Using Contours to Detect and Localize Junctions in Natural Images

Michael Maire[1], Pablo Arbeláez[1], Charless Fowlkes[2], and Jitendra Malik[1]
[1]University of California, Berkeley - Berkeley, CA 94720
[2]University of California, Irvine - Irvine, CA 92697
{mmaire,arbelaez,malik}@eecs.berkeley.edu, fowlkes@ics.uci.edu

Abstract

Contours and junctions are important cues for perceptual organization and shape recognition. Detecting junctions locally has proved problematic because the image intensity surface is confusing in the neighborhood of a junction. Edge detectors also do not perform well near junctions. Current leading approaches to junction detection, such as the Harris operator, are based on 2D variation in the intensity signal. However, a drawback of this strategy is that it confuses textured regions with junctions. We believe that the right approach to junction detection should take advantage of the contours that are incident at a junction; contours themselves can be detected by processes that use more global approaches. In this paper, we develop a new high-performance contour detector using a combination of local and global cues. This contour detector provides the best performance to date (F=0.70) on the Berkeley Segmentation Dataset (BSDS) benchmark. From the resulting contours, we detect and localize candidate junctions, taking into account both contour salience and geometric configuration. We show that improvements in our contour model lead to better junctions. Our contour and junction detectors both provide state of the art performance.

1. Introduction

We present a new, high-performance detector for contours in natural images, and use its output to detect and localize image junctions. These are each well-studied problems in computer vision; our novel contributions are the following:

- **Contour Detection:** We combine the use of local information derived from brightness, color, and texture signals, with global information obtained from spectral partitioning, to produce a contour detector. On the BSDS benchmark of 100 images, this provides the best results to date (comparison data is available for several algorithms tested over the last 4 years [10]).
- **Junction Detection and Localization:** Local junction detection is difficult, even for humans [18]. Yet much previous work in the computer vision community focused on operators based on local image patches [5, 8, 11, 14, 23]. These approaches detect the presence of edge energy at mutiple orientations, as might happen in textured areas, and as such provide a useful signal for structure from motion and object recognition. We are interested in a more classic notion of junctions, as in line drawing interpretation and perceptual organization. Our approach is based on detecting junctions as the approximate intersection points of contours in a way that makes us immune to the well-known problems of contour detectors in the neighborhood of a junction. Benchmarks for junction detection seem non-existent, so we develop a new one based on the BSDS. We compare our junction detector to baselines built on variants of the Harris detector [11].

The two main sections of this paper present our approach to and results on these twin problems. Discussions of relevant past work on contour and junction detection are embedded in their respective sections.

2. Contour Detection

There is an extensive literature on contour detection. For the purposes of this paper, we consider two main approaches to this task. A first family of methods aims at quantifying the presence of a boundary at a given image location through local measurements. Early local approaches, such as the Canny detector [3], model edges as sharp discontinuities in the brightness channel. A richer description can be obtained by considering the response of the image to a family of filters of different scales and orientations. An example is the Oriented Energy approach [19, 21], in which the filterbank is composed of quadrature pairs of even and odd symmetric filters. More recent approaches also take into account color and texture information and make use of learning techniques for cue combination [17, 6].

A second family of methods relies on integrating global

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

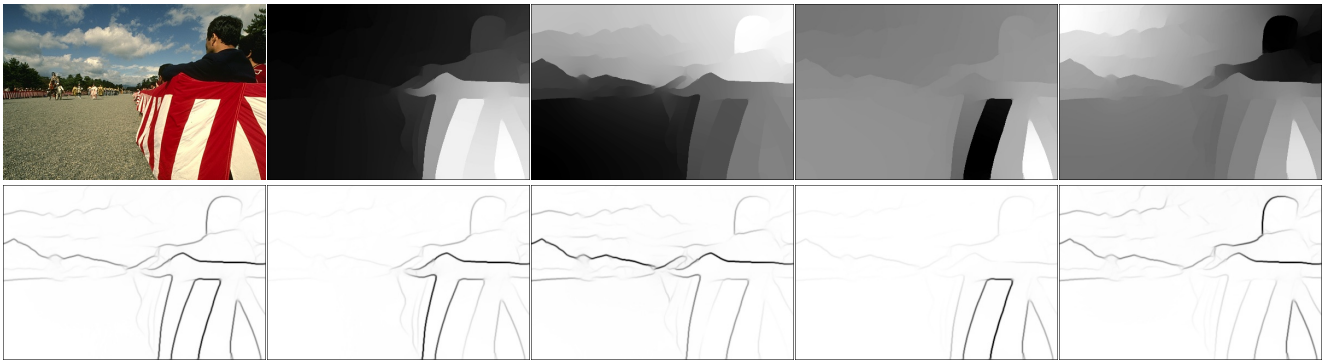

Figure 1. **Top:** Original image and first four generalized eigenvectors. **Bottom:** Maximum response over orientations θ of $sPb(x, y, \theta)$, and of $sPb_{\mathbf{v_j}}(x, y, \theta)$ for each eigenvector $\mathbf{v_j}$.

