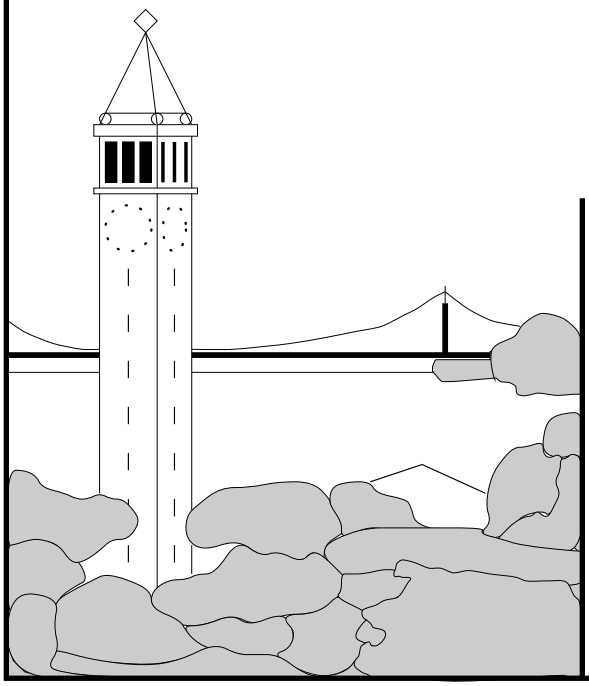
Mid-level Cues Improve Boundary Detection

Xiaofeng Ren, Charless Fowlkes and Jitendra Malik

{xren,fowlkes,malik}@eecs.berkeley.edu



Report No. UCB/CSD-5-1382

March 2005

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Abstract

While mid-level perceptual cues have long been of interest in the human vision community, their role in computer vision has remained limited. In this report, we evaluate several algorithms which make use of mid-level processing in order to improve boundary detection. Our first technique builds a probabilistic model of the relation between prototypical local shapes of edges and the presence or absence of a boundary. We also present a more explicit local model of curvilinear continuity using piecewise linear representations of contours and the Constrained Delaunay Triangulation (CDT). Lastly we consider a global random field on the whole CDT which captures continuity along with the frequency of different of junction types. All three models are trained on human labeled groundtruth. We measure how each model, by incorporating mid-level structure, improves boundary detection. To our knowledge, this is the first time that such cues have been shown quantitatively useful for a large set of natural images. Better boundary detection has immediate application in the problem of object recognition.

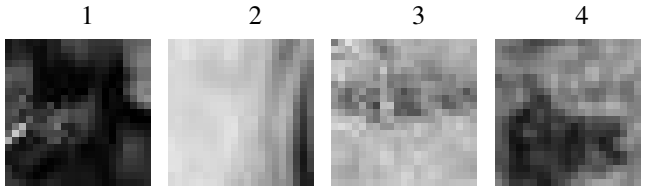
1. Introduction

Finding the boundaries of objects and regions in a scene is a problem of fundamental importance for computer vision. There is a large body of work on object recognition which relies on bottom-up boundary detection to provide information about object shape [4, 12, 8, 1, 7, 28]. Even in cases where simple intensity features are sufficient for object detection, e.g. faces, it is still desirable to incorporate bottom-up boundary detection in order to provide precise object segmentation [3, 36, 29]. The availability of high quality boundary location estimates will ultimately govern whether these algorithms are successful in real-world scenes where clutter and texture abound.

Boundaries are typically detected using some local operator. For example, the recent work by [16] trains a classifier that predicts the **p**robability of **b**oundary, Pb, at each pixel location using local brightness, texture and color gradient cues as features. Training data comes from a set of images where boundaries have been marked by human subjects. This elaborate algorithm still misses many true boundaries and falsely detects others not marked by humans. The authors argue that detection failures are primarily due to lack of context since human observers presented with only local image patches perform no better than the algorithm [18]. In this paper, we propose the use of mid-level cues in order to provide missing context and hence boost performance of such a local boundary detector.

In the human vision community there has been extensive research stressing the importance of mid-level cues to visual processing. However, these ideas, such as good continuation, symmetry, parallelism and familiar configurations, seem to have had little practical impact on the design of computer vision algorithms. One reason is that it's often hard to quantify when such mechanisms are actually performing some function that will ultimately be useful for recognition. We circumvent this issue by treating the output of mid-level processing as another boundary map, projecting back to the pixel grid as necessary. This gives a clear criterion for success, the output should be a better estimate of the true boundaries than the input was. Quantitative evaluation is made possible by utilizing three existing image collections which have been labeled with ground-truth boundaries: a set of 30 news photos of baseball players [20], 350 images of horses [3] and the BSDS300 [5], a boundary detection benchmark based on images of natural scenes.

We present three models of mid-level processing which take locally computed Pb [1] as input and provide quantifiably superior output. First, in Section 2 we describe a generic approach using clustering to model prototypical boundary shapes in the vicinity of a pixel. This can capture such cues as continuity, parallelism and familiar configuration. In Section 3 we focus more explicitly on curvilinear continuity, a specific but powerful mid-level cue closely related to boundary detection. We develop both a simple local model which makes decisions on pairs of edge segments using a linear classifier and a global model based on *conditional random fields* which jointly estimates probabilities for all edge segments. Lastly, the performance evaluation of the 3 models on 3 datasets is presented in Section 4.



