Discriminative Generative Contour Detection

Chao Zhang[1]
bjzchao@cn.ibm.com
Xiong Li[2]
flit.lee@gmail.com
Xiang Ruan[3]
gen@omm.ncl.omron.co.jp
Yuming Zhao[2]
arola_zym@sjtu.edu.cn
Ming-Hsuan Yang[4]
mhyang@ucmerced.edu

[1] IBM China Research,
China
[2] Shanghai Jiao Tong University,
China
[3] Omron Corporation,
Japan
[4] University of California Merced,
United States

Abstract

Contour detection is an important and fundamental problem in computer vision which finds numerous applications. Despite significant progress has been made in the past decades, contour detection from natural images remains a challenging task due to the difficulty of clearly distinguishing between edges of objects and surrounding backgrounds. To address this problem, we first capture multi-scale features from pixel-level to segment-level using local and global information. These features are mapped to a space where discriminative information is captured by computing posterior divergence of Gaussian mixture models and then used to train a random forest classifier for contour detection. We evaluate the proposed algorithm against leading methods in the literature on the Berkeley segmentation and Weizmann horse data sets. Experimental results demonstrate that the proposed contour detection algorithm performs favorably against state-of-the-art methods in terms of speed and accuracy.

1 Introduction

Object contour is of prime importance as it contains essential visual information, such as shape and identity that finds numerous applications. Contour detection is a fundamental problem in computer vision which is closely related to other tasks, e.g., segmentation, shape analysis, pose estimation, visual saliency, and object recognition, to name a few.

Numerous contour detection methods have been proposed in the literature [16]. The Canny edge detector [6] computes image gradients with non-maximum suppression which facilitates contour extraction. The compass operator [20] uses distributions of color pixels to determine the orientation of a diameter that maximizes the difference between two halves of a circular window, thereby identifying edges and contours. A tree-structured boosted edge learning (BEL) method [8] is proposed by selecting a large number of features extracted from image patches and combining them across different scales for edge and object boundary detection. However, this method has high computational complexity as it involves a large number of features. Stein et al. [21] propose a method based on over-segmented image regions and appearance as well as motion cues to train a classifier for finding object boundaries. Ren [18] introduces a multi-scale boundary detection algorithm by combining strength of both large-scale and small-scale detection results. With this multi-scale approach, significant improvements are demonstrated in detecting boundaries on large sets of natural images. The global probability of boundary (gPb) detector [3, 13] combines multiple local cues in a probabilistic framework based on spectral clustering, which shows state-of-the-art performance on the Berkeley segmentation data Set (BSDS500). Most recently, Papari et al. [15] use steerable filters to construct a model with multilevel inhibition terms to remove spurious edges in textured regions for contour detection.

In this work, we propose a learning algorithm for contour detection based on multi-level visual cues. We extract pixel-level features that integrate both local and global visual information. In addition, segment-level features are extracted to exploit structural information of contours. All the features are mapped to a score space [10, 11] for extracting discriminative ones via a novel algorithm based on posterior divergence of Gaussian mixture models. The motivates of introducing the feature mapping will be further explained in the second part of Section 4. A random forest classifier is trained based on these features for contour detection. In addition, a contour completion method based on the Gestalt theory is adopted to further improve the detection results. We evaluate the proposed algorithm with state-of-art methods on several databases including the Berkeley segmentation and Weizmann horse data sets. Experimental results bear out feature selection from multi-scale visual cues via posterior divergence with a random forest classifier facilitates effective contour detection in natural images.

2 Pixel-Level Features

Pixel-level features provide raw and basic visual cues for detection of object contours. To capture effective visual information on the pixel level, we extract local and global features at multiple scales. Local visual cues extracted from edges are first exploited in order to account for object contours at different scales. Global information extracted from visual saliency is then incorporated to provide cues of salient objects in the scenes. These features are integrated to form effective pixel-level features to represent object contours.

2.1 Multi-Scale Point Features

Basic point (pixel) features have been widely used for representing edge information of grayscale and color images such as image gradients, texture inhibition, brightness and color gradients as well as compass operators. As these features capture different visual information, we extract and combine these features for contour detection.