image information into the grouping process. Spectral graph theory [4] has often been used for this purpose, particularly, the Normalized Cuts criterion [16, 25]. In this framework, given an affinity matrix W whose entries encode the similarity between pixels, one defines $D_{ii} = \sum_j W_{ij}$ and solves for the generalized eigenvectors of the linear system:

$$(D - W)\mathbf{v} = \lambda D\mathbf{v} \tag{1}$$

Traditionally, after this step, clustering is applied to obtain a segmentation into regions. This approach often breaks uniform regions where the eigenvectors have smooth gradients. One solution is to reweight the affinity matrix [26]; others have proposed alternative graph partitioning formulations [9, 27, 29]. Recently, Zhu *et al*. [30] proposed detecting closed topological cycles in a directed edgel graph by considering the complex eigenvectors of the normalized random walk matrix. Although contour detection methods based on spectral partitioning have been reported to do well in the high precision / low recall regime, their performance is generally poor in the high recall / low precision regime [9, 29].

There is of course a much larger tradition in boundary detection and region segmentation. Classic approaches include the variational formulation introduced by Mumford and Shah [20], level-set methods [24] and techniques based on perceptual organization of contour outputs [15, 28]. Recent methods in contour detection include [7], where salient smooth curves are extracted by using a greedy approximate solution to the minimum-cover problem; [22], where the Conditional Random Fields framework is used to enforce curvilinear continuity of contours; and [1], where Ultrametric Contour Maps are obtained from hierarchical agglomerative clustering of regions.

2.1. Combining Local and Global Contours

We consider in this paper the detector of [17], whose output $Pb_\sigma(x, y, \theta)$ predicts the posterior probability of a boundary at each image pixel by measuring the difference in several feature channels on the two halves of a disc of radius σ centered at (x, y) and divided by a diameter at angle θ. In our experiments, we sample θ at 8 orientations in the interval $[0, \pi)$.

In order to detect fine as well as coarse structures, we consider brightness, color, and texture gradients at three scales: $[\frac{\sigma}{2}, \sigma, 2\sigma]$, where σ is the default scale of the Pb detector. We then combine linearly the local cues, denoted $\{G_i\}$, in a single multiscale oriented signal:

$$mPb(x, y, \theta) = \sum_{i=1}^{9} \alpha_i \cdot G_i(x, y, \theta) \tag{2}$$

In order to introduce global information, we consider the first $k + 1$ generalized eigenvectors of the system (1), noted $\{\mathbf{v_0}, ..., \mathbf{v_k}\}$, where the corresponding eigenvalues $\{\lambda_0, ..., \lambda_k\}$ are such that $0 = \lambda_0 \leq ... \leq \lambda_k$. In our experiments, we use $k = 8$.

We construct the affinity matrix W by using the *intervening contour* cue [9, 13], the maximal value of mPb along a line connecting two pixels. We then reshape each $\mathbf{v_j}$ in the size of the original image and extract its contours using Gaussian directional derivatives at multiple orientations θ, obtaining an oriented signal $sPb_{\mathbf{v_j}}(x, y, \theta)$. The information from different eigenvectors is then combined to provide the "spectral" component of our boundary detector:

$$sPb(x, y, \theta) = \sum_{j=1}^{k} \frac{1}{\sqrt{\lambda_j}} \cdot sPb_{\mathbf{v_j}}(x, y, \theta) \tag{3}$$

where the choice of the weights is motivated by the physical interpretation of generalized eigensystems as mass-spring systems [2]. Figure 1 presents an example. Note that, by extracting an oriented contour signal from the eigenvectors instead of producing hard segmentations, we eliminate the problem of large uniform regions sometimes being broken up by the Normalized Cuts algorithm.

