Optimizing the second moment matrix model. For this model, the two features are the smaller and larger eigenvalues of the locally averaged 2nd moment matrix. (a) Log10 (sample count): shows the histogram of samples from the 200 training images along with the 100 samples/bin contour. (b) Empirical posterior: shows the empirical posterior probability of a boundary. (c) Fitted posterior: shows the fitted posterior using logistic regression. We did not find more complex models of the posterior to be superior. The linear decision boundary of the fitted logistic is drawn in both (b) and (c). The coefficients of the fitted logistic are -0.27 for the larger eigenvalue and 0.58 for the smaller eigenvalue, with an offset of -1.

corners or inside certain textures. Thus, 2MM reduces the number of false positives from high contrast texture.

The operators based on BG and TG significantly outperform both classical and state of the art boundary detectors. The main reason for the improved performance is a robust treatment of texture. Neither GM nor 2MM can detect texture boundaries. For the same reason that 2MM suppresses false positives inside textured areas, it also suppresses edges between textured areas.

Fig. 3 also shows the performance of the human subjects in the segmentation data set. Each plotted point shows the precision and recall of a single human segmentation when it is compared to the other humans' segmentations of the same image. The median human F-measure is 0.80. The solid line in the upper right corner of the figure shows the iso-F-measure line for 0.80, representing the F-measure frontier of human performance.

Each of the curves in Fig. 3 uses a fixed distance tolerance $d_{\max} = 1$ percent of the image diagonal (2.88 pixels). Fig. 14 shows how each detector's F-measure varies as this tolerance changes. The digital pixel grid forces a discretization of this parameter, and the figure shows the result for $d_{\max} = \{\sqrt{2}, 2, \sqrt{5}, \sqrt{10}\}$. Figs. 14a, 14b, 14c, and 14d show the PR curves for each detector. Since these curves do not intersect and are roughly parallel, the F-measure captures the differences effectively. Fig. 14e shows how the F-measure changes as a function of $d_{\max}$ for each detector and for the human subjects. If a detector's localization were good to within 1 pixel, then the detector's curve would be flat. In contrast, all of the machine curves reveal localization error greater than that shown by the human subjects. Additional work on local boundary detection will no doubt narrow the gap between machine and human performance, but large gains will ultimately require higher-level algorithms. Preliminary work [39] suggests that human subjects viewing local patches such as those in Fig. 2 perform at a level equivalent to our best detector.

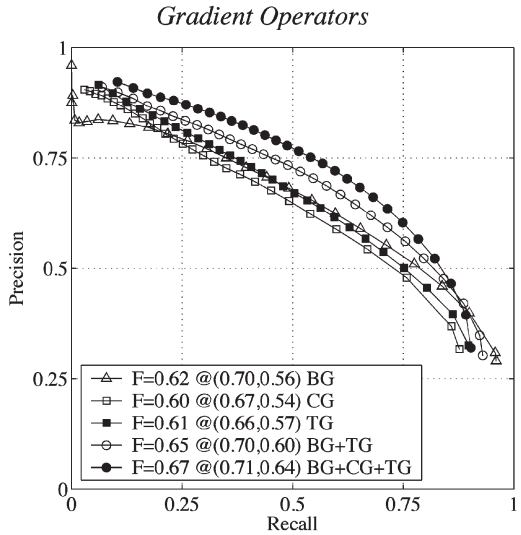


Fig. 13. Detector Comparison. The performance of the boundary detectors proposed in this paper, both independently and in combination.

We present qualitative results in Figs. 15, 16, and 17. The first figure shows various versions of our detectors along with the humans' boundaries. The second figure shows a comparison between the GD + H, 2MM, and BG + TG detectors alongside the humans' boundaries. The third figure shows close-up views of several interesting boundaries. Each machine detector image in these figure shows the soft boundary map after non-maxima suppression, and after taking the maximum over $\theta$. In Fig. 15, we see the complementary information contained in the three channels, and the effective combination by the logistic model. For example, color is used when present in (b, c, i) to improve the detector output. Fig. 16 shows how the BG + TG detector has eliminated false positives from texture while retaining good localization of boundaries. This effect is particularly prominent in image Fig. 16e.

The man's shoulder from Fig. 16e is shown in more detail in Fig. 17a. This image illustrates several interesting issues. The striped shirt sleeve is a difficult texture boundary due to the large scale of the stripes compared to the width of the region. Nevertheless, the boundary is successfully detected