Image Gradient. Each image I is convolved with a Gaussian kernel of width σ to compute its gradient ∇I. The magnitude $|\nabla I|$ reveals the strength of an edge at each pixel and the direction or angle θ_∇ contains the intensity discontinuity information. The magnitude (MG) and direction (DG) of Gaussian gradients are extracted as local features.

Texture Inhibition. In order to remove small edges in highly textured regions, we use the inhibition term that suppress the response on the texture regions based on steerable filters [15]. It is computed by the convolution of the Gaussian gradient magnitude with the inhibition term (IT):

$$t(x,y) = \{V_o * |\nabla I|\}(x,y) + \mathrm{re}\{e^{2i\theta_\nabla(x,y)}[\{V_2 * |\nabla I|\}(x,y)]\}, \tag{1}$$

where $\mathrm{re}\{\cdot\}$ returns the real part and $*$ is the convolution operator. The steering bases are defined by $V_0(\rho,\phi) = \frac{\rho^2}{2}$, $V_2(\rho,\phi) = \frac{\rho^2}{2}e^{2i\phi}$, which are controlled by two parameters ρ and ϕ.

It can be shown that the difference of magnitude of Gaussian gradient and the inhibition term is the difference of Gaussian (DoG) feature [15] which provides useful edge information.

Brightness and Color Gradients. The probabilistic boundary (Pb) [14] and generalized probabilistic boundary (gPb) methods [3, 13] exploit brightness, color, texture and segmented regions to detect contours. For computational efficiency, we exploit the brightness and color gradients in this work. Similar to [14], we use a circular disc of radius r at pixel (x,y) and split into two half discs by a diameter at angle θ where they are represented by histograms of brightness and color in the CIELAB space. We compute the χ^2-distance between two histograms of half discs to compute the oriented gradient $G(x,y,\theta,r)$, thereby encoding both the brightness gradient (BG) and color gradient (CG) features.

Compass Operator. The compass operator [20] determines the orientation of a diameter which maximizes the difference between two half discs of a circular compass at each pixel (x,y). It detects edges without assuming that the regions on both sides have constant color by pixel distributions rather than the means. The distance between two color signatures is computed by $d_{ij} = 1-\exp(-E_{ij}/\gamma)$, where E_{ij} is the Euclidean distance between color i and color j, and γ is a constant. The distance between color signatures of equal mass of half discs S_1 and S_2 are computed by aggregating the earth mover's distance (EMD) [19] between the color signatures of every pair of colors i and j which minimizes $\sum_{i\in S_1}\sum_{j\in S_2} d_{ij}f_{ij}$, where f_{ij} indicates the flow between color i and j subject to all the constraints that move all the mass from S_1 to S_2 [19]. The resulting EMD can be represented as a function $f(\theta)$ ($0^\circ \leq \theta \leq 180^\circ$) that finds the orientation of a diameter to maximize the difference between two half discs, i.e., $\hat{\theta} = \arg\max_\theta f(\theta)$, and used as the compass operator (CO) feature.

Multi-Scale Representation. We extract the above-mentioned features on every point of an image and integrate them to detect contours. In order to deal with the scale-space problem [12]. we obtain local features at different scales by changing the standard deviation of image gradient and texture inhibition, direction of brightness and color gradient, and the standard deviation of a Gaussian derivative in the compass operator. These features provide rich descriptions of image details at different levels, thereby rendering a multi-scale representation. As edges can be extracted at different scales, several pixels on one edge have equally strong response and they should be considered to describe contours. Thus, we extract local features at three different scales.

2.2 Multi-Scale Global Features

It has been shown that object contours can be better extracted by incorporating global information (e.g., the gPb method [3, 13]) than simply local visual cues (e.g., the pb algorithm [14]). However, existing methods that exploit global information (e.g., gPb) are often time consuming. For efficiency and effectiveness, we incorporate global visual saliency [7] in our approach. Cheng et al. [7] present a simple and efficient saliency extraction algorithm based on region contrast which exploits histogram contrast and spatial information. Each image is fist segmented into regions and the saliency value is computed by measuring its color contrast to all other regions in the image: $S(r_k) = \sum_{r_k \neq r_i} w(r_i) D_r(r_k, r_i)$, where $w(r_i)$ is the weight of region r_i and $D_r(\cdot,\cdot)$ is the color distance between the two regions. The weighting term can increase the effects of closer regions and decrease those farther regions. With this method, the distinctness of each pixel is described in a saliency map S.