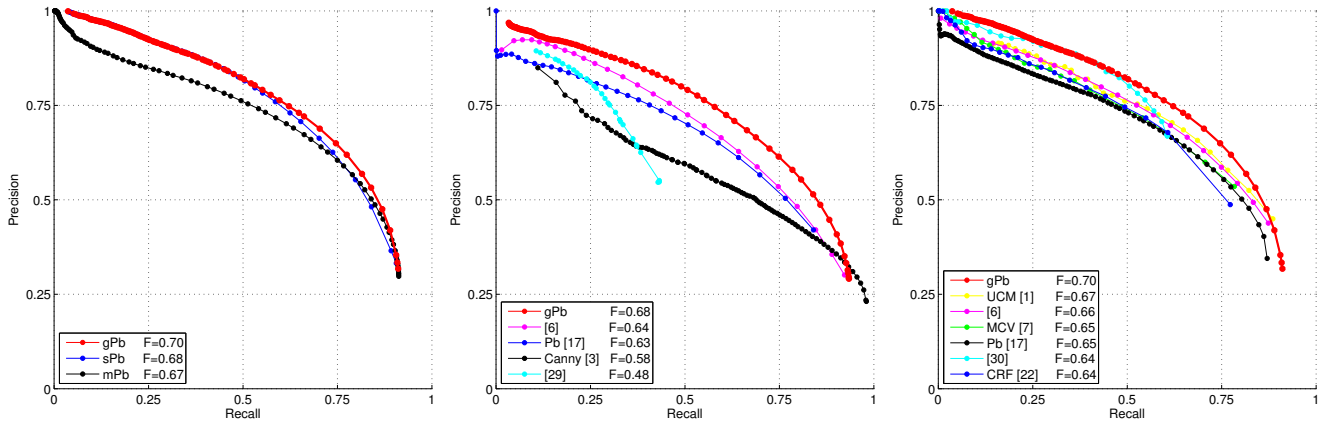

Figure 2. Evaluation of boundary detectors. **Left:** Comparison of the detectors in this paper. The spectral detector sPb improves the precision of the local signal mPb, while their combination gPb provides uniformly better performance. **Middle:** Comparison of gPb with leading boundary detectors on grayscale images. **Right:** Comparison of gPb with leading boundary detectors on color images. gPb obtains the highest F-measure $(2 \cdot Precision \cdot Recall/(Precision + Recall))$ to date on the BSDS benchmark.

The signals mPb and sPb convey different information, as the former fires at all the edges while the latter extracts only the most salient curves in the image. We found that a simple linear combination is enough to benefit from both behaviors. Our final **globalized probability of boundary** is then written as:

$$gPb(x, y, \theta) = \sum_{i=1}^{9} \beta_i \cdot G_i(x, y, \theta) + \gamma \cdot sPb(x, y, \theta) \quad (4)$$

where the weights are learned by gradient ascent on the F-measure.

A thinned, real-valued contour image can be obtained from gPb by considering the maximal response over orientations at each location and applying non-maximum suppression [3, 12].

2.2. Evaluation

Figure 2 presents the evaluation of our boundary detection results using the BSDS benchmark [10]. The precision-recall curves show that the reduction of false positives due to the use of global information in sPb is concentrated in the high thresholds, while gPb takes the best of both worlds, relying on sPb in the high precision regime and on mPb in the high recall regime. The mean improvement in precision of gPb with respect to the single scale Pb is 10% in the recall range $[0.1, 0.9]$. The gain in precision of the grayscale version of gPb with respect to the Canny detector is 17% in the same recall range. Qualitatively, this improvement translates as reduction of clutter edges and completion of contours in the output, as shown in Figure 3. The central and right panels of Figure 2 show that gPb compares favorably with the leading contour detection techniques evaluated on the BSDS [1, 3, 6, 7, 17, 22, 29, 30].

3. Junction Detection

Evidence from psychophysics [18] suggests that junctions are difficult to detect locally. However, much previous work in the computer vision community has focused on the development of operators based on local image patches [5, 8, 11, 14, 23]. While our approach still analyzes an image neighborhood, we do not necessarily rely on image information in the immediate vicinity of a junction. Rather, we choose the support of the neighborhood large enough with respect to the support of the boundary operator so that our algorithm may recover from errors in contour detection near a junction.

3.1. Local Operators

We use a version of the Harris operator [11] as a baseline with which to compare our algorithm. Given image I let G be a two-dimensional Gaussian smoothing kernel and define:

$$A(x, y) = G * [\nabla I \nabla I^T]\big|_{(x,y)} \quad (5)$$

where $*$ denotes convolution [14].

Let λ_1 and λ_2 be the eigenvalues of $A(x, y)$. The Harris corner operator is based on the observation that near a corner, both λ_1 and λ_2 are large and positive, whereas near an edge or featureless region, at least one of $\lambda_1, \lambda_2 \approx 0$. We define our Harris operator to be:

$$H(x, y) = \lambda_1 \lambda_2 / (\lambda_1 + \lambda_2) \quad (6)$$

Applying non-maximum suppression to $H(x, y)$ yields candidate junction locations.

Reliance on the grayscale image derivative ∇I leaves the Harris operator vulnerable to erroneous responses in textured regions. The next section describes a novel junction

