

Contour Detection Refined by A Sparse Reconstruction-based Discrimination Method

Qi Wang & M. W. Spratling

Department of Informatics, King's College London, London, WC2R 2LS

qi.l.wang@kcl.ac.uk Michael.Spratling@kcl.ac.uk

Abstract Sparse representations have been widely used for many image processing tasks. In this paper, a sparse reconstruction-based discrimination (SRBD) method, which was previously proposed for the classification of image patches, is utilized to improve boundary detection in colour images. This method is applied to refining the results generated by three different algorithms: a biologically-inspired method, and two state-of-the-art algorithms for contour detection. All of the contour detection results are evaluated by the BSDS300 and BSDS500 benchmarks using the quantitative measures: F-score, ODS, OIS and AP. Evaluation results shows that the performance of each algorithm is improved using the proposed method of refinement with at least one of the quantitative measures increased by 0.01. In particular, even two state-of-the-art algorithms are slightly improved by applying the SRBD method to refine their contour detection results.

Keywords edge detection; contour detection; colour image segmentation; sparse coding; sparse representation

1. Introduction

Object contours play an important role in human visual perception such as scene understanding and object recognition [1]. Contour detection is also a fundamental technique for many applications in computer vision such as image segmentation [2], object recognition [3], robot vision [4] and medical image analysis [5]. It is widely accepted that the locations of object contours can typically be derived from discontinuities in the intensity, texture and colour of adjacent image regions. Therefore, there are numerous algorithms for contour detection which aim to locate these discontinuities. For instance, Canny [6] utilized first order derivative of Gaussian filters to locate intensity discontinuities. Similarly, other linear filters such as Sobel, Prewitt, Beaudet, Robert [7], Laplacian of Gaussian [8], higher order derivative of Gaussian and Gabor, have also

been devised. Algorithms that employ a linear filter can be interpreted as using a measure of the match between pixels and a given edge template to locate intensity discontinuities [9]. In contrast to utilizing an edge template (linear filter), other methods to locate intensity and colour discontinuities by comparing each pixel and its neighbouring pixels have been defined such as Mean shift [10], Bilateral filtering [11], Vector order statistics [12], and Morphological edge detector [13].

Because the algorithms mentioned above are not able to locate contours defined by texture discontinuities, several statistical approaches were devised for locating texture boundaries. For instance, Huang et al. [14] utilized two-sample statistical test of independence to compute the dissimilarity between adjacent image regions. Ojala et al. [15] employed distributions of local binary patterns and pattern contrast to evaluate the similarity of neighbouring image regions using G statistics to compare distributions. There are also many different types of algorithms which were developed for contour detections such as psychophysically inspired methods (e.g. Phase congruency [16] and Multiresolution analysis [17]), interactive detection methods (e.g. active contours [18]), graph-based methods (e.g. Normalized cuts [19]), and region-based methods (e.g. Watershed transform [20], Ultrametric contour map [21], Plaster cast model [22], Pixon-based method [23], and meta-heuristic method [24]).

Another set of algorithms attempted to simulate some operations performed by the human/animal visual system for contour detection. For instance, Spratling [25] applied a Predictive Coding/Biased Competition (PC/BC) model of the primary visual cortex (V1), which simulated neuron activity in the V1 [26], to locate contours defined by intensity discontinuities. Afterwards, Wang et al. [27] extended the PC/BC model of V1 to locate colour discontinuities, which was inspired by neurophysiological data from single neurons in Macaque V1 [28]. There are many other algorithms inspired by different kinds of operations in the human/animal visual system, including - but not limited to - the Eye

movement model [29], the Retina and V1 model [30], and the Surround suppression model [31].

One of the state-of-the-art contour detection algorithms for natural images is gPb-owt-ucm [2], which was derived from the integration of some previously proposed methods. Specifically, Martin et al. [32] developed the Probability of Boundary (Pb) algorithm first, which applied three different gradient methods to locate local discontinuities in brightness, texture and colour within an image. Secondly, Maire et al. [33] and Ren [34] found that the combination of these local discontinuities across different scales can improve the performance of the Pb algorithm. Thirdly, Maire et al. [33] also integrated these multi-scaled local discontinuities using the Normalized cuts [19] technique and proposed the globalization of probability of boundary (gPb) algorithm. Finally, Arbelaez et al. [2] post-processed the gPb algorithm by utilizing an improved Watershed transform algorithm [20] and an Ultrametric contour map [21], to generate the gPb-owt-ucm method. Song et al. [35] used a measure of Laplacian energy to remove redundant contours in gPb-owt-ucm [2].