Given a pixel $I(x,y)$, we consider the local contrast of the saliency values with respect to its four neighbors. We take the maximum value of the difference between saliency values of its neighbors.With this saliency contrast (SC) feature, the difference of saliency values is maximized when the pixel is right on the contour, thereby facilitating boundary detection.

3 Segment-Level Features

While the pixel-level features described in Section 2 can be utilized to determine contour points, structural cues such as segments contain important information more than pixelwise evidence. Toward this goal, we compute superpixels to extract structural segments with the SLIC algorithm [1], which has been verified to perform well terms of efficiency and effectiveness. Point features described in Section 2 are used to describe edge pixels on the line fragment, and segment-level features are then extracted by computing their mean value, variance and differences in this work.

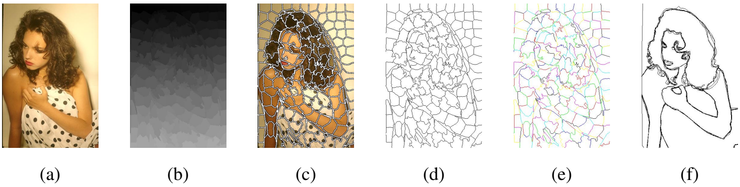

(a) (b) (c) (d) (e) (f)

Figure 1: With 200 superpixels: (a) input. (b) superpixel. (c) edges in source image. (d) all edges. (e) segment. (f) ground truth.

In this paper, we vary the number of superpixels (from 200 to 2000) to extract segments at different scales. Figure 1 shows one example how segments are extracted when 200 superpixels are used. From the superpixel results, edges can be extracted (Figure 1(c)-(d)) based on cluster value of each point (Figure 1(b)) with respect to its neighborhood. When the pixels within a neighborhood of a point belong to more than two clusters, it indicates the existence of a endpoint (e.g., the point on the T-junctions or Y-junctions of Figure 1(b)). On the other hand, when the pixels within a neighborhood of a point belong to exactly two clusters, it indicates the existence of a segment point. Thus, segments and endpoints can be extracted as denoted by different colors in Figure 1(e) for contour extraction. We determine whether a pixel belongs to a segment or not and then concatenate segment-level features and the pixel level features.

At each point, as described in Section 2, 18 (6 features at 3 scales) local and 1 global pixel-level features are extracted. We compute the mean, variance, minimum and maximum values of 19 features from all the points on a segment. In addition, we compute 4 local statistics (mean, variance, minimum and maximum values) in the neighborhood of the corresponding segment and obtain a 76-dimensional feature. Figure 1(f) shows the ground truth of contour for comparison with the segments extracted from our method. One advantage of our approach is that edge thinning is not necessary, and instead we directly operate on pixels to extract segments. By controlling the number of generated superpixels, segment-level features at different scales can be obtained, and the smallest segment is a pixel itself.

4 Contour Detection via Random Forest

To extract discriminative information from features, we propose a mapping method based on the posterior divergence (PD) measure over Gaussian mixture model (GMM).

The posterior divergence approach is a generative discriminative scheme that determines one or a few samples to update the model in every iteration of the EM step which has been shown to be effective in several tasks [10]. We transform the vectors formed by point-level and segment-level features based this mapping to obtain more discriminative information for contour detection using a random forest classifier. While our method bears some similarity to [10], the generative models and derivations for feature mappings are different.