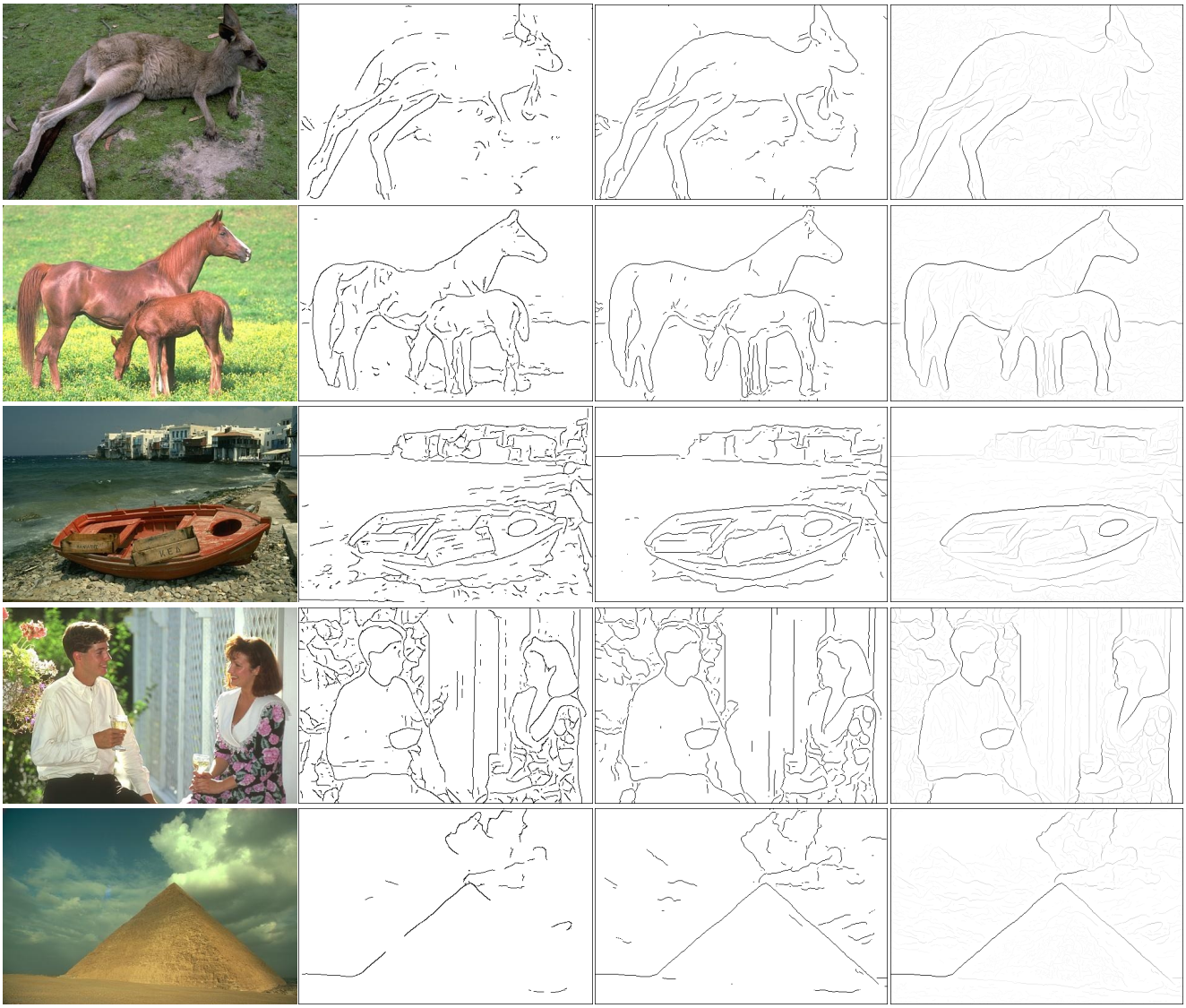

Figure 3. When compared with the local detector Pb, our detector gPb reduces clutter and completes contours. **From left to right:** Original image, thresholded Pb, thresholded gPb, and gPb. The thresholds shown correspond to the points of maximal F-measure on the curves in Figure 2.

detection algorithm based on Pb or its variants mPb, gPb, etc. To illustrate that our gains are partially due to this algorithm and not just the result of a better image derivative estimate, we include as a comparison point the Harris operator with ∇I computed from $Pb(x, y, \theta)$.

3.2. Contour-based Approach

Junctions may be viewed as points at which two or more distinct contours intersect. For each junction, we would like to recover its location and salience, and also identify the contours passing through it.

Given a set of contours defined by the non-maximum suppressed output of Pb (or a similar boundary detector), if one knew the junction locations, it would be easy to identify the associated contours. Conversely, if one knew which contours intersect at each junction, it would be easy to estimate the optimal locations. This suggests an EM-style algorithm. For an image neighborhood I_N:

1. Estimate the optimal junction location $L = (x_L, y_L)$ by minimizing its weighted distance from the contours $\{C_i\} \in I_N$

$$L = argmin_{(x,y)\in I_N} \sum_i w_i d(C_i, (x, y)) \quad (7)$$

where $d(C_i, (x, y))$ is the distance from contour C_i to point (x, y). In practice, we approximate each contour fragment C_i as a line segment and let $d(\cdot, \cdot)$ be the distance from a point to the line containing this segment. Hence, $d(\cdot, \cdot)$ is small provided the smooth continuation of the contour intersects the junction.