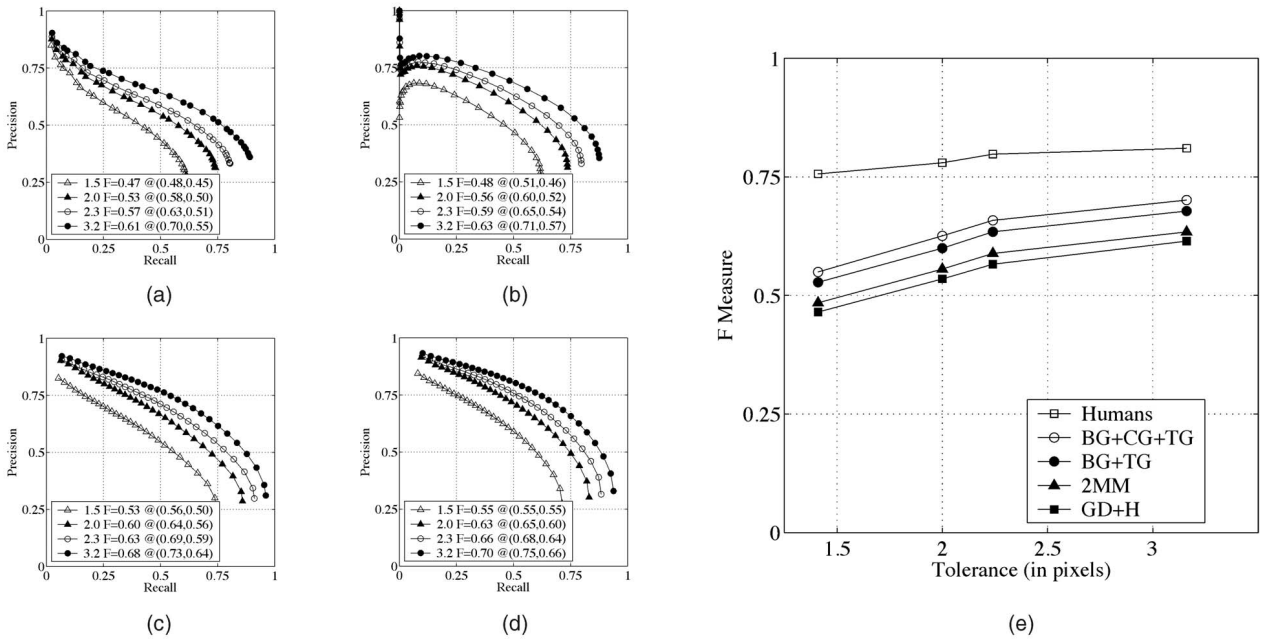


Fig. 14. Detector comparison at various distance tolerances. (a) GD + hysteresis, (b) second moment matrix, (c) BG + TG, and (d) BG + CG + TG show the precision recall curves for each detector as the matching tolerance varies from $\sqrt{2}$ to $\sqrt{10}$ pixels. The curves for each detector do not intersect, and so the F-measure is a good representation of the performance regardless of threshold. (e) shows the relationship between F-measure and the distance tolerance for the four detectors, along with the median human performance. The human curve is flatter than the machine curves, showing that the humans' localization is good. The gap between human and machine performance can be reduced but not closed by better local boundary models. Both mid-level cues and high-level object-specific knowledge are likely required to approach the performance of the human subjects.

by TG with good localization and without the false positives marked by brightness-based approaches such as GM. The 2MM detector also has grave difficulty with this texture because is it not isotropic, so that the eigengap remains large inside the texture. Note that no detector found the top edge of the man's shoulder. There is no photometric evidence for this boundary, yet it was marked by the human subjects with surprising agreement. It is clear that we cannot hope to find such boundaries without object-level information.

Figs. 17e and 17g in show the reduction in false positives in our detectors compared to the GM and 2MM detectors. Fig. 17c shows another difficult texture boundary along the underside of the boat where the texture is anisotropic, and its direction is oblique to the object boundary.

Figs. 17b, 17d, and 17f show how different feature channels in our detector can cooperate to find composite boundaries. Especially in Fig. 17b, we can see that all three channels (BG, CG, and TG) have found the boundary of the ear. The BG has good localization because of its smaller scale, but also has more false positives inside the ear. The CG has a powerful response from the skin tone, but its larger support sacrifices localization somewhat around the ear-lobe. The texture gradient has strong responses around the ear, between the ear and face, and around the eye, but localization is everywhere a few pixels off. By combining the three responses, important boundaries found in any one channel survive while boundaries found in multiple channels are reinforced. This reinforcement not only strengthens the response, but also benefits localization where, for example, the BG response around the ear pulls the TG response into better alignment. The final result is strikingly similar to the human marked boundaries.

## 7 Conclusion

We have defined a novel set of brightness, color, and texture cues appropriate for constructing a local boundary model, as well as a methodology for benchmarking boundary detection algorithms. By using a large data set of human-labeled boundaries in natural images, we have formulated the task of cue combination for local boundary detection as a supervised learning problem. This approach models the true posterior probability of a boundary at every image location and orientation, which is particularly useful for higher-level algorithms. The choice of classifier for modeling the posterior probability of a boundary based on local cues is not important—a simple linear model is sufficiently powerful. Based on a quantitative evaluation on 100 natural images, our detector outperforms existing methods, indicating that a proper treatment of texture is essential for detecting boundaries in natural images.

## Appendix A
## Computing Gradient Features

The most computationally expensive part of the gradient computations is the computation of the half-disc feature histograms. At each pixel, we must compute two histograms over semicircular neighborhoods at several orientations and several scales. Properly structured, this computation can be done efficiently.