(a) the outlook of a local boundary detector, 21x21 pixel patches

(b) context emerges in 71x71 patches

Figure 1: Purely local boundary detection can easily be fooled. Utilizing more context could greatly improve boundary detection.

2. Familiar Configuration

Consider the image patches in Figure 1(a). The task of a local boundary detector is to decide whether there is a boundary running through the center of the patch based on the pattern of intensities in a small neighborhood. Unfortunately, strictly local processing, no matter how clever, can

[1] Computed using MATLAB code downloaded from here [5]

be fooled. For example, 3 and 4 have higher brightness and texture contrast than 1 and 2 but in examining the larger context seen in Figure 1(b), it becomes evident that there are contours in patches 1 and 2, but not in patches 3 and 4.

Exploiting context algorithmically is clearly not simply a matter of elongating the support of our gradient magnitude calculations.[2] The structure here is much richer. In patch 1, the high contrast edges on either side, although not colinear, necessitate the presence of a boundary somewhere in the middle. Patch 2 features parallelism in addition to continuity. Patch 3 contains an isolated edge that fails to continue into a boundary while patch 4 lies in a textured region, surrounded by high contrast edges.

In order to capture these diverse patterns, we will seek out prototypical shape configurations, or *shapemes*, associated with both boundary and non-boundary points and learn how best to utilize them from "past experience", i.e., by training a statistical model based on human-marked groundtruth data. This is in the spirit of Wertheimer's principle of *familiar configuration* [32].

2.1 A Prototypical Shape Model

To discover prototypical shapes, we have to choose a shape descriptor and a scheme for clustering these descriptors. We choose the *geometric blur* [2] descriptor aligned to local boundary orientation and cluster using a *mixture-of-gaussian* model with symmetry constraints on the parameters. Shapemes have been used in previous work on object recognition by [19] who performed vector quantization on *shape context* descriptors to build an alphabet of local shape prototypes and coded images by their frequencies.

Let I be an input image and I_{pb} be its Pb (probability of boundary [16]) image, which associates a probability of boundary to each pixel. The *geometric blur* centered at location x, $GB_x(y)$, is a linear operator applied to I_{pb} whose value is another image given by the "convolution" of I_{pb} with a spatially varying Gaussian. GB_x has the property that points farther away from x are more blurred, making the descriptor robust to affine distortions. The value $GB_x(y)$ is the inner product of I_{pb} with a Gaussian centered at y whose standard deviation is $\alpha|y - x|$. We rotate the blurred image GB_x so that the locally estimated contour orientation at x is always horizontal. We choose $\alpha = 0.5$ and sample the blurred and rotated image GB_x at 4 different radii (increasing by a factor of $\sqrt{2}$) and 12 orientations, to obtain a feature vector of length 48.

We use the *mixture-of-gaussian* model to represent the distribution of shapes in this 48-dimensional feature space. Since the geometric blur descriptor is aligned to the contour tangent, there is a 180 degree orientation ambiguity. We explicitly enforce this symmetry by using $2m$ mixture components in m pairs having tied means (being rotated copies of one another) and equal, diagonal covariances. For our experiments we use $m = 64$ and fit the model using an easy adaptation of the typical *EM* algorithm.

Figure 2 provides a visualization of some mixture components for the horse dataset. These are the average Pb images of the components in the mixture that have the highest prior probability (ignoring clusters with very few members). These average shapes, albeit blurry, do represent interesting prototypical local shapes, or "shapemes": roughly speaking, they contain straight lines (row 6, col 6), curved lines (row 1, col 2), line endings (row 6, col 3), parallel lines (row 3, col 3), sharp corners (row 4, col 1), texture edges (row 6, col 1), etc. Taken together they provide a representation of the local context.

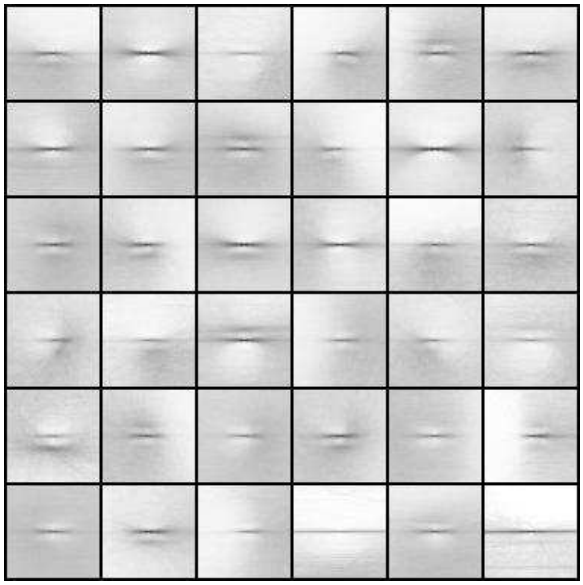


Figure 2: Examples of *shapemes*, or prototypical shape configurations, for the horse dataset. Shown are average Pb patches for each cluster in the mixture-of-gaussian.