Posterior Divergence of Gaussian Mixture. Let $\mathbf{x} \in \mathrm{R}^D$ be the observed random variable. In the context of contour detection, $\mathbf{x}$ denotes the combination of multi-scale features. Let $\mathbf{z} = \{z_1, \cdots, z_K\}$ be the hidden variable, where $z_k = 1$ if the k-th mixture center is selected to generate a sample and $z_k = 0$ otherwise. The joint distribution of Gaussian mixture models can be expressed as $P(\mathbf{x}, \mathbf{z} \,|\, \theta) = \prod_{k=1}^{K} N(\mathbf{x}; \mathbf{u}_k, \Sigma_k)^{z_k} \prod_{k=1}^{K} a_k^{z_k}$, where $\mathbf{a} = (a_1, \cdots, a_K)^\top$ are the mixture prior satisfying $a_k = \mathrm{E}_{P(\mathbf{z})}[z_k]$; $\mathbf{u}_k$ and Σ_k respectively are the mean and variance matrix of the k-th mixture center.

For any observed sample $\mathbf{x}^t$, similar to [9], we assume that the posterior distribution of $\mathbf{z}$ takes the same from with its prior $P(\mathbf{z})$ but with different parameter $\mathbf{g}^t = (g_1^t, \cdots, g_K^t)^\top$, $Q^t(\mathbf{z}) = \prod_{k=1}^{K} g_k^{t\, z_k}$. With the above joint distribution and approximate posterior distribution, the free energy function $\mathcal{F}$ of the sample $\mathbf{x}^t$ can be formulated with variational inference [9],

$$\mathcal{F}(Q^t, \theta) = \mathrm{E}_{Q^t(\mathbf{z})}[\sum_{k=1}^{K} z_k (\sum_{d=1}^{D} -\frac{(x_d^t - u_d)^2}{2\delta_d^2} - \log\sqrt{2\pi} \prod_{d=1}^{D} \delta_d^{D/2}) + \sum_{k=1}^{K} z_k \log \frac{g_k^t}{a_k}]. \quad (2)$$

Let θ be the model estimated from a set of $N-1$ training samples $\mathcal{X} = \{\mathbf{x}^i\}_{i=1}^{N-1}$, and θ_{+t} be the model estimated from a set of N samples $\mathcal{X} \cup \{\mathbf{x}^t\}$. The log likelihood of the EM algorithm for the input sample $\mathbf{x}^t$ is

$$\begin{aligned} \mathcal{L}^t &= \sum_{i=1}^{N}[-\mathcal{F}(Q_{+t}^i, \theta_{+t})] - \sum_{i \neq t}[-\mathcal{F}(Q^i, \theta)] \\ &= \sum_{i=1}^{N}(-E_{Q_{+t}^i(\mathbf{z})}[\log \frac{Q_{+t}^i(\mathbf{z})}{P(\mathbf{x}^t|\mathbf{z}, \theta_{+t}) P(\mathbf{z}|\theta_{+t})}]) - \sum_{i \neq t}^{N}(-E_{Q^i(\mathbf{z})}[\log \frac{Q^i(\mathbf{z})}{P(\mathbf{x}^t|\mathbf{z}, \theta) P(\mathbf{z}|\theta)}]). \quad (3) \end{aligned}$$

Similar to [10], we assume that $Q_{+t}^i(\mathbf{z}) = Q^i(\mathbf{z})$, and thus have

$$\begin{aligned} \mathcal{L}^t = & [\underbrace{\sum_{i \neq t}^{N} E_{Q^i(\mathbf{z})} \log \frac{P(\mathbf{x}^t|\mathbf{z}, \theta_{+t})}{P(\mathbf{x}^t|\mathbf{z}, \theta)}}_{\Phi_x^{pd}} + \underbrace{E_{Q^t(\mathbf{z})} \log P(\mathbf{z}|\theta_{+t})}_{\Phi_x^{fit}}] \\ & + [\underbrace{\sum_{i \neq t}^{N} E_{Q^i(\mathbf{z})} \log \frac{P(\mathbf{z}|\theta_{+t})}{P(\mathbf{z}|\theta)}}_{\Phi_z^{pd}} + \underbrace{E_{Q^t(\mathbf{z})} \log P(\mathbf{z}|\theta_{+t})}_{\Phi_{\mathbf{z}}^{fit}} - \underbrace{E_{Q^t(\mathbf{z})} \log Q^t(\mathbf{z})}_{\Phi_z^{ent}}], \quad (4) \end{aligned}$$