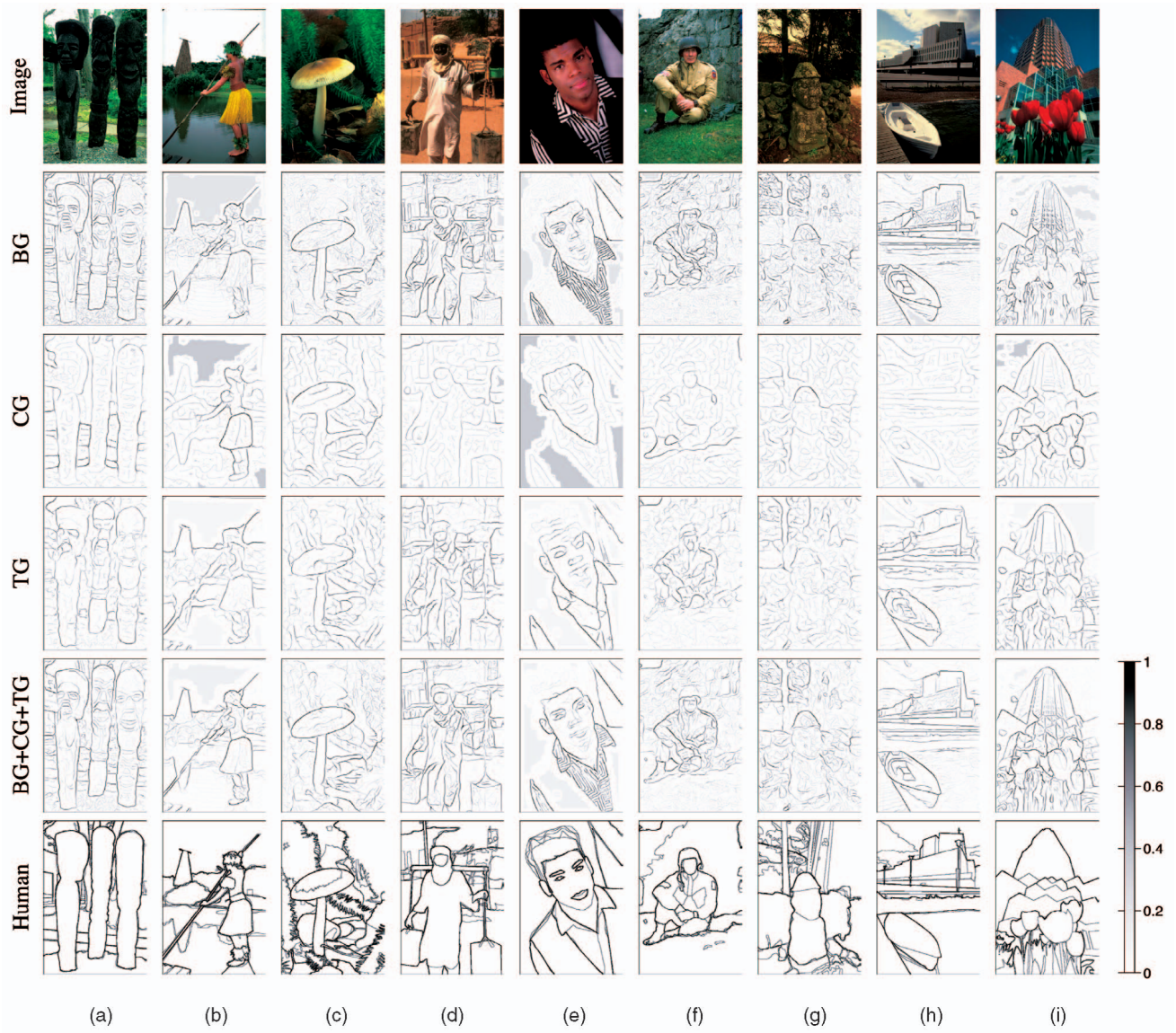


Fig. 15. Boundary images for the gradient detectors presented in this paper. Rows 2, 3, and 4 show real-valued probability-of-boundary ($P_b$) images after nonmaxima suppression for the three cues. The complementary information in each of the three BG, CG, and TG channels is successfully integrated by the logistic function in row 5. The boundaries in the human segmentations shown in row 6 are darker where more subjects marked a boundary.

The most significant speedup is achieved by optimizing the loop over orientations. Assuming that we wish to compute the gradient at $n$ evenly spaced orientations, we can divide the disc into $2n$ pie slices. If we compute the pixel histogram for each pie slice, then any half-disc histogram is simply the sum of $n$ adjacent pie slice histograms. In addition, we can compute the histograms for orientation $i+1$ incrementally from the histograms from orientation $i$ by subtracting the last slice and adding the next slice as we spin the disc. Note also that the initial step of computing the pie slice histograms can be optimized by precomputing a slice membership mask.