We apply this contextual information to boundary detection as follows: each location x is associated with both the Pb value and the shape descriptor $GB(x)$. We can compute the posterior probability $\{p_1, \cdots, p_m\}$ of $GB(x)$ in the mixture-of-gaussian model. This gives us a new feature vector for x: $\{\log(Pb), \log(p_1), \cdots, \log(p_m)\}$. Then we can use this set of features to classify whether x is boundary or not.

We train a logistic classifier on this vector of features. Section 4 gives the quantitative comparison to raw Pb. The shapemes shown in Figure 2 are sorted by their weights in the logistic classifier, decreasing in order from left to right, top to bottom. As we are detecting boundaries of horses, extended contours (row 1, col 2) are positive evidence, possibly corresponding to the back or neck of a horse. Parallel lines with texture on one side and nothing on the other side

[2] Nor is it just a matter of recognition, patch 2 could be a crack between two rocks or the leg of an animal; patch 4 might be gravel or alfalfa.

(row 1, col 6) are also positively weighted, possibly recognizing horse legs. On the other hand, very straight lines (row 6, col 6) are uncommon for horse silhouettes. Edges embedded in texture (row 6, col 1) are also suppressed.

3. Curvilinear Continuity

In this section we consider a more specific form of mid-level information: *curvilinear continuity*, sometimes known as "good continuation" or "illusory contour completion". Inspired by Wertheimer and Kanizsa, the study of this phenomenon has a long and influential tradition in psychophysics as well as neurophysiology [22]. More recently, *ecological statistics* of contours have confirmed the existence of curvilinear continuity in natural images (e.g., [9]) and its scale-invariant properties [24].

In computer vision there exists a rich literature on how to model curvilinear continuity (e.g., [25, 23, 21, 33]). A typical approach consists of two stages: the first stage detecting line fragments based on brightness contrast, the second stage linking the fragments using various continuity measures. More recent developments focus on finding *closed* salient contours [6, 34, 15, 30].

Most of the previous approaches, however, are demonstrated on synthetic or simple real images and are not quantitatively evaluated on natural images. While we may be able to complete low-contrast edges using continuity, spurious completions are often introduced in the process. Is the net effect positive or negative? This question can only be answered by quantitative measurements. To the best of our knowledge, no such measurements have been done on a large, diverse set of real-world images.

Our approach starts with discretizing an image into a set of piecewise linear segments utilizing the *constrained Delaunay triangulation*. We then develop two models of curvilinear continuity on the resulting graph: a local model which classifies each pair of edges independently and a global model which enforces long-range probabilistic constraints on junctions using belief propagation on a *conditional random field*.

3.1 Constrained Delaunay Triangulation

As with the shapeme model, we use the local Pb operator [16] to detect candidate boundary locations. We then convert a map of per-pixel boundary probability into a discrete graph of line segments. This representation has many advantages over using the pixel grid: (1) By moving from 100,000 pixels to 1000 edges, it achieves high statistical and computational efficiency. (2) this discrete representation of the image is scale-invariant. A probabilistic model of continuity on the pixel grid depends on the resolution of the image. (3) By restricting completions to the edges in the graph, it partially solves the problem of having too many spurious completions. Moreover, we will show empirically that very little of the true boundary structure is lost in this discretization process.

Our discretization step starts with using Canny's hysteresis thresholding to trace the Pb contours and then recursively split them until each segment is approximately straight. Figure 3(a) shows an illustration of this linearization: for a given contour, let θ be the angle between segments $\overline{ca}$ and $\overline{cb}$. Pick the set of points $\{a, b, c\}$, in a coarse-to-fine search, such that the angle θ is maximized; if θ is below a threshold, we split the contour by adding a vertex at c and continue. A heuristic is added to handle T-junctions: when a vertex is very close to another line, we split this line and insert an additional vertex.

We use the *constrained Delaunay triangulation* to complete the piecewise linear approximations. The standard *Delaunay triangulation* (DT) is the dual of Voronoi diagrams and is the unique triangulation of a set of vertices in the plane such that no vertex is inside the circumcircle of any triangle. The constrained Delaunay triangulation (CDT) is a variant of of the DT in which a set of user-specified edges must lie in the triangulation. The CDT retains many nice properties of DT and is widely used in geometric modeling and finite element analysis. It was also recently been applied to image segmentation [35].

We use the TRIANGLE program [27] to produce CDTs as shown in Figure 4. The linearized edges extracted from the Pb contours become constrained edges in the triangulation which we refer to as gradient edges or G-edges. The remaining completion edges or C-edges are filled in by the CDT. Of particular interest to us is CDT's ability to complete contours across gaps in the local detector output. A typical scenario of contour completion is one low-contrast contour segment (missed by Pb) in between two high-contrast segments (both detected by Pb). If the low-contrast segment is short in length, chances are good that no other vertices lie in the circumcircle and CDT will correctly complete the gap by connecting the two high-contrast segments.