where the posterior divergence Φ^{pd} measures how much $\mathbf{x}$ affects the model, the fitness function Φ^{fit} measures how well the sample fits the model, the entropy function Φ^{ent} measures how uncertain the fitting is. The feature mapping given by posterior divergence are derived as follows,

$$\Phi_x^{pd} = \sum_{i \neq t}^{N} \sum_{k=1}^{K} \sum_{d=1}^{D} g_k^i \left(-\frac{(x_d^t - u_{d,+t})^2}{2\delta_{d,+t}^2} + \frac{(x_d^t - u_d)^2}{2\delta_d^2} + \delta_{d,+t}^{D/2} - \delta_d^{D/2}\right) = \sum_{d=1}^{D} \Phi_{x_d}^{pd}, \tag{5}$$

where Φ_x^{pd} is further decomposed to D terms according to the dimension of $\mathbf{x}$, and $\Phi_{x_d}^{pd}$ measures how x_d affects the model. Similarly, we have,

$$\Phi_x^{fit} = \sum_{k,d=1}^{K,D} g_k^t \left(-\frac{(x_d^t - u_{d,+t})^2}{2\delta_{d,+t}^2} + \delta_{d,+t}^{D/2} \log\sqrt{2\pi}\right) = \sum_{d=1}^{D} \Phi_{x_d}^{fit}, \tag{6}$$

where $\Phi_{x_d}^{fit}$ measures how well x_d fits the model. The feature mapping according to the hidden variable $\mathbf{z}$ can be derived as follows

$$\Phi_z^{pd} = \sum_{i \neq t}^{N} \sum_{k=1}^{K} g_k^i \log \frac{a_{k,+t}}{a_k} = \sum_{k=1}^{K} \Phi_{z_k}^{pd}, \tag{7}$$

where $\Phi_{z_k}^{pd} = \sum_{i \neq t}^{N} g_k^i \log \frac{a_{k,+t}}{a_k}$, and,

$$\Phi_z^{fit} = \sum_{k=1}^{K} g_k^t \log a_{k,+t} = \sum_{k=1}^{K} \Phi_{z_k}^{fit}, \quad \Phi_z^{ent} = \sum_{k=1}^{K} g_k^t \log g_k^t = \sum_{k=1}^{K} \Phi_{z_k}^{ent}. \tag{8}$$

Therefore for input $\mathbf{x}^t$, we obtain a set of feature mapping:

$$\Phi^t = \mathrm{vec}(\{\Phi_{x_d}^{pd}, \Phi_{x_d}^{fit}, \Phi_{z_k}^{pd}, \Phi_{z_k}^{fit}, \Phi_{z_k}^{ent}\}_{d,k}). \tag{9}$$

Feature Mapping via Posterior Divergence. To extract discriminative information from multi-scale features, we map multi-scale features to the space via Equation 9 instead of simply stacking features in a long vector. The reasons for using this mapping are twofold. First, this feature mapping includes a data normalization procedure which reduces the metric difference among different features. The normalization is carried out by $(x_d^t - u_d)^2/(2\delta_d^2)$, with which the derived feature mapping only responses to the relative quantities with respect to the mean and variance. Second, this feature mapping exploits the hidden variable $\mathbf{z}$ which encodes additional information, i.e., cluster or mixture center which is informative in image representation (e.g., bag-of-words).

Random Forest Classifier. With the features mapped to the space induced in Section 4, various classifiers within the random forest [5] learning framework can be constructed based on different representations. A random forest is an ensemble classifier consisting of numerous decision trees where the class label is determined based on the mode of the outputs by individual trees. Random forest algorithms have been shown to deal with large amount of data points effectively and efficiently. In a randomized decision forest, the learning examples are split into two descendant trees. We use the Gini ratio [17] to split the training examples so that the descendant trees are "purer" than their parents. Aside from combining random forest classifiers, better results can be achieved by considering the edges extracted from superpixels for contour completion based on the Gestalt theory. Toward this, we compute the overlap percentage of detected contours and the extracted edges from superpixels. If the percentage exceeds some threshold, the incomplete edges are replaced by the entire ones extracted from superpixels.