For the texture gradient, these optimizations are sufficient. However, the soft binning required by BG and CG suggest other opportunities for speedup. Each pixel contributes one point to the histogram for each kernel sample. Simply precomputing kernel offsets and values is effective, though this approach is slow if the number of kernel samples is large. If there are more kernel samples than bins, then one should precompute the total histogram contribution from each pixel.

Other loops may admit additional optimization opportunities. In the same way that we split the disc by orientation into pie slices, one could additionally split the disc into concentric rings corresponding to the multiple scales. Since our half-octave scales produce an area increment for the disc of 2x per scale, our computation is dominated by the larger scale. A smaller scale increment might motivate this optimization.

There is still much redundant computation as we sweep the disc across a scan-line. The pie slice histograms change slowly between adjacent pixels, especially when the number of orientations is not large. It is possible to compute them incrementally by computing slice update masks. For

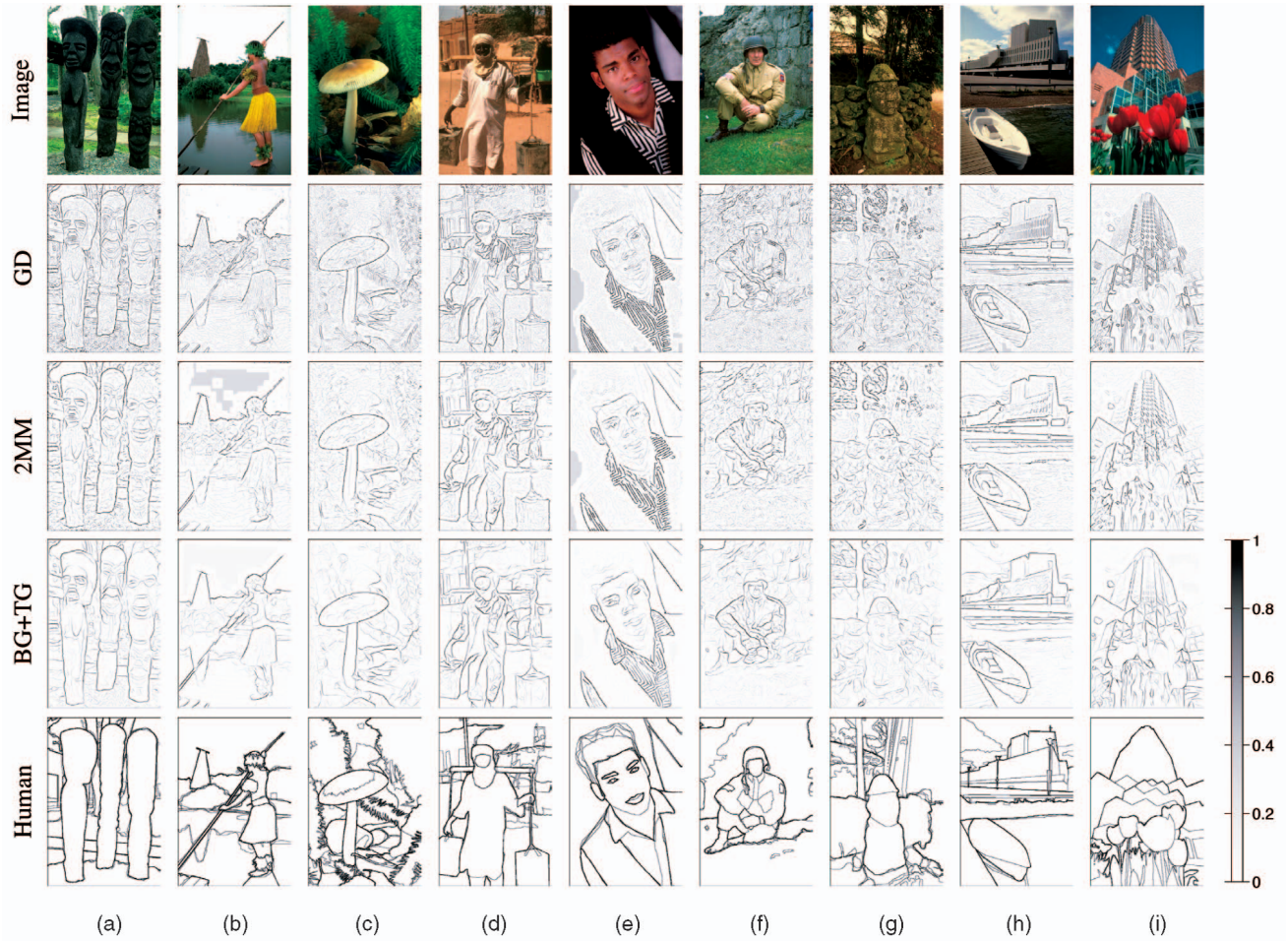


Fig. 16. Boundary images for three grayscale detectors. Compare with Fig. 15. Rows 2, 3, and 4 show $P_b$ images for the Gaussian derivative (GD), the second moment matrix (2MM), and our brightness+texture detector (BG + TG). The human segmentations are shown once more for comparison. The BG + TG detector benefits from 1) operating at a large scale without sacrificing localization and 2) the suppression of edges in the interior of textured regions.

large radii, this optimization achieves an order of magnitude speedup.