In order to establish groundtruth labels on the CDT edge for fitting our models, we need to transfer human marked boundaries from the pixel grid. This is accomplished by running a simple maximum-cardinality bipartite matching with a fixed distance threshold between the human marked boundaries and the CDT edges. We label a CDT edge as

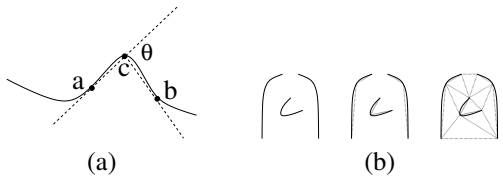


Figure 3: Building a discrete graph. **(a)** we recursively split a line until the angle θ is below a threshold. **(b)** an illustration of the process: the input edge map, the linearization, and the Constrained Delaunay Triangulation.

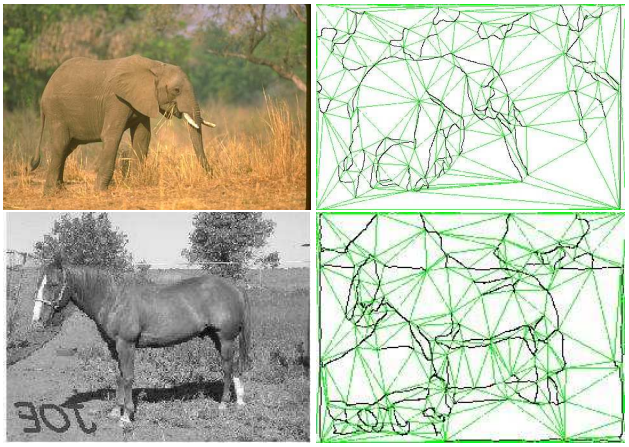


Figure 4: Examples of CDT triangulations. G-edges (gradient edges detected by Pb) are in black and C-edges (completed by CDT) in green. Note how CDT manages to complete gaps on the front legs of the elephant and on the back of the horse.

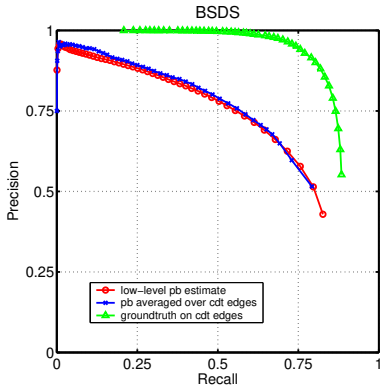


Figure 5: This Precision-Recall curve verifies that moving from pixels to the CDT completion doesn't give up any boundaries found by the local measurement and is able to push up the recall by completing some gaps in the local measurement. For comparison, we show the performance of the training data on the CDT edges.

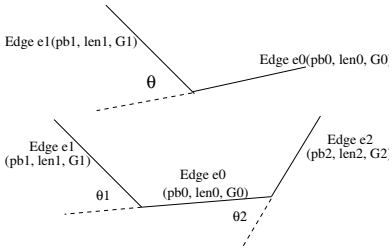


Figure 6: **(a)** A simple 2-edge model of curvilinear continuity. each edge has an associated set of features. continuity is measured by the angle θ. **(b)** evidences of continuity come from both ends of edge e_0. The new"probability of boundary" for e_0 is the product of the 2-edge model on both pairs (e_0, e_1) and (e_0, e_2).

boundary if 75% of the pixels lying under the edge are matched to human boundaries; otherwise we label it as non-boundary.

Figure 5 shows the performance of the local boundary detector Pb as well as the performance when we assign the average underlying Pb to each CDT edge.[3]

Figure 5 shows that moving from pixels to the CDT completion seldom gives up any boundaries found by the local measurement. The green curve documents the soft ground-truth labellings of the CDT edges (percentage matched to human marked pixels). This is the target output of the two learning algorithms we describe next. The gap between the asymptotic recall of Pb and the ground-truth shows that the CDT is even completing a few contours which are completely illusory (i.e. there was no local evidence)

3.2 A local continuity model

Each edge e in a CDT graph is naturally associated with a set of features including the average Pb of pixels along the edge e and whether it is a G-edge (detected in Pb) or C-edge (completed by the CDT). The local context of e includes these features and those of neighboring edges in the CDT graph. We now describe a simple model of *local* curvilinear continuity using this local context of e.

Consider the simplest case of context: a pair of connected edges (Figure 6(a)). Each edge can be on or off (being a true boundary or not), hence four possible labelings of this pair. However, the groundtruth contours in our datasets are almost always closed; line endings and junctions are rare. Therefore we make the simplifying assumption that there are only two possible labelings: either both of them are on, or both are off.

Our best local model uses as features $\overline{Pb}$, average Pb over the pair of edges; G, an indicator variable whether both of the edges are G-edges, and θ, the angle formed at the connection of the pair. We use logistic regression to fit a linear model to these features. We have found that logistic regression performs as good as other classifiers (we also tested support vector machines and hierarchical mixture of experts). It also has the advantage of being computationally efficient and simple to interpret.