Table 1: F-measures of contour detection on BSDS500.(a) different parts. (b) different contour detection algorithms.

BSDS500			
	ODS	OIS	AP
MG+DG+IT+CG+BG+CO+Mulit+GMM-PD	0.72	0.74	0.78
MG+DG+IT+CG+BG+CO+Mulit	0.70	0.72	0.75
MG+DG+IT+CG+BG+CO	0.69	0.71	0.72
MG+DG+IT+CG+BG	0.68	0.70	0.71
MG+DG+IT+CG	0.67	0.69	0.70
MG+DG+IT	0.65	0.67	0.68

(a)

BSDS500			
	ODS	OIS	AP
human	0.80	0.80	-
MCDRF	0.72	0.74	0.78
gPb [3]	0.71	0.74	0.65
BEL [8]	0.66	0.67	0.68
Canny [6]	0.60	0.63	0.58
compass operator [20]	0.49	0.53	0.36

(b)

5 Experiments

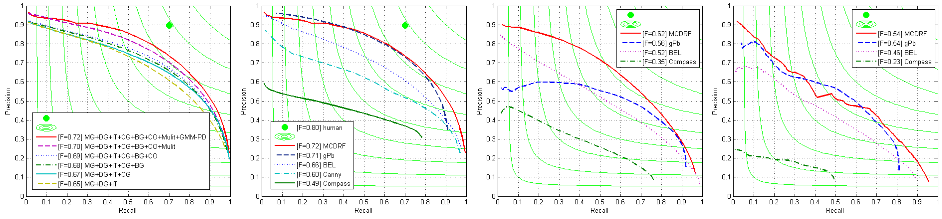
We evaluate the proposed algorithm for contour detection via a multi-scale random forest of 50 trees (referred as MCDRF) on several data sets. The Berkeley segmentation data set [3] includes 500 images of 481×321 pixels and human labeled segmentation results. For fair comparisons, we use 300 images for training and the remaining ones for tests [3]. The pixels on the ground-truth contours of the training set are used as positive examples, whereas other pixels are utilized as negative examples. All experiments are performed on a machine with 3.10 GHz CPU and 8 GB memory. More results can be found in the supplementary document.

Feature Combination. We first evaluate different feature combination of the proposed algorithm for contour detection. Table 1(a) and Figure 2(a) show the F-measures of contour detection results using different feature combinations, where the F-measure is computed by $\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$. The results show that contours can be better detected by adding more features.

Evaluation of Contour Detection Algorithms. We evaluate the proposed MCDRF algorithm against other methods including the compass filter [20], Canny edge detector [6], gPb [3, 13], and BEL [8] methods using the BSDS500 data set. Figure 2(b) shows the precision-recall curves with respect to human labeled ground truth. We use a training set to determine the optimal data set (ODS) scale and fix it for all the test images [3]. We also evaluate the performance the optimal image scale (OIS) for each image, and the average precision (AP) on the full recall range [3]. Figure 2(b) shows the precision-recall curves and the F-measures with different thresholds. The proposed MCDRF algorithm achieves the highest F-measures and average precision among all methods. While the gPb method performs well in terms of accuracy, the computational load is significant. On average, the proposed MCDRF method is about 9 times faster than the gPb method (20 and 180 seconds on MATLAB respectively).

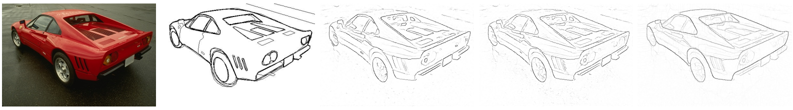
Figure 3 shows one contour detection results using features with and without using the proposed feature extraction method via posterior divergence. In addition, the results in Table 1(a) and Figure 2(a) show that contours can be better detected with the proposed feature extraction method.

Figure 4 shows the contour detection results by all the evaluated algorithms. We note that both the gPb and MCDRF methods are able to extract object contours with fewer spurious edges. In addition, our method capture more contours and details than the gPb method. Figure 4(a) shows the gPb misses some windows of the building and the contours in roof is not clear or sharp as those generated by the proposed method. The results of the Canny detectors without non-maximum suppression and no hysteresis are noisy (consistent with results shown in [3]) as the Gaussian kernel width is set based on ODS and OIS. The results of the compass filter are based on the code provided by [20] with default parameters.