## Appendix B

## Corresponding Boundary Maps

In this section, we present the algorithm used for computing the correspondence between a thresholded machine boundary map and a human labeled boundary map. We convert the correspondence problem into a minimum cost bipartite assignment problem, where the weight between a machine boundary pixel and a human boundary pixel is proportional to their relative distance in the image plane. One can then declare all boundary pixels matched beyond some threshold $d_{\max}$ to be nonhits.

The best dense assignment algorithms [40], [41] have typical runtime complexity somewhere between $O(n^2)$ and $O(n^3)$. This is too slow for our purposes and, so, we must formulate a sparse assignment problem. We use Goldberg's CSA package, which implements the best known algorithms for min-cost sparse assignment [42], [43]. The CSA code appears to run in time linear in the size of the graph, $O(n)$.

What remains is to construct a sparse version of our assignment problem that conforms to certain practical algorithmic concerns. In order to make the problem sparse, we include in the graph only those edges with weight $w \leq d_{\max}$, since an edge with $w > d_{\max}$ could only serve to vacuously assign a missed human boundary to a machine's false positive. After this sparsification step, any isolated node can be removed from the assignment problem and immediately counted as a miss or false positive.

The min-cost assignment problem requires one to specify the degree of the assignment to restrict the search to nontrivial solutions. Since we cannot know the degree a priori, we must request a perfect matching, i.e., a matching that involves all nodes. However, the sparsification step will almost certainly have removed edges required for a perfect matching. This problem is easily solved by adding outlier nodes on both sides of the match. All edges incident on an outlier node have higher weight than any real edge in the graph, ensuring that they are used only when necessary to extend the true min-cost partial matching to a valid perfect matching.

Given a sparse assignment problem with $n_L$ nodes on the left side and $n_R$ nodes on the right, we add $n_R$ outlier nodes to the left and $n_L$ outlier nodes to the right. This squared problem has enough nodes to ensure a perfect matching, but we cannot afford dense outlier connections. We can, however, exploit the fact that all the outlier connections