To evaluate the local continuity model, we use the classifier to assign a new "probability of boundary" value to each edge e. Consider Figure 6(b): evidence of continuity comes from both ends of an edge e_0, as a contour at e_0 would have

[3]Throughout this paper, performance is evaluated with using a **precision-recall curve** which shows the trade-off between false positives and missed detections. For each given thresholding t of the algorithm output, above threshold boundary points are matched to human-marked boundaries H and the precision $P = P(H(x, y) = 1|Pb(x, y) > t)$ and recall $R = P(Pb(x, y) > t|H(x, y) = 1)$ are recorded (see [16] for more discussion).

to continue in both directions. We assume that these two sources of information are independent and take a product. Let $X_e = 1$ if the pixels corresponding to e lie on a true boundary and 0 otherwise. The logistic model gives an estimate of $P(X_{e_0} = 1, X_{e_1} = 1)$, the posterior probability that the pair of edges (e_0, e_1) are both true. If S_1 and S_2 are the two sets of edges connecting to e_0 at the two ends we define the new boundary operator Pb_L under the 2-edge product model to be:

$$Pb_L = \max_{e_1 \in S_1} P(X_{e_0}{=}1, X_{e_1}{=}1) \cdot \max_{e_2 \in S_2} P(X_{e_0}{=}1, X_{e_2}{=}1)$$

The quantitative performance evaluation of Pb_L against Pb is shown in Section 4.

To evaluate relative contribution of each feature, we have also fit classifiers to subsets of features and compared their performance. The results on the baseball player dataset are shown in Figure 7. Performance is evaluated with both a precision-recall curve and a cross-entropy loss L [10].

We observe that $\overline{Pb}$ is the most useful feature with $L = 0.418$. Angle θ by itself is not as useful; however, when combined with $\overline{Pb}$, θ helps reduce the loss to 0.385. G by itself is quite informative, as 85% of the positive examples have $G = 1$ (both edges being G-edges). When the three features are combined, the loss is reduced to 0.374. We have experimented with many additional features but they are at most marginally useful, hence not included in the model.

We have also considered variants such as: a second-layer classifier to combine information from the two ends of e_0; a 3-edge classifier which directly takes as input the features from triples of edges; and a full 4-way classification on each pair of edges. The simplest 2-edge product model described above performs as well as these variants.

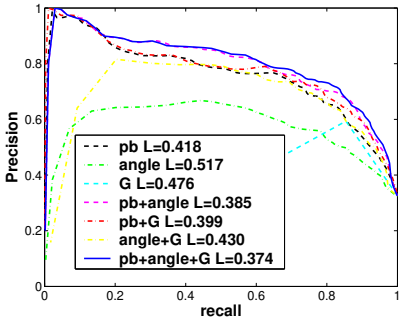


Figure 7: Evaluating combinations of features with precision-recall curves and cross-entropy loss L. $\overline{Pb}$ is the most powerful feature and the continuity θ significantly improves the performance.

3.3 A global continuity model

In order to capture longer range statistics of boundary presence we also consider a global probabilistic model built on top of the CDT. We would like to utilize the same measurements as in the local-model (Pb,G,θ) along with the frequency with which junctions of different degrees appear. In order to capture the dependency of edge presence on these features, we use a conditional random field (CRF) model as first introduced by [14].

Unlike the MRF models traditionally used in vision which model the joint distribution of image measurements and hidden labels, a CRF focus directly on the conditional distribution of labels given the observations. One key advantage from our perspective is that the observed variables need not be conditionally independent given the hidden variables. This allows much greater freedom in choosing model features. Conditional random fields have been proposed as a method for image segmentation by [13, 26, 11], however, the focus there is on pixel-level labeling rather than mid-level tokens.

We base the independence structure of our hidden variables on the topology of the CDT. Recall the random variable X_e whose value is 1 if the pixels corresponding to e lie on a true boundary. Let X_v be the collection of variables for all edges which intersect at a vertex v of the CDT. We consider log-linear distributions over the collection of edges of the form

$$P(X|I,\Theta) = \frac{e^{\{\sum_e \phi(X_e|I,\Theta) + \sum_V \psi(X_V|I,\Theta)\}}}{Z(I,\Theta)}$$

The ϕ potential function captures the extent to which the image evidence I supports the presence of a boundary under edge e. ψ describes the continuity conditions at a junction between contour segments.

Our edge potential is given by

$$\phi(X_e|I,\Theta) = \beta Pb_e X_e$$

where Pb_e is the average Pb recorded over the pixels corresponding to edge e. The vertex potential is given by

$$\begin{aligned}\psi(X_V|I,\Theta) &= \sum_{i,j} \alpha_{i,j} \mathbf{1}_{\{\deg_g = i, \deg_c = j\}} \\ &\quad + \gamma \mathbf{1}_{\{\deg_g + \deg_c = 2\}} f(\theta)\end{aligned}$$

where $\deg_g$ is the number of G-edges at vertex V for which $X_e = 1$ and similarly $\deg_c$ is the number of C-edges which are turned on. When the total degree of a vertex is 2, γ weights the continuity of the two edges. f is a function which is smooth and symmetric around $\theta = 0$ and falls of as $\theta \to \pi$. If the angle between the two edges is close to 0, they form a good continuation and $\gamma f(\theta)$ is large and they are more likely to both be turned on.