(a) (b) (c) (d)

Figure 2: Precision-recall curves (a) BSDS500: different feature combinations. (b) BSDS500: different contour detection algorithms. (c) WHD data set. (d) WSD data set.



input Image ground truth only features using mutli-features MCDRF

Figure 3: Contour detection results with and without feature extraction.

We carry out experiments on the Weizmann horse database (WHD) [4] and the Weizmann segmentation database (WSD) [2]. The Weizmann horse database contains 328 side-view color images with manually segmented results. It contains horses of different breed, color, and size in various scenes. The Weizmann segmentation database contains 200 color images with manual segmentation results by several subjects. The images in this data set contain only one or two salient objects with relatively simple backgrounds. The foreground objects differ significantly from the background either by intensity, texture, or other low level cues.

We note that all the parameters of the proposed MCDRF algorithm are fixed in the experiments on three data sets. The proposed algorithm is evaluated against the gPb [3], compass operator [20] and BEL [8] methods. Although these databases are developed for segmentation evaluation, we extract contours using the same approach as the BSDS500 [3] to compute F-measures and precision-recall curves. Figure 2(c)-(d) shows the precision-recall curves of MCDRF are significantly better than those by the gPb, compass operator and BEL methods.

Table 2 shows the F-measure and average precision by the MCDRF algorithm are greater than those by the gPb [3], compass operator [20] and BEL [8] methods. The extracted contours show that the proposed MCDRF algorithm performs better than other methods with more details and less noise. Similar to the results with Weizmann horse data set, the extracted contours from this segmentation database show that the proposed MCDRF algorithm performs better than other methods with sharper details and fewer spurious edges.

6 Conclusion

In this paper, we propose an efficient and effective algorithm for contour detection based on a random forest classifier on features mapped from multi-scale local and global image features. The model parameters of the feature space are learned from the posterior divergence of the log likelihood of a Gaussian mixtures incrementally. We use posterior divergence to exploit more information rather than use raw features with random forest classifier directly. The

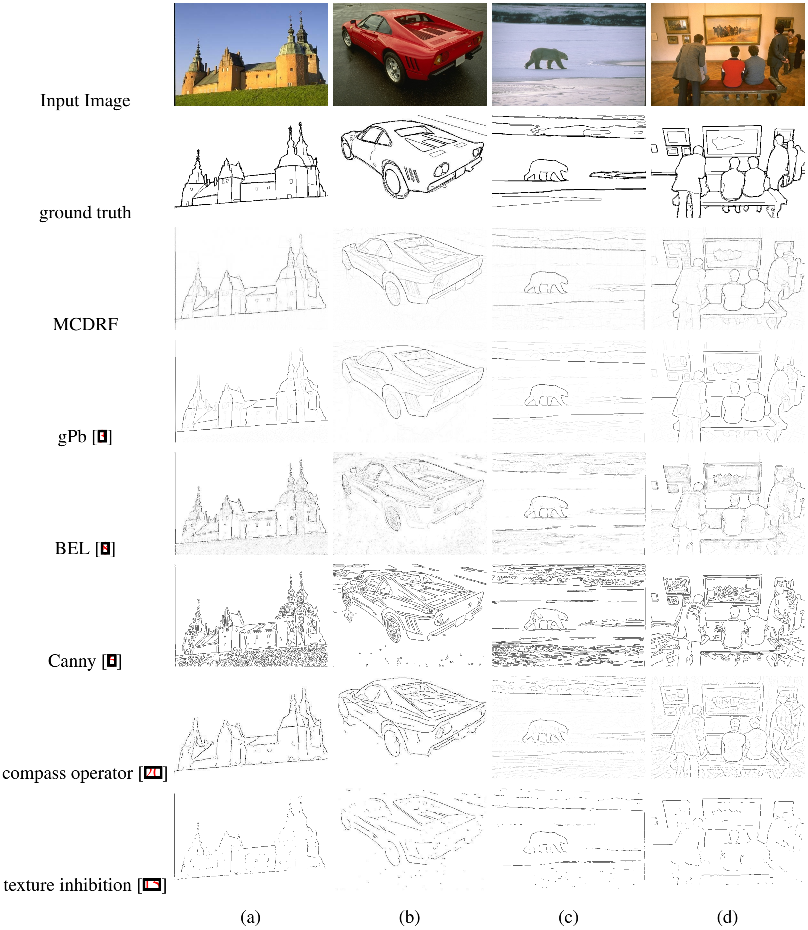


Figure 4: Sample experimental results from the Berkeley segmentation data set.

