Boundary extraction using supervised edgelet classification

Ji Zhao
Jiayi Ma
Jinwen Tian
Jie Ma
Sheng Zheng
Boundary extraction using supervised edgelet classification

Ji Zhao
Jiayi Ma
Jinwen Tian
Jie Ma
Huazhong University of Science and Technology
Institute for Pattern Recognition and Artificial Intelligence
Wuhan 430074, China
E-mail: zhaoji84@gmail.com

Sheng Zheng
China Three Gorges University
Institute of Intelligent Vision and Image Information, College of Science
Yichang 443002, Hubei, China

Abstract. Traditional learning-based boundary extraction algorithms classify each pixel edge separately and then get boundaries from the local decisions of a classifier. However, we propose a supervised learning method for boundary extraction by using edgelets as boundary elements. First, we extract edgelets by clustering probabilities of boundary. Second, we use features of edgelets to train a classifier that determines whether an edgelet belongs to a boundary. The classifier is trained by utilizing edgelet features, including local appearance, multiscale features, and global scene features such as saliency maps. Finally, we use the classifier to decide the probability that the edgelet belongs to the boundary. The experimental results in the Berkeley Segmentation Dataset demonstrate that our algorithm can improve the performance of boundary extraction. 

Subject terms: boundary extraction; edgelet; supervised learning; boosted decision tree; logistic regression.

Paper 111008 received Aug. 23, 2011; revised manuscript received Oct. 31, 2011; accepted for publication Nov. 11, 2011; published online Feb. 6, 2012.

1 Introduction

Boundary extraction has been investigated in computer vision for decades; however, it is still an open problem. This could be attributed to the fact that the origin of an edge is very complex: it can be caused by surface normal discontinuity, depth discontinuity, surface color discontinuity, or illumination discontinuity. In early days, boundary extraction algorithms used gradient-based methods, modeling edges as sharp changes in image intensity. Such algorithms used a local template to filter the image, along with pixels that had strong responses corresponding to boundaries. Edges calculated by this kind of modeling were not guaranteed to correspond with real boundaries, since the ideal boundary should correspond to sharp changes of geometric structure, object occluding, and the like.

In recent decades, many elegant technologies have been developed to preserve true edges and suppress false edges. Meanwhile, as the machine learning technologies emerge, good performance of boundary extraction can be achieved through carefully selected features and well-trained classifiers. In order to make the extracted boundary correspond with the perception of human beings, a large-scale image database with human labeling as ground truth was collected for training classifiers. Such a database, the Berkeley Segmentation Dataset, provides a benchmark for quantitative evaluation for image segmentation and boundary extraction.

Learning-based boundary extraction is very flexible because it can easily integrate geometric and semantic cues, occluding cues, and 3-D cues such as geometric context and surface layout. Global information can also be integrated into boundary extraction. Based on large-scale datasets, better detectors can be constructed by combining multiple low-level cues such as brightness, color, and texture. In order to take advantage of global information, some researchers have used multiscale features, and local appearance to build up global affinity matrices and have performed spectral clustering to get global salient boundaries.

In this paper, we present a new method for learning boundaries. We adopt a hierarchy clustering method to extract edgelets, and train a classifier to determine whether an edgelet belongs to a boundary. This could be a promising approach because it could allow the use of an array of interesting features to classify edges compared with the traditional methods that work on only separate pixels and use the local contrast. The main contributions of our work include (a) our use of edgelets as boundary grouping units and (b) our proposed method of an edgelet extraction method with traditional probability-of-boundary (PB) images as input. These are in contrast to traditional boundary extraction methods that classify pixels separately. Then, (c) a learning-based method determines whether an edgelet belongs to a true boundary by using multilevel features to improve boundary extraction performance. We believe that edgelet representation can make the use of high-level information more convenient and the higher-level grouping more efficient.