2. Update the weight w_i of each contour C_i in order to select only those contours passing close to the junction:

$$w_i = |C_i| \cdot \exp(-d(C_i, L)^2/\epsilon^2) \tag{8}$$

where $|C_i| = \sum_{(x,y)\in C_i} Pb(x, y)$ is the total contrast of contour C_i and ϵ is a parameter controlling the distance tolerance.

3. Repeat the above two steps for a set number of iterations or until the optimal junction location L reaches a fixed point. In practice, convergence is fast and in our experiments we capped the number of iterations at 5.

The weights are initialized by setting $w_i = |C_i| \; \forall i$.

Figure 4 provides an example of a set of contour fragments in an image neighborhood and their line segment approximations. Step 1 minimizes a weighted distance to these lines. Notice that some contours do not participate in the junction. Step 2 removes such distractors.

The error measure minimized in Step 1 is similar in spirit to that used by Forstner and Gulch [8] for corner localization. However, they minimize a sum of squared distances from lines through directed pixels and are vulnerable to clutter without the reweighting step.

Note that we need only consider neighborhoods I_N likely to contain a junction: those near existing endpoints of our approximately linear contour fragments.

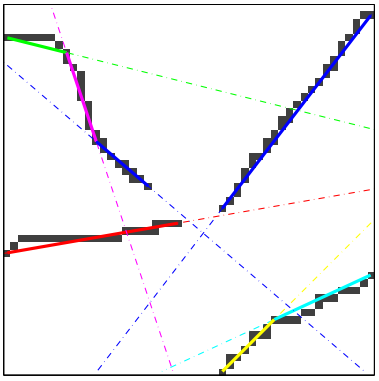

Figure 4. Contour fragments in an image neighborhood. We subdivide contours into approximately linear pieces and consider the corresponding straight line segments when localizing junctions.

Figure 5 shows the results of our junction localization algorithm. The left panel contains the Pb response in the neighborhood of a junction. The right panel displays each pixel with intensity equal to the weight of its corresponding contour after localization. Reweighting removes interfering contours which pass through the image neighborhood, but not the junction.

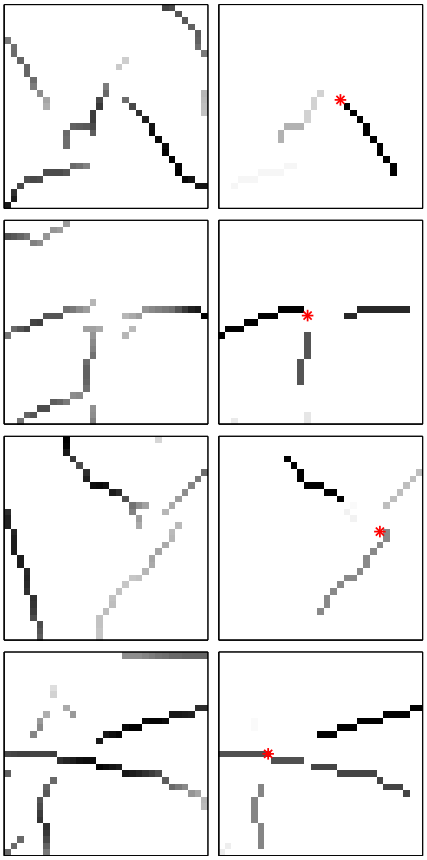

Figure 5. Junction localization. **Left:** Contours detected in an image neighborhood. **Right:** Contours reweighted according to their intersection with the detected junction (red star). Our reweighting technique removes clutter which is not part of the junction. The surviving contours are precisely those participating in the junction.

The above approach leaves open the question of junction salience. We would like a quantitative measure of junction strength in order to compare our results to local operators such as Harris. We design a "probability of junction" operator, Pj, motivated by the following considerations:

- Contours meeting at a junction should be salient.
- Junctions should be sharp. The intersecting contours should be nearly orthogonal.
- We wish to treat all junction types in the same framework. Let n be the degree of a junction (usually 2 or 3). We evaluate an n-junction by regarding it as the maximally salient 2-junction formed by its components.

The following Pj measure satisfies all three of these constraints:

$$Pj(x_L, y_L) \propto max_{C_u, C_v \in I_N} \{\sqrt{w_u w_v} \cdot \sin(\theta(C_u, C_v))\} \quad (9)$$

where $\theta(C_u, C_v)$ is the angle between contour fragments C_u and C_v at the junction location (x_L, y_L). The term $\sin(\theta(C_u, C_v))$ is a counterpart to extracting the component perpendicular to the dominant orientation, analogous to the second eigenvector of $\nabla I \nabla I^T$.

3.3. Junction Benchmark

To our knowledge, no standard dataset exists in the computer vision community for the purpose of benchmarking junction detection algorithms. We adapt the human ground truth of the BSDS to serve as a junction benchmark. From the ground truth segmentations, we extract 3-junctions as places where three or more regions intersect, and 2-junctions as locations of high curvature along human drawn boundaries. Just as for contour detection, the correspondence between machine and human marked junctions is used to produce a precision-recall curve for each junction detection algorithm.