Table 2: F-measures of contour detectors in WSD and WHD.

Weizmann horse data set			
	ODS	OIS	AP
MCDRF	0.62	0.63	0.62
gPb [3]	0.56	0.58	0.47
BEL [8]	0.52	0.54	0.50
compass operator [20]	0.35	0.36	0.23

(a)

Weizmann segmentation data set			
	ODS	OIS	AP
MCDRF	0.54	0.59	0.51
gPb [3]	0.54	0.58	0.45
BEL [8]	0.46	0.46	0.39
compass operator [20]	0.23	0.25	0.09

(b)

proposed approach is evaluated qualitatively and quantitatively on three benchmark data sets against several state-of-art methods. Experimental results demonstrate the proposed algorithm performs favorably against leading methods for contour detection. Our future work includes more effective contour completion algorithms. In addition, we will develop efficient algorithms for object recognition based on contours.

Acknowledgements

C. Zhang, X. Li and Y. Zhao are supported in part by NSFC Grant 61175009. M.-H. Yang is supported in part by the NSF CAREER Grant 1149783 and NSF IIS Grant 1152576.

References

[1] Radhakrishna Achanta, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine SÍźsstrunk. SLIC Superpixels. Technical Report 149300, EPFL, 2010.

[2] Sharon Alpert, Meirav Galun, Achi Brandt, and Ronen Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 34(2):315–327, 2012.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[4] Eran Borenstein and Shimon Ullman. Learning to segment. In *Proceedings of European Conference on Computer Vision*, pages 315–328, 2004.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] John Canny. A computational approach to edge detection. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[7] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–416, 2011.

[8] Piotr Dollar, Zhuowen Tu, and Serge Belongie. Supervised learning of edges and object boundaries. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2006.

[9] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Sail. An introduction to variational methods for graphical models. In *Machine Learning*, volume 37, pages 183–233, 1999.

[10] Xiong Li, Tai Sing Lee, and Yuncai Liu. Hybrid generative-discriminative classification using posterior divergence. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2713–2720, 2011.

[11] Xiong Li, Bin Wang, Yuncai Liu, and Tai Sing Lee. Learning discriminative sufficient statistics score space for classification. In *Proceedings of European Conference on Machine Learning*, 2013.

[12] T. Lindeberg. *Scale-space theory in computer vision*. Springer, 1993.

[13] Michael Maire, Pablo Arbelaez, Charless C. Fowlkes, and Jitendra Malik. Using contours to detect and localize junctions in natural images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[14] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 26(1):530–549, 2004.

[15] Giuseppe Papari and Nicolai Petkov. An improved model for surround suppression by steerable filters and multilevel inhibition with application to contour detection. *Pattern Recognition*, 44(9):1999–2007, 2011.

[16] Giuseppe Papari and Nicolai Petkov. Edge and line oriented contour detection: State of the art. *Image and Vision Computing*, 29(2-3):79–103, 2011.

[17] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the Gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[18] Xiaofeng Ren. Multi-scale improves boundary detection in natural images. In *Proceedings of European Conference on Computer Vision*, pages 533–545, 2008.

[19] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of IEEE International Conference on Computer Vision*, pages 59 –66, 1998.

[20] Mark A. Ruzon and Carlo Tomasi. Color edge detection with the compass operator. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2166, 1999.

[21] Andrew Stein, Derek Hoiem, Andrew Stein, Derek Hoiem, and Martial Hebert. Learning to find object boundaries using motion cues. In *Proceedings of IEEE International Conference on Computer Vision*, 2007.