1.1 Related Work

Martin et al. determined edges and boundaries by using local appearance, including color, oriented-energy, and texture gradients, combined with supervised learning. Maire et al. improved on these results by combining the local appearance–based detector with a spectral detector. In more recent work, Arbelaez and Arbelaez et al. proposed an ultrametric contour map and used a variant of the watershed algorithm to set up a relation between boundary and segmentation. This method can convert an extracted boundary to segmentation; as a result, a closed boundary...
can be achieved. Learning-based boundary extraction methods also include the work of Dollar et al. and Ren.

Boundary extraction and image segmentation present a duality. On the one hand, getting boundaries from segmentation is straightforward, on the other, hand, boundaries provide much useful information for segmentation. Some popular segmentation algorithms include normalized cuts (Ncuts), mean shift, and efficient graph-based segmentation. Malik et al. integrated contour and texture to achieve good segmentation in the framework of Ncuts. Alpert et al. proposed an algorithm by using low-level cue integration. An evaluation of the performance of several popular segmentation algorithms can be found in the study by Arbelaez et al., which demonstrates that boundary extraction algorithms usually perform better than segmentation algorithms in protecting boundaries. These authors invented a watershed segmentation to set up relations between closed boundaries and segmentation.

Multiscale technology can be used for boundary extraction and in many other areas—for instance, as a key element in feature extraction and object detection. Felzenszwalb et al. used a multiscale histogram of oriented gradients (HOG) to construct a feature pyramid and combined the HOG with a deformable module to construct a state-of-the-art object recognition algorithm. In the texton calculation procedure, the well-designed filter bank has already considered the scale factor. In the process of boundary extraction, scale selection also plays an important role. Large-scale edges are robust to noise but difficult to localize. Small-scale edges conserve more edge details but are sensitive to noise. Therefore, color and texture gradients are usually calculated in a fixed number of scale and then are integrated.

Contour completion is another issue highly related to boundary extraction. It usually makes use of gestalt law and sets up an energy function. Then a multiscale conditional random field is used to complete the contour. In contour analysis for image sequence, Liu et al. classified boundaries into three levels: edgelet, boundary fragment, and contour. For textureless image sequences, they used smoothness, contrast consistency, and the temporal-spatial relation of images as soft constraints to group boundaries into contour. As a result, they could achieve interpretation of boundaries as consistently as possible.

The rest of our paper is organized as follows. In section 2, we give an edgelet extraction algorithm. In section 3, we describe the features used for edgelet classification. In section 4, we design the classifier. The experimental results are presented in section 5. We conclude in section 6.

2 Edgelet Extraction

There are many edgelet methods for textureless images. Liu et al. used steerable filters to analyze oriented energy and traced boundaries in a manner similar to that of the Canny edge detector. For images with strong textures, it is necessary to consider texture cues. We begin our research on the basis of PB and adopt edgelets for boundary representation instead of pixels. So we need to extract reliable edgelets that will support the following research.

This paper adopts a data clustering method to segment PB images into edgelets. The data clustering method is also used in a form of image segmentation known as segmentation by weighted aggregation (SWA), which is a variant of the NCuts algorithm and provides a framework for multiple cue integration. It is very efficient because it has linear computational complexity.

The algorithm includes a coarsening process and a sharpening process. The coarsening process in a hierarchical fashion is inspired by algebraic multigrid techniques. The procedure of coarsening is as follows:

1. Regard the pixels with nonzero PB values as nodes of an undirected weighted graph, and then construct the affinity matrix of the graph. Denote the weighted graph as \( G = (V, E) \), where \( V = \{ i \mid P_i > 0 \} \) and \( E = \{ e_{ij} \mid i \in V, j \in V \} \) are the set of nodes and the set of edges, respectively. \( P_i \) is the PB value of pixel \( i \), and edge \( e_{ij} \) is the affinity between neighboring nodes \( v_i \) and \( v_j \). Since the PB images are thin edges, we choose eight connected pixels as neighbors. The affinity \( e_{ij} \) between node \( i \) and node \( j \) is determined by their PB contrast and distance:

\[
e_{ij} = \begin{cases} e^{-\alpha||P_i - P_j||} & \text{if } ||l_i - l_j|| \leq \sqrt{2} \\ 0 & \text{otherwise} \end{cases}
\]

where \( P_i \) is the PB value of node \( i \); \( l_i \) is the location of node \( i \); and \( \alpha \) is a constant that takes the value of 10 in our work.