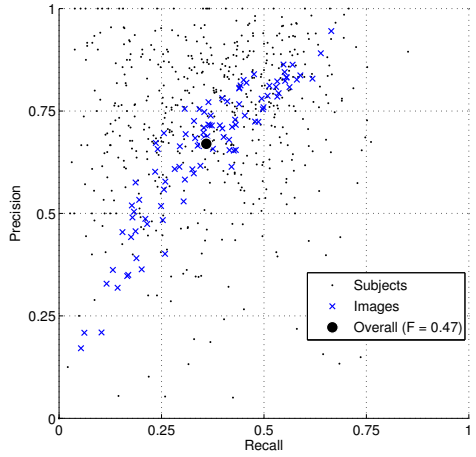

Figure 6. Human agreement on junction detection. Junction locations are automatically extracted from human marked ground truth contours in the BSDS. Points represent all pairwise comparisons of junction locations between subjects who marked the same image. The X's represent average agreement for each image. The overall average agreement is the maximum performance a junction detection algorithm could expect to achieve.

Human agreement on the junction detection task, as shown in Figure 6, is lower that human agreement on the boundary detection task. Contours added or omitted can more drastically change the junction structure, as can the degree to which subjects traced sharp details. Hence, we use a larger tolerance (6 pixels) for junction agreement. There is a lower ceiling (in terms of F-measure) on this task than on boundary detection. However, as the next section will show, the junction benchmark can distinguish the performance of our contour-based approach from that of traditional local operators.

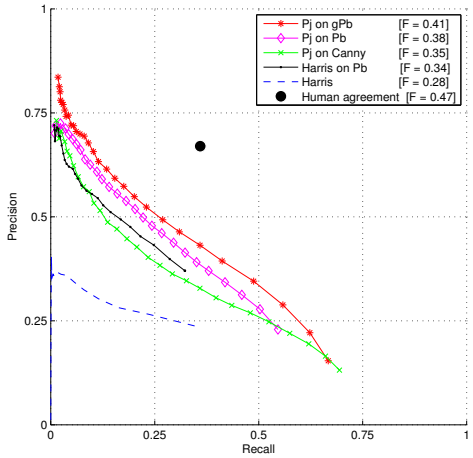

Figure 7. Evaluation of junction detectors. Performance improvements come from two sources: a better contour signal (notice the difference between standard Harris and Harris on Pb, as well as the difference between Pj on Canny, Pj on Pb, and Pj on gPb), and a better junction measure (notice the difference between Harris on Pb and Pj on Pb).

3.4. Evaluation

Figure 7 presents the results of evaluating several junction detectors on our BSDS-based junction benchmark. The baseline Harris operator on the grayscale image performs worst of all, with an F-measure of 0.28. Using Pb to estimate the derivatives serving as input to Harris boosts its F-measure to 0.34. Our contour-based junction detector Pj, using the best contour signal gPb, performs best of all at F=0.41. Human agreement on this task yields a ceiling of F=0.47. Figure 8 illustrates the qualitative aspects of our junction detector in comparison to the baseline Harris operator.

Our junction detector determines which contours participate in a junction (see Figure 5) in addition to estimating the junction salience. Figure 9 shows that we can use this information to restore missing contours near salient junctions.

4. Conclusion

In this paper, we have presented a unified framework for contour and junction detection and shown leading results on the BSDS benchmark. Since we have connectivity information associating junctions and contours, this allows us to go partway to producing idealized line drawings. We believe these results will serve as useful input for later stages of perceptual organization and object recognition.