Our model has the collection of parameters $\Theta = \{\alpha, \beta, \gamma\}$. We fit Θ by maximizing the log likelihood. Since the likelihood is log-linear in the parameters, taking a derivative always yields a difference of two expectations. For example, the derivative with respect to the continuation

parameter γ for a single training image/ground truth labeling, (I, X) is:

$$\begin{aligned}
&\frac{\partial}{\partial\gamma}\log\left(\frac{e^{\{\sum_e \phi(X_e|I,\Theta)+\sum_V \psi(X_V|I,\Theta)\}}}{Z(I_n,\Theta)}\right)\\
&= \sum_V \frac{\partial}{\partial\gamma}\{\gamma \mathbf{1}_{\{\deg_g+\deg_c=2\}} f(\theta)\} - \frac{\partial}{\partial\gamma}\log Z(I_n,\Theta)\\
&= \sum_V \mathbf{1}_{\{\deg_g+\deg_c=2\}} f(\theta)\\
&\quad -\left\langle \sum_V \mathbf{1}_{\{deg_g+\deg_c=2\}} f(\theta)\right\rangle_{P(X|I,\Theta)}
\end{aligned}$$

The first term is the observed sum of $f(\theta)$ on degree 2 vertices while the second term is the expectation under the model given our current setting of the parameters. When the model expectations match those observed in the training data, we have found the maximum likelihood setting of the parameters. Until we reach that point, we take a small step in the gradient direction. Parameters typically converge after a few hundred steps.

Unfortunately, computing the expectations of our features with respect to model parameters is intractable. Unlike the sequence modeling tasks where conditional random fields were first investigated our graph is not tree structured, it contains many triangles (among other loops). We approximate the edge and vertex degree expectations using loopy belief propagation [31]. For the graphs in question, belief propagation appears to converge quickly to a reasonable solution.

We find that the parameters learned from groundtruth boundary data match our intuition well. For example, the weight $\alpha_{1,0}$ is much smaller than $\alpha_{2,0}$, indicating that line endings are less common than continuation and reflecting the prevalence of closed contours. For degree 2 vertices, we find $\alpha_{2,0} > \alpha_{1,1} > \alpha_{0,2}$, indicating that continuation along G-edges is preferable to invoking C-edges.

Once the model has been trained, we use belief propagation to estimate the marginal distributions $\{X_e\}$ on the edges of the CDT and then project these down to the pixel grid, yielding Pb_G which is compared against the other two models.

4. Results: How useful are Mid-level Cues?

We have described three different algorithms, each which outputs a new estimate of the boundary probability at each pixel: Pb_S, the output of the classifier with shapeme responses as features, Pb_L, the local model on the CDT and Pb_G, the global random field model. In order to evaluate these, we use three human-segmented datasets: the baseball player dataset split into 15 for training and 15 for testing, the horse dataset split into 170 training and 170 test images, and the Berkeley Segmentation Dataset [5] (BSDS) which contains 200 training images and 100 test images of various natural scenes. Performance on these datasets is quantified using the precision-recall framework as in [16].

These quantitative comparisons clearly demonstrate that mid-level information *is* useful in a generic setting. The use of shapemes for local context improves boundary detection, especially in the low-recall/high-precision range. Both models of curvilinear continuity outperform Pb and the shapeme model. The global model, which is able to combine local evidence of continuity and global constraints such as closure, performs the best. The improvement is most noticeable in the low-recall/high-precision range which corresponds to the case of boosting the most prominent boundaries and suppressing background noise. These boundaries are typically smooth; thus continuity helps suppress false positives in the background. This is also evident in the examples shown in Figure 9.

We also observe that the benefit of continuity on the baseball player and horse dataset is much larger than that on the BSDS dataset. As we may tell from the precision rates of Pb, this baseball and horse datasets are harder. This is in part because the groundtruth for these data sets only includes the boundary of the prominent foreground object. In all cases, the remaining *semantic gap* may best be closed by detecting objects in the scene using the mid-level boundary map and then "cleaning up" the boundaries in a top-down fashion. In the case of the BSDS, this will require the development of systems which can recognize thousands of different object categories.

5. Conclusion

We have described mid-level processing which have a verifiably favorable impact on the problem of boundary detection.

Clustering local boundary shape is a flexible technique for incorporating intermediate context such as continuity and contrast normalization as well as familiar configurations. The local model of curvilinear continuity, though quite simple, yields a significant performance gain. The global model, by making long-range inference over local continuity constraints, is the most successful in utilizing mid-level information.

The key step in our approach to modeling continuity is moving from pixels to the piecewise linear approximations of contours and the constrained Delaunay triangulation. This provides a scale-invariant geometric representations of images which tends to complete gaps in the low-level edge map. Moving from 100,000 pixels to 1000 Delaunay edges is also important as it yields huge gains in both statistical and computational efficiency.