2. Select the most representative nodes as seeds. The seed set is a subset \( C \subseteq V \) whose elements are strongly coupled to all of the original nodes except for the seeds. Seeds are used to define the equivalent problem at a coarser scale. The seed \( C \) set is defined as:

\[
\min \frac{|C|}{n} \quad \text{s.t.} \quad \forall i \notin C: \sum_{j \in C} e_{ij} \geq \phi,
\]

where \( |\cdot| \) is the cardinality of the set, and the parameter \( \phi \) is set to 0.2. We adopt a greedy algorithm to select the seeds.

3. Calculate the interpolation matrix. Once a set of coarse variables has been selected, an interlevel interpolation matrix is used to define the equivalent problem at a coarser level.

4. Compute the saliency of each seed and reweight the affinity matrix of the selected seeds. The newly selected seeds and their affinity matrix is used to construct a graph at a coarser level.

5. Evaluate the saliency of each node and determine whether the aggregation has finished. If not, return to step 1. In the saliency evaluation step, use the average PB value and curvature as cues.

The sharpening procedure is the inverse procedure of the coarsening procedure. The sharpening procedure includes the following steps:

1. Using nodes and their affinity matrix at the coarser level, calculate the affinity matrix at a finer level.

2. Determine whether we have arrived at the finest level.

If not, repeat step 1. For further details about SWA clustering, see Sharon et al.
Considering that we use features of edgelets to perform classification, we need to split the T-junction to ensure that every edgelet has only two ends. We simply eliminate the junction points to ensure that the extracted edgelets have no T-junction. Figure 1 demonstrates a typical PB image and extracted edgelet by our method. Usually a typical image has 500 to 1000 edgelets (for $321 \times 481$ resolution images).

### 3 Features for Edgelet Classification

Cue integration at low, middle, and high levels is important for boundary extraction and image segmentation. Low-level cues include color, brightness, and texture gradients. Good performance can be obtained only by integrating them properly.\(^5,20\)

In computer vision, multiscaling is an issue that always needs to be considered. Edges, like any other image feature are multiscale in nature. As a result, multiscale features has been considered in boundary extraction\(^6\)—for example, Ren et al.\(^17\) used disks of different diameters to aggregate multiscale color and texture gradients. In this paper, we also consider multiscale gradients. We run the PB operator at three scales, an octave apart, starting at the default PB scale. For color gradient and texture gradient, we calculate four features at each scale: the mean of the gradient, the variance of the gradient, the 20th percentile of the gradient, and the 80th percentile of the gradient. As a result, the dimensionalities of the color gradient and the texture gradient.

#### Table 1 Features related to a pixel.

<table>
<thead>
<tr>
<th>Features and their description</th>
<th>Dimensionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color gradient: 3 color channels</td>
<td>3</td>
</tr>
<tr>
<td>Texture gradient</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>4</td>
</tr>
</tbody>
</table>

#### Table 2 Features related to an edgelet.

<table>
<thead>
<tr>
<th>Features and their description</th>
<th>Dimensionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color gradient</td>
<td>36</td>
</tr>
<tr>
<td>Texture gradient</td>
<td>12</td>
</tr>
<tr>
<td>PB of edgelet: mean, variance, 20th percentile, 80th percentile</td>
<td>4</td>
</tr>
<tr>
<td>Length of edgelet</td>
<td>1</td>
</tr>
<tr>
<td>Curvature of edgelet</td>
<td>1</td>
</tr>
<tr>
<td>Saliency of edgelet</td>
<td>1</td>
</tr>
<tr>
<td>Location: mean of normalized x and y</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>57</td>
</tr>
</tbody>
</table>
are 36 (three color channels, three scales, and four features per scale) and 12 (three scales and four features per scale), respectively. Table 1 lists the pixel features used for calculating PB. Table 2 lists the edgelet features used by our method.