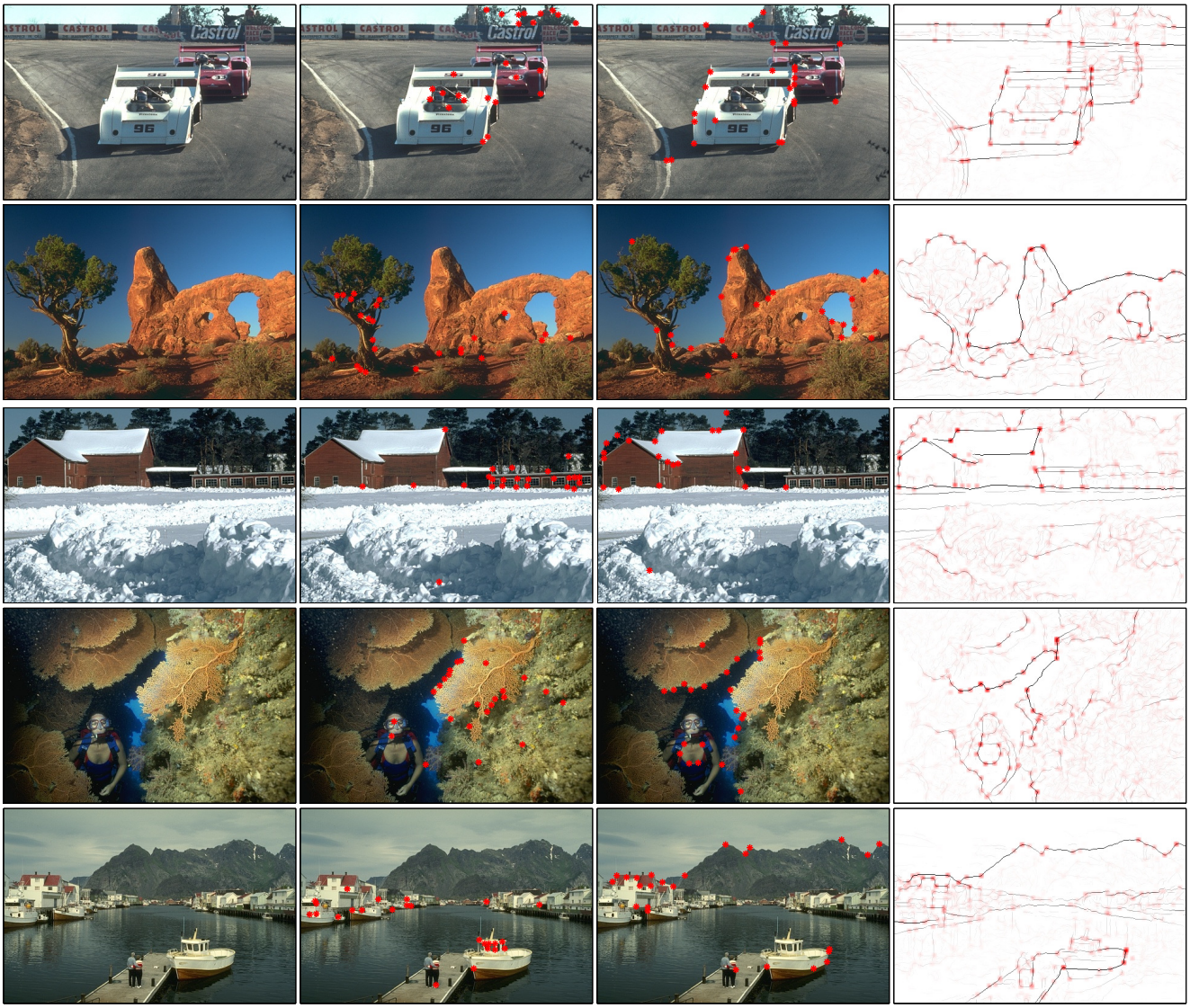

Figure 8. Comparison of Pj with the Harris operator. **From left to right:** Original image, top junctions detected by Harris, top junctions detected by Pj, junctions (Pj) (in red) overlayed on contours (gPb). For the purposes of display, we placed a Gaussian blob at each junction location, with brightness corresponding to junction salience. In each of the center images, we show the same number (25) of junctions. Additional results are provided in the supplemental material.

Acknowledgements

This work was supported by ONR MURI N00014-06-1-0734.

References

[1] P. Arbeláez. Boundary extraction in natural images using ultrametric contour maps. *POCV*, 2006.

[2] S. Belongie and J. Malik Finding boundaries in natural images: a new method using point descriptors and area completion. *ECCV*, 1998.

[3] J. Canny. A computational approach to edge detection. *PAMI*, 1986.

[4] F. Chung. *Spectral Graph Theory*. *CBMS*, 1997.

[5] R. Deriche, G. Giraudon A computational approach for corner and vertex detection. *IJCV*, 1993.

[6] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. *CVPR*, 2006.

[7] P. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. *POCV*, 2006.

[8] W. Forstner and E. Gulch. A fast operator for detection and precise localization of distinct corners. *ISPRS*, 1987.

[9] C. Fowlkes and J. Malik. How much does globalization help segmentation? Technical Report CSD-04-1340, UC Berkeley, 2004.

[10] C. Fowlkes, D. Martin, and J. Malik. The Berkeley Segmentation Dataset and Benchmark.