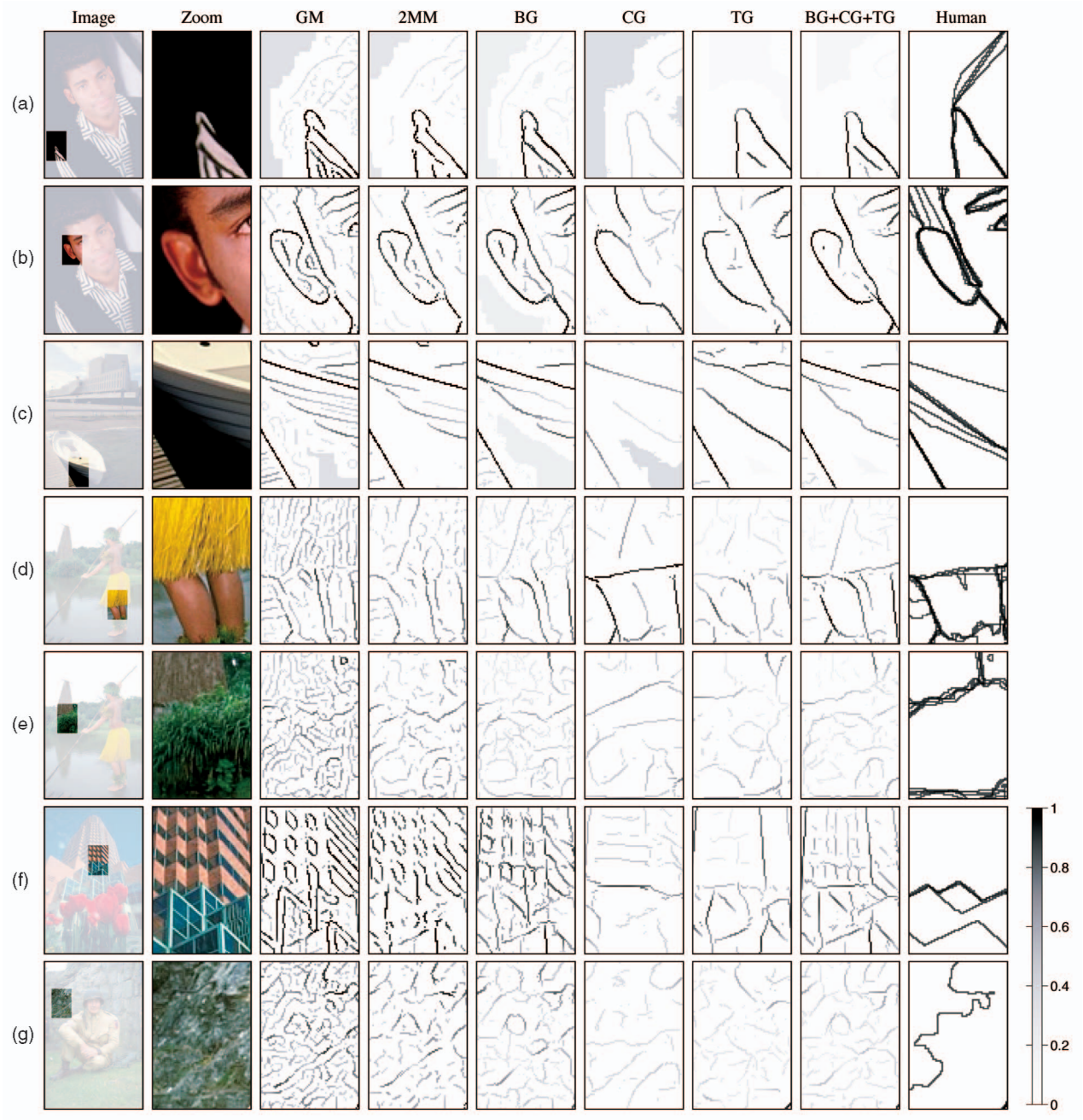


Fig. 17. Close-up boundary and nonboundary examples. These examples are taken from the images shown in Figs. 15 and 16. They have been chosen to illustrate the strengths of the different features, as well as the shortcomings of these various local detectors. Briefly, they show (a) a difficult texture boundary and an illusory contour, (b) useful CG signal and appropriate scale for TG, (c) a difficult texture boundary (bottom of boat) found by TG, (d) an example where BC, CG, and TG cooperate effectively, (e) and (f) more difficult texture boundaries (from images (b) and (i) in Figs. 15 and 16) arguably localized by our detectors but completely lost in the GM and 2MM responses, and (g) the interior of a textured region (from the wall in image (g)) showing the reduced false positive responses of our detectors inside a natural texture.

have identical weight. Given an assignment solution, used outlier edges are interchangeable and unused outlier connections could not have affected the solution. Consequently, dense outlier connections contain enormous redundancy and are overly conservative. By appealing to the high degree of connectivity present in random graphs, we can keep the size of our graph linear in the number of nodes by including a constant number of outlier connections per node. We found $d = 6$ connectivity to be sufficient, so that there are $d$ random outlier connections to each real node and $d$ random outlier connections to each outlier node.

One small detail remains, as the graph still does not guarantee the existence of a perfect matching. As a safety net, we overlay a perfect matching of high cost that matches each real node to an outlier node in a parallel fashion. We add these connections before the random outlier connections and add the outlier connections randomly without replacement. The minimum cost perfect matching in this graph provides the best correspondence of pixels between the machine and human boundary maps, with a maximum localization tolerance of $d_{\max}$. Fig. 18 depicts the graph construction procedure.

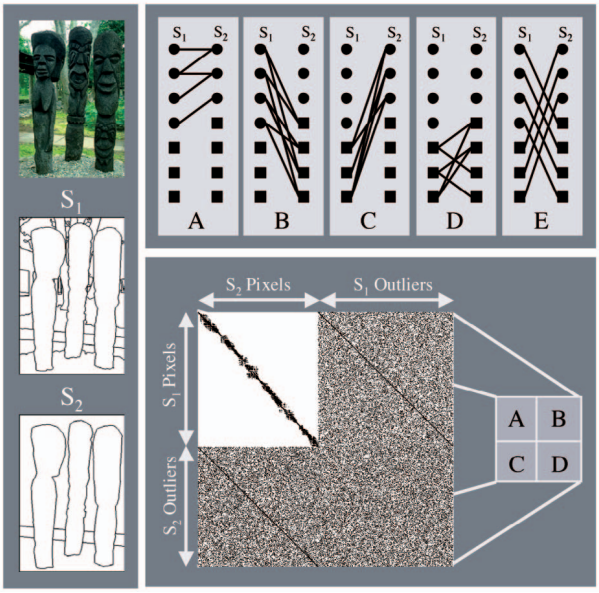


Fig. 18. Bipartite Graph for Comparing Boundary Maps. We compare two boundary maps by corresponding boundary pixels. The figure shows the construction of the bipartite graph used in computing the correspondence. The top panel contains an illustration of the five types of edges in the graph. The bottom panel contains the adjacency matrix for the graph. The two boundary maps $S_1$ and $S_2$ contribute $n_1 = 3,664$ and $n_2 = 4,124$ nodes to the graph after pruning isolated pixels. After adding outlier nodes to both sides, we have a square $n \times n$ assignment problem, where $n = n_1 + n_2 = 7,788$. The adjacency matrix for the bipartite graph has a block structure. Each block contains the corresponding edges from the top panel. The top-left block A contains the sparse local connections between pixels—the only "real" edges in the graph. Blocks B and C contain random outlier connections, and block D contains random outlier-to-outlier connections. The E edges lie on the diagonals of the B and C blocks, providing the safety-net high-cost perfect matching. The entire matrix has 64,470 nonzeros, for a density of 0.1 percent.