We have shown that the outputs of our algorithms are quantifiably better than a low-level edge detector on a wide

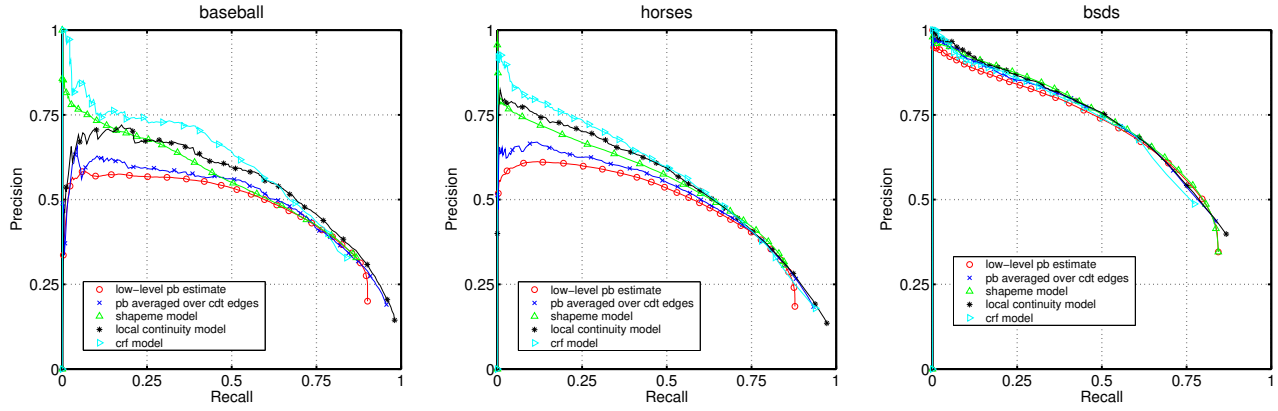


Figure 8: Pixel-based precision-recall evaluations comparing Pb_S, Pb_L and Pb_G to Pb. All three techniques improve boundary detection on all three datasets and the overall ordering of the curves is generally preserved across datasets.

variety of natural images. We hope these models will find immediate applications in object recognition.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. 8th Int'l. Conf. Computer Vision*, volume 1, pages 454–461, July 2001.

[2] A. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, 2001.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proc. 7th Europ. Conf. Comput. Vision*, volume 2, pages 109–124, 2002.

[4] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.

[5] Berkeley Segmentation Dataset, 2002. http://www.cs.berkeley.edu/projects/vision/bsds.

[6] J.H. Elder and S.W. Zucker. Computing contour closures. In *Proc. Euro. Conf. Computer Vision*, volume I, pages 399–412, Cambridge, England, Apr 1996.

[7] P. Felzenszwalb. Learning models for object recognition. In *CVPR*, 2001.

[8] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. 7th Int'l. Conf. Computer Vision*, pages 87–93, 1999.

[9] W. S. Geisler, J. S. Perry, B. J. Super, and D. P. Gallogly. Edge co-occurrence in natural images predicts contour grouping performance. *Vision Research*, 41:711–724, 2001.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: data mining, inference and prediction*. Springer-Verlag, 2001.

[11] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labelling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[12] D.P. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):850–863, Sept. 1993.

[13] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, 2003.

[14] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, 2001.

[15] S. Mahamud, L.R. Williams, K.K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(4):433–444, 2003.

[16] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. In *Advances in Neural Information Processing Systems 15*, 2002.

[17] 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. In *Proc. 8th Int'l. Conf. Computer Vision*, volume 2, pages 416–423, 2001.

[18] D. Martin, C. Fowlkes, L. Walker, and J. Malik. Local boundary detection in natural images: Matching human and machine performance. In *European Conference on Visual Perception [Perception, 32 supp, p. 55]*, 2003.

[19] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *Proc. IEEE Conf. Comput. Vision and Pattern Recogn.*, December 2001. to appear.

[20] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004.

[21] D. Mumford. Elastica and computer vision. In Chandrajit Bajaj, editor, *Algebraic Geometry and Its Applications*, pages 491–506. Springer Verlag, 1994.

[22] S. Palmer. *Vision Science: Photons to Phenomenology*. MIT Press, 1999.

[23] P. Parent and S.W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(8):823–39, Aug. 1989.

[24] X. Ren and J. Malik. A probabilistic multi-scale model for contour completion based on image statistics. In *Proc. 7th Europ. Conf. Comput. Vision*, volume 1, pages 312–327, 2002.

[25] A. Shashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. In *Proc. 2nd Int. Conf. Computer Vision*, pages 321–7, 1988.

[26] N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Learning and inferring image segmentations with the gbp typical cut algorithm. In *ICCV*, 2003.

[27] J. Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *First Workshop on Applied Computational Geometry*, pages 124–133, 1996.

[28] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *ECCV*, 2002.

[29] Z.W. Tu, X.R. Chen, A.L Yuille, and S.C. Zhu. Image parsing: segmentation, detection, and recognition. In *ICCV*, 2003.

[30] S. Wang, T. Kubota, and J. Siskind. Salient boundary detection using ratio contour. In *Advances in Neural Information Processing Systems 16*, 2003.