Although boundary extraction can be implemented using only low-level cues, we believe that it is beneficial to use high-level features. For example, Oliva and Torralba developed a scene context known as Gist, a feature that provides a low-dimensional representation for the scene. Scene features are useful for boundary extraction. For example, if scene features indicate that the image is most likely a natural scene, then the original PB image may have plenty of false edgelets produced by texture. On the other hand, if the scene features indicate that the image is probably an artificial scene, then the image may have many long straight lines with high contrast. Another example of global information of the image is a saliency map. In the past decade, many computational models on visual attention have been developed many of which analyze the images and output saliency maps that indicate the probability of eye fixations. Figure 2 demonstrates two examples of saliency maps. We adopt saliency maps as global scene features.

Besides the above features, we use many heuristic cues borrowed from occluding analysis. Edgelet location also provides strong edge cues because people often place the object(s) of interest in the center of photos. We normalize the pixel locations by width and height of the image and compute the mean of the vertical and horizontal of an edgelet in the image. Additionally, curvature helps distinguish the object from the background because convexity cues are useful for distinguishing foreground and background. Using all of these cues, we learn to determine whether an edgelet is likely to belong to the boundary. Following the methodology of Hoiem et al. we compute all cues that might be useful and use a classifier to decide which to use and how to use them.

4 Classifier

We consider two kinds of classifier: a boosted decision tree and a logistic regression. The former is a nonparametric classifier and the latter is a parametric classifier.

For a boosted decision tree, the classifier outputs the confidence of an edgelet belonging to the boundary. Given the

Fig. 2 Saliency map as a global feature. This figure demonstrates two examples of a saliency map. (a) and (c) are original images, and (b) and (d) are their corresponding saliency maps. The intensity of the image corresponds to saliency.

Fig. 3 Precision recall curve. Our F-measure is 0.66.
confidence, we can calculate the probability of the edgelet belonging to the boundary:

\[ P_i^{\prime} = \frac{1}{1 + e^{(A \cdot c_i + B)}}, \]  

(3)

where \( c_i \) is the confidence of edgelet \( i \) belonging to the boundary, and \( P_i^{\prime} \) is the probability of edgelet \( i \) belonging to the boundary. Constants \( A \) and \( B \) are parameters that can be estimated by maximum likelihood estimation.

In our implementation, each strong classifier consists of 20 decision trees, each of which has eight leaf nodes. Decision trees are learned using a weighted version of the Matlab treefit function. We first grow the tree to two times the number of nodes desired and then prune it using the Matlab treeprune function, as done by Hoiem et al.\(^2\)

The posteriori probability of a pixel belonging to a boundary can be computed as follows:

\[ P(\text{pixel}|\text{edgelet}) = \frac{P(\text{edgelet}|\text{pixel}) \cdot P(\text{pixel})}{P(\text{edgelet})}, \]  

(4)

where \( P(\text{pixel}|\text{edgelet}) \) is the posteriori probability of a pixel belonging to the boundary after our edgelet classification; \( P(\text{edgelet}|\text{pixel}) \) is the output probability of the edgelet classifier in Eq. (3); \( P(\text{pixel}) \) is the PB value of a pixel; and \( P(\text{edgelet}) \) is the prior of an edgelet, estimated simply by computing the proportion of the edgelet in training images.

Martin et al.\(^6\) used a variety of classifiers and found that logistic regression has the best speed-space-accuracy tradeoff. We also tested logistic regression as a classifier. A logistic classifier is simpler than many other classifiers.
and is efficient to train and test. Our experimental results described below are consistent with those of other researchers.6,7.