The main shortcoming of the algorithm as presented is in the area of junctions, where assignments can be made between boundary pixels that occur on boundaries at different orientations. One can easily incorporate an orientation penalty into the bipartite graph's edge weights, but we have verified that this enhancement has no perceptible effect on the aggregate precision and recall values because of the scarcity of junctions relative to simple edges. One could also count hits, misses, and false positives in a soft manner by using the values of the edge weights in the match. However, the simple binary counting is sufficient given the large number of images we used, not to mention the lack of a convincing cost function.

## Acknowledgments

The authors would like to thank the Berkeley Computer Vision Group, in particular Xiaofeng Ren. They also thank Mike Jordan and Peter Bartlett for advice and discussion regarding classifiers, Kobus Barnard for discussions on color, and Andrew Goldberg for the CSA code and advice on using it. Finally, they thank the anonymous reviewers whose thoughtful comments and suggestions led to a much improved manuscript. Jitendra Malik is supported by the Miller Research Professorship at UC Berkeley. This work was supported by the US Office of Naval Research grant N00014-01-1-0890 (MURI).

## References

[1] Z. Tu, S. Zhu, and H. Shum, "Image Segmentation by Data Driven Markov Chain Monte Carlo," *Proc. Int'l Conf. Computer Vision*, vol. 2, pp. 131-138 July 2001.
[2] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *Int'l J. Computer Vision*, vol. 43, no. 1, pp. 7-27, June 2001.
[3] L. Williams and D. Jacobs, "Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience," *Proc. Int'l Conf. Computer Vision*, 1995.
[4] X. Ren and J. Malik, "A Probabilistic Multi-Scale Model for Contour Completion Based on Image Statistics," *Proc. Seventh European Conf. Computer Vision*, 2002.
[5] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.
[6] Y. Rubner and C. Tomasi, "Coalescing Texture Descriptors," *Proc. ARPA Image Understanding Workshop*, 1996.
[7] S. Will, L. Hermes, J.M. Buhmann, and J. Puzicha, "On Learning Texture Edge Detectors," *Proc. Int'l Conf. Image Processing*, pp. 877-880, 2000.
[8] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026-1038, Aug. 2002.
[9] L. Hermes, T. Zoller, and J. Buhmann, "Parametric Distributional Clustering for Image Segmentation," *Proc. European Conf. Computer Vision*, 2002.
[10] J. Rivest and P. Cavanagh, "Localizing Contours Defined by More Than One Attribute," *Vision Research*, vol. 36, no. 1, pp. 53-66, 1996.
[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," *Proc. Int'l Conf. Computer Vision*, 2001.
[12] S. Konishi, A.L. Yuille, J. Coughlan, and S.C. Zhu, "Fundamental Bounds on Edge Detection: An Information Theoretic Evaluation of Different Edge Cues," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 573-579, 1999.
[13] M. Morrone and D. Burr, "Feature Detection in Human Vision: A Phase Dependent Energy Model," *Proc. Royal Soc. of London B*, vol. 235, pp. 221-245, 1988.
[14] P. Perona and J. Malik, "Detecting and Localizing Edges Composed of Steps, Peaks and Roofs," *Proc. Int'l Conf. Computer Vision*, 1990.
[15] W. Niblack et al., "The QBIC Project: Querying Image by Content Using Color, Texture, and Shape," *Proc. SPIE*, vol. 1908, 1993.
[16] M. Ruzon and C. Tomasi, "Color Edge Detection with the Compass Operator," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1999.
[17] M. Ruzon and C. Tomasi, "Corner Detection in Textured Color Images," *Proc. Int'l Conf. Computer Vision*, pp. 1039-1045, 1999.
[18] I. Fogel and D. Sagi, "Gabor Filters as Texture Discriminator," *Biological Cybernetics*, vol. 61, pp. 103-113, 1989.
[19] J. Malik and P. Perona, "Preattentive Texture Discrimination with Early Vision Mechanisms," *J. Optical Soc. Am.*, vol. 7, no. 2, pp. 923-932, May 1990.
[20] D. Heeger and J. Bergen, "Pyramid-Based Texture Analysis/Synthesis," *Proc. SIGGRAPH*, 1995.
[21] J. Puzicha, T. Hofmann, and J. Buhmann, "Non-Parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 267-272, 1997.
[22] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann, "Empirical Evaluation of Dissimilarity Measures for Color and Texture," *Proc. Int'l Conf. Computer Vision*, 1999.
[23] C. Mallows, "A Note on Asymptotic Joint Normality," *Annals of Math. Statistics*, vol. 43, no. 2, pp. 508-515, 1972.
[24] E. Levina and P. Bickel, "The Earth Mover's Distance is the Mallows Distance: Some Insights From Statistics," *Proc. Int'l Conf. Computer Vision*, vol. 2, pp. 251-256, 2001.
[25] S. Palmer, *Vision Science*. MIT Press, 1999.