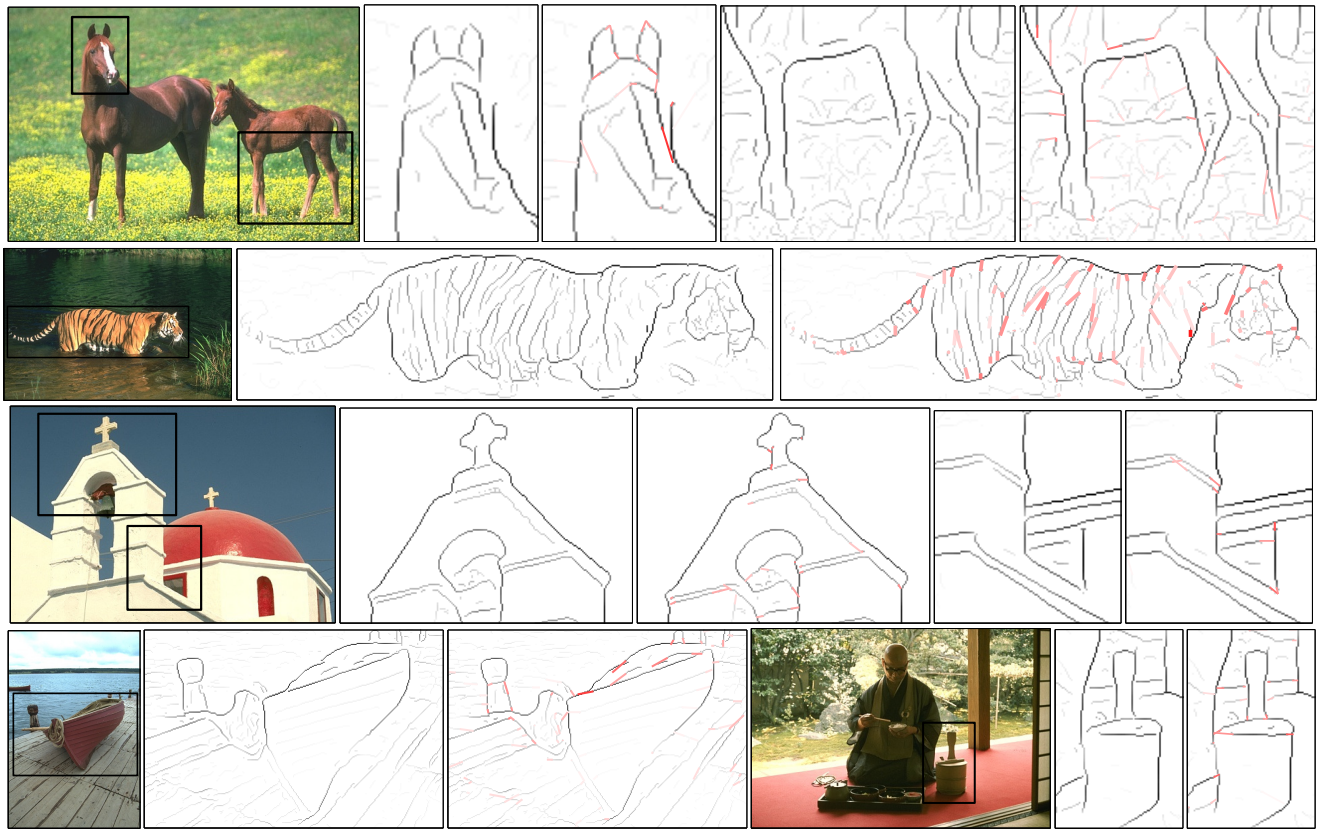

Figure 9. Junction restoration by extension of existing contours to detected junction locations. The magnified view of each boxed area shows contours before and after junction restoration. Extended contours are shown in red, with brightness corresponding to their estimated strength.

www.cs.berkeley.edu/projects/vision/grouping/segbench/.

[11] C. Harris and M. Stephens. A combined corner and edge detector. *Proc. of the 4th Alvey Vision Conference*, 1988.

[12] P. Kovesi. Matlab and octave functions for computer vision and image processing. http://www.csse.uwa.edu.au/ pk.

[13] T. Leung and J. Malik. Contour continuity in region-based image segmentation. *ECCV*, 1998.

[14] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 1998.

[15] S. Mahamud, L. Williams, K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *PAMI*, 2003.

[16] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001.

[17] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 2004.

[18] J. McDermott. Psychophysics with junctions in real images. *Perception*, 2004.

[19] M. Morrone and R. Owens Feature detection from local energy. *Pattern Recognition Letters*, 1987.

[20] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and variational problems. *Communications on Pure and Applied Mathematics*, 1989.

[21] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. *ICCV*, 1990.

[22] X. Ren, C. Fowlkes, and J. Malik. Scale-invariant contour completion using conditional random fields. *ICCV*, 2005.

[23] M. Ruzon and C. Tomasi Edge, junction, and corner detection using color distributions. *PAMI*, 2001.

[24] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, UK, 2 edition, 1999.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[26] D. Tolliver and G. Miller. Graph partitioning by spectral rounding: Applications in image segmentation and clustering. *CVPR*, 2006.

[27] S. Wang, T. Kubota, J. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *PAMI*, 2005.

[28] L. Williams and D. Jacobs. Stochastic completion fields: a neural model for illusory contour shape and salience. *ICCV*, 1995.

[29] S. Yu. Segmentation induced by scale invariance. *CVPR*, 2005.

[30] Q. Zhu, G. Song, and J. Shi. Untangling cycles for contour grouping. *ICCV*, 2007.