[31] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 2000.

[32] M. Wertheimer. Laws of organization in perceptual forms (partial translation). In W.B. Ellis, editor, *A sourcebook of Gestalt Psychology*, pages 71–88. Harcourt Brace and Company, 1938.

[33] L.R. Williams and D.W. Jacobs. Stochastic completion fields: a neural model of illusory contour shape and salience. In *Proc. 5th Int'l. Conf. Computer Vision*, pages 408–15, 1995.

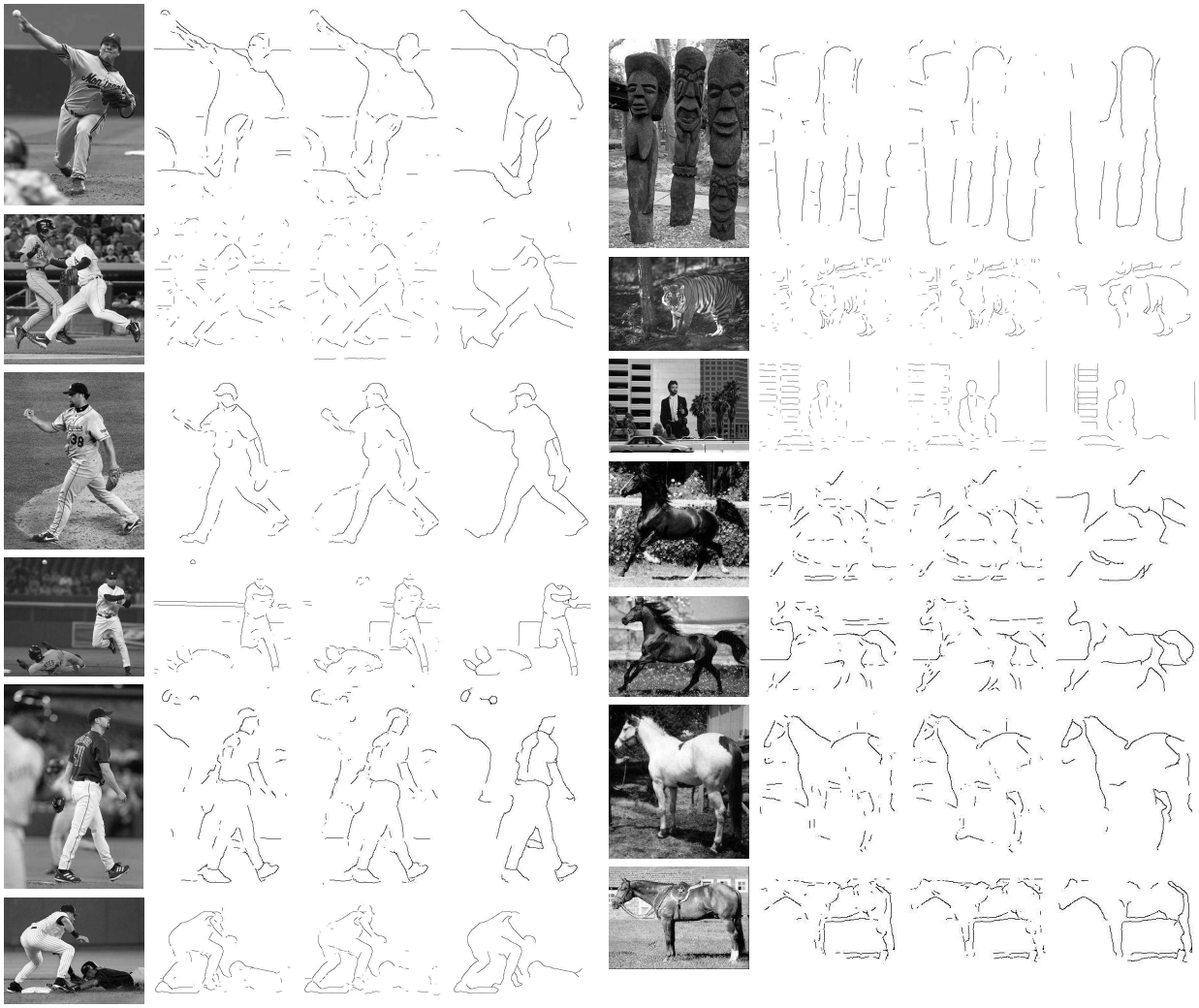


Figure 9: Example results on the three data sets. The three columns of edge maps show the local boundary detector *Pb*, the shapeme model, and the CRF model respectively. The algorithms outputs have been thresholded at a level which yields 2000 boundary pixels for the baseball/BSDS images and 1000 pixels for the smaller horse images.

[34] L.R. Williams and K.K. Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. *Int'l. Journal of Computer Vision*, 34(2/3):81–96, 1999.

[35] Q. Wu and Y. Yu. Two-level image segmentation based on region and edge integration. In *Proc. DICTA*, pages 957–966, 2003.

[36] S. Yu, R. Gross, and J. Shi. Concurrent object segmentation and recognition with graph partitioning. In *Advances in Neural Information Processing Systems 15*, 2002.