[26] E. Levina, "Statistical Issues in Texture Analysis," PhD thesis, Univ. of California, Berkeley, 2002.
[27] "Berkeley Segmentation and Boundary Detection Benchmark and Dataset," 2003, http://www.cs.berkeley.edu/projects/vision/grouping/segbench.
[28] C. Van Rijsbergen, *Information Retrieval,* second ed. Dept. of Computer Science, Univ. of Glasgow, 1979.
[29] I. Abdou and W. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Proc. IEEE,* vol. 67, no. 5, pp. 753-763, May 1979.
[30] K. Bowyer, C. Kranenburg, and S. Dougherty, "Edge Detector Evaluation Using Empirical ROC Curves," *Proc. Conf. Computer Vision and Pattern Recognition,* 1999.
[31] D. Martin, C. Fowlkes, and J. Malik, "Learning to Detect Natural Image Boundaries Using Brightness and Texture," *Neural Information Processing Systems,* 2002.
[32] M. Meila and J. Shi, "Learning Segmentation by Random Walks," *Neural Information Processing Systems,* 2001.
[33] R.E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-Rated Predictions," *Machine Learning,* vol. 37, no. 3, pp. 297-336, 1999.
[34] M.I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation,* vol. 6, pp. 181-214, 1994.
[35] C. Chang and C. Lin, *LIBSVM: A Library for Support Vector Machines,* 2001, available at http://www.csie.ntu.edu.tw/cjlin/libsvm.
[36] C. Harris and M.J. Stephens, "A Combined Corner and Edge Detector," *Proc. Fourth Alvey Vision Conf.,* pp. 147-151. 1988,
[37] W. Förstner and E. Gulch, "A Fast Operator for Detection and Precise Locations of Distinct Points, Corners, and Centres of Circular Features," *Proc. Intercommission Conf. Fast Processing of Photogrammetric Data,* pp. 281-305, 1987.
[38] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation, and Depth.* Springer-Verlag, 1993
[39] D. Martin, C. Fowlkes, and J. Malik, "Local Boundary Detection in Natural Images: Matching Human and Machine Performance," *Proc. European Conf. Visual Perception,* 2003.
[40] R. Jonker and A. Volgenant, "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems," *Computing,* vol. 38, pp. 325-340, 1987.
[41] G. Carpaneto, S. Martello, and P Toth, "Algorithms and Codes for the Assignment Problem," *Annals of Operations Research,* vol. 13, pp. 193-223, 1988.
[42] A. Goldberg and R. Kennedy, "An Efficient Cost Scaling Algorithm for the Assignment Problem," *SIAM J. Discrete Math.,* 1993.
[43] B.V. Cherkassky and A.V. Goldberg, "On Implementing Push-Relabel Method for the Maximum Flow Problem," *Proc. Fourth Integer Programming and Combinatorial Optimization Conf.,* pp. 157-171, May 1995.

**David R. Martin** received the BSE degree summa cum laude in computer science from Princeton University in 1992, the MS degree in computer architecture from the University of California at Berkeley in 1998, and the PhD degree in computer vision from UC Berkeley in 2002. At Berkeley, he was supported in part by a US National Science Foundation Graduate Research Fellowship. He is currently an assistant professor in the Computer Science Department at Boston College. His professional interests include reliable software, parallel systems, and human and machine vision. He is a member of the IEEE.

**Charless C. Fowlkes** received the BS degree with honors in engineering and applied sciences from the California Institute of Technology in 2000. He is a PhD student at UC Berkeley in the Computer Science Division. His research at Berkeley has been supported by a UC MICRO fellowship and by a US National Science Foundation Graduate Research Fellowship. His research interests include the ecological statistics of natural scenes, image segmentation, and machine learning.

**Jitendra Malik** received the BTech degree in electrical engineering from Indian Institute of Technology, Kanpur, in 1980 and the PhD degree in computer science from Stanford University in 1985. In January 1986, he joined the University of California at Berkeley, where he is currently the Arthur J. Chick Endowed Professor of EECS, and the associate chair for the Computer Science Division. He is also on the faculty of the Cognitive Science and Vision Science groups. His research interests are in computer vision and computational modeling of human vision. His work spans a range of topics in vision including image segmentation and grouping, texture, stereopsis, object recognition, image based modeling and rendering, content based image querying, and intelligent vehicle highway systems. He has authored or coauthored more than a hundred research papers on these topics. He received the gold medal for the best graduating student in electrical engineering from IIT Kanpur in 1980, a Presidential Young Investigator Award in 1989, and the Rosenbaum fellowship for the Computer Vision Programme at the Newton Institute of Mathematical Sciences, University of Cambridge in 1993. He received the Diane S. McEntyre Award for Excellence in Teaching from the Computer Science Division, University of California at Berkeley, in 2000. He was awarded a Miller Research Professorship in 2001. He serves on the editorial boards of the *International Journal of Computer Vision* and the *Journal of Vision*. He is also a member of the scientific advisory board of the Pacific Institute for the Mathematical Sciences. He is a member of the IEEE.