In order to train the classifiers, we need to generate positive and negative samples. We define an edgelet as belonging to the boundary if at least 20% of the edgelet pixels correspond to the ground-truth boundary. We selected 5200 edgelets as positive samples and 2800 edgelets as negative samples.

5 Experimental Results

We quantitatively evaluated our method in terms of the precision-recall curve and F-measure in the Berkeley Segmentation Dataset. We used 200 human-labeled images for training and then ran tests on a different set of 100 images.

Use of a precision-recall curve is a standard technique in information retrieval. It was introduced to boundary detection by Martin et al.6 A particular application will define a relative cost α between precision and recall quantities, which focuses attention at a specific point on the precision-recall curve. The F-measure, defined as $F = P \cdot R / (\alpha R + (1 - \alpha)P)$, captures this tradeoff, where $P$ and $R$ are precision and recall, respectively. In this work, parameter α is set to 0.5, as in previous works.6

Figure 3 shows the precision-recall curve for our boundary extraction method. Its performance is better than that of Martin et al.6 at low recall. Its F-measure is higher than that of the PB algorithm. Figure 4 shows a group of boundary extraction results through our method. Compared with the method of Martin, Fowlkes, and Malik, the true boundaries have been enhanced, at the cost of increasing only slightly more false boundaries.

The confusion matrix for the edgelet classification is as follows. Tables 3 and 4 give the confusion matrix of a boosted decision tree and a logistic regression, respectively. We can see that despite its simplicity the logistic regression performs better than the boosted decision tree.

Table 5 gives our experiment results for the selection of features. From this table, we can see that performance can benefit from more features.

Table 3 Confusion matrix for the edgelet classification using boosted decision tree.

<table>
<thead>
<tr>
<th></th>
<th>Edgelet</th>
<th>Non-edgelet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edgelet</td>
<td>0.67</td>
<td>0.33</td>
</tr>
<tr>
<td>Non-edgelet</td>
<td>0.10</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Table 4 Confusion matrix for the edgelet classification using logistic regression.

<table>
<thead>
<tr>
<th></th>
<th>Edgelet</th>
<th>Non-edgelet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edgelet</td>
<td>0.71</td>
<td>0.29</td>
</tr>
<tr>
<td>Non-edgelet</td>
<td>0.05</td>
<td>0.95</td>
</tr>
</tbody>
</table>

6 Conclusion

In this paper, we used edgelets as representation for boundary extraction. We used the output of edge detection as our starting point. We split the edge into edgelets according to their features, gathering the homogeneous edgelets. Then we trained a classifier to determine whether the edgelet belonged to a true boundary. Lastly, we demonstrated the performance of our method on a large-scale dataset.

Our future work will focus on improving boundary extraction performance. We believe that our edgelet extraction and classification method is a good tool for long line detection, occluding analysis, and edgelet-based optical flow estimation for natural images.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant nos. 61074516 and 60875009.

References


Ji Zhao received his BS degree in 2005 and his MS degree in 2008. Currently he is a PhD candidate at the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), China. His research area is computer vision.

Jiayi Ma received his BS degree in mathematics from HUST in 2008. Currently he is a PhD candidate at the Institute for Pattern Recognition and Artificial Intelligence, HUST, China. His research area is computer vision and machine learning.

Jinwen Tian received his PhD degree in pattern recognition and intelligent systems in 1998 from HUST, China. He is a professor and PhD supervisor of pattern recognition and artificial intelligence at HUST. His main research topics are remote sensing image analysis, wavelet analysis, image compression, computer vision, and fractal geometry.

Jie Ma received his PhD degree in pattern recognition and intelligent systems in 2004 from HUST, China. He is currently an associate professor at HUST. His main research topics are remote sensing image analysis, image compression, navigation, and object recognition.

Sheng Zheng received his MS degree from Huazhong Normal University, China, in 1992, and PhD degree from HUST in 2005. He is currently a professor in the College of Electrical and Information Engineering at China Three Gorges University, Yichang. His research interests include image processing, object recognition, computer vision, neural networks, and intelligent control.