Based on the hypothesis that natural images allow a sparse decomposition in some redundant basis (or dictionary), sparse representation techniques have been widely used for different image processing tasks, such as image denoising [36], image restoration [37], texture classification [38], and face recognition [39]. Sparse Coding has also been applied to contour detection. Sparse Code Gradients (SCG) [40] uses a learned sparse representation method [41] to generate a sparse code of each input image. It then applies the gradient method [32] to compute the gradients of the sparse code across multiple scales and multiple orientations. Finally, a support vector machine (SVM) is applied to locate contours within an image. SCG generates results that equal the state-of-the-art performance of gPb-owt-ucm.

Mairal et al. [42] proposed a sparse reconstruction-based discrimination (SRBD) method, which use two dictionaries to place image patches into two classes: edges and non-edges. The dictionaries were obtained by training with two different classes of image patches extracted from a set of training images. Afterwards, a test image patch was represented as a sparse code using these two dictionaries. Finally, the classification of the test image patch was determined by assigning it the category of the dictionary that produces the smallest reconstruction error. Mairal et al. [43] increased the discriminative power of the SRBD method by adding a discriminative training algorithm to the procedure of dictionary learning, and then applied the SRBD method to refining the result

derived from the Canny edge detector [6], which is significantly improved compared to the unrefined Canny detector.

In this paper, a slightly modified version of the SRBD method is applied to refining the contour detection results produced by three other, previously proposed, algorithms. These three algorithms are the Colour PC/BC model of V1 [27] (c-PC/BC), gPb-owt-ucm [2] and SCG [40]. The SRBD method is described in more detail in section 2. Section 3 describes the procedure used to apply the modified version of SRBD to refining the results derived from the three contour detectors, highlighting the modified steps compared to [43]. The Results section evaluates the performance of each of the three refined contour detection methods using the BSDS300 [44] and BSDS500 [2] datasets.

2. The Sparse Reconstruction-based Discrimination Method

The sparse representation technique can be summarized as: an input signal x admits a sparse approximation over a dictionary D comprised of many elements, when a linear combination of only a few elements from D can be found that approximates the original signal x .

In the SRBD method, input signals are $p \times p$ pixels patches extracted from images. These patches are converted to column vectors x with length $n=p^2$ pixels. The K-SVD algorithm [41] is applied to learning the dictionary D with a fixed number of k elements and a fixed sparsity factor L , using M patches (column vectors) extracted from training images. The problem of dictionary learning is to work out an optimal dictionary D which can minimize the reconstruction error given a fixed sparsity constraint, as described by the following equation:

$$\min_{\alpha, D} \sum_{l=1}^M \|x_l - D\alpha_l\|_2^2, s. t. \|\alpha_l\|_0 \leq L \quad (1)$$

where $\|\cdot\|_2$ represents the Frobenius l^2 -norm; x_l represents an input image patch converted to a column vector with the size n pixels; D represents the dictionary to be learnt with the size $n \times k$. The k -element column vector α_l is the sparse representation for the l -th patch using the dictionary D . $\|\cdot\|_0$ denotes the number of non-zero elements in a vector.

Afterwards, dictionary D learnt from equation 1 as well as patches (column vectors x) extracted from a test image become inputs to the OMP algorithm [45], which is applied to generate optimal sparse representations (α^*) of the test image patches using the same sparsity

factor L ; as described by the following equation 2:

$$\alpha^*(x, D) \equiv \arg \min_{\alpha} \|x - D\alpha\|_2^2$$

$$\text{s. t. } \|\alpha_l\|_0 \leq L \quad (2)$$

Afterwards, sparse reconstruction error $R^*(x, D, \alpha^*)$ can be straightforwardly obtained by encoding patches x extracted from a test image using the dictionary D learnt from training patches by equation 1 and sparse representations α^* computed from equation 2, as described by the following equation 3:

$$R^*(x, D, \alpha^*) \equiv \|x - D\alpha^*(x, D)\|_2^2 \quad (3)$$

Suppose that there are $N=2$ sets of image patches extracted from a set of training images, belonging to two different classes: edges and non-edges. The class of a training patch being determined by the class of its centre pixel. These two sets of patches can be used to define two dictionaries D by applying the K-SVD algorithm [41] as described in equation 1. Afterwards, two sets of optimal sparse parameters α^* can be determined for each test image patch using the OMP algorithm [45] as described in equation 2. Similarly, two sets of sparse reconstruction errors R^* can be obtained, as described in equation 3, by encoding the test patch using the two different dictionaries. Finally, the centre pixel of the patch from the test image can be assigned class i_0 (edge or non-edge) if the sparse reconstruction error R^* is smaller when encoded with dictionary D_i than with the other dictionary, as shown in the following equation 4:

$$i_0 = \arg \min_{i=1,2} R^*(x, D_i, \alpha_i^*) \quad (4)$$

3. Procedure of Refinement

Contour detection results derived from three previously proposed boundary-detection algorithms are refined by the application of a slightly modified version of the SRBD method. These three algorithms include a biologically-inspired model (c-PC/BC [27]), and two state-of-the-art algorithms (gPb-owt-ucm [2] and SCG [40]). Each step of the refinement is described in the following paragraphs. The steps in SRBD which are modified compared with [43] are highlighted.

Step 1: Two classes of patches are extracted from the set of training images provided with the BSDS300 [44] or BSDS500 [2] dataset. These two classes are true contour and false contour. Only if the centre pixel of a patch satisfies the following condition is the surrounding patch extracted. The centre pixel is identified as an ‘effective’ contour by the previously proposed boundary-detection algorithm that is being refined. Afterwards, the extracted patch x^T is classified as a true contour if the centre pixel of

the patch is selected as a contour pixel by at least one human subject [44]. In contrast, the extracted patch x^F is classified as a false contour if the centre pixel with its surrounding $r \times r$ pixels of the patch are all not selected as a contour pixel by all human subjects [44]. Because c-PC/BC, gPb-owt-ucm and SCG all produce a continuous value of probability of boundary (Pb) at each pixel, ‘possible’ contours are defined as those pixels with Pb values are larger than a threshold $T_1=0.01$. The value of the parameter T_1 was determined by trial and error. 1000 patches that satisfied these conditions were selected, at random, from each training image. As there are 200 training images this gave a total of 200000 training patches for each dictionary.

As previously mentioned, each image patch x^T or x^F belonging to the true or false contour with size $p \times p$ pixels are converted to a column vector with size $n=p^2$ pixels. If those training images are chromatic, image patches extracted from the red, green, and blue channel of the RGB colour space are concatenated to a single column vector with the size $3n$ pixels. It has been found that utilizing multiple sizes of patches and resolutions of images can improve the performance of the refinement [42]. Mairal et al. [43] used $p=5, 7, 9, 11, 15, 19, 23$ for both full scale and half scale images to refine the Canny edge detector. However, in this modified version, only some of these sizes of patches were used. $p=7, 9, 11, 15, 19$ pixels with $r=5$ pixels for full scale images and $p=7, 9, 11$ pixels with $r=3$ pixels for half scale images were used with the three algorithms. The selection of these particular sizes of patches to be extracted and the value of the parameter r were all determined by trial and error. As a result, there were overall $N=16$ (8 sizes \times 2 classes) sets of training patches.

Step 2: When SRBD was applied to refining the Canny edge detector by Mairal et al. [43], all dictionaries belonging to the true and false contour classes were learnt by using both the K-SVD algorithm [41] and the discriminative training method defined in [42]. The K-SVD algorithm, as described in equation 1, defines dictionaries that minimize the reconstruction error obtained when encoding true (or false) contour training patches using the true (or false) contour dictionary. At the same time, the discriminative training method was employed to increase the discriminative power of each pair of dictionaries by maximizing the reconstruction error obtained by encoding true (or false) contour training patches using the false (or true) contour dictionary. The dictionaries used in [43] had a fixed number of 256 elements. The number of iterations for the K-SVD algorithm

was set to 25 and a series of sparsity factors ranging from 1 to 15 were used.

In this modified version of SRBD, the procedure of dictionary learning was simpler. The $N/2=8$ sets of dictionary D^T and D^F belonging to the true and false contour classes were learnt using the K-SVD algorithm [41], as described in equation 1. The dictionaries D^T and D^F also had a fixed number of $k=256$ elements, the number of iterations for the K-SVD algorithm was set to 1000 and a fixed sparsity factor $L=12$ was used. The problem of dictionary learning is to work out an optimal dictionary D^T or D^F which can minimize the reconstruction error given a fixed sparsity constraint, as described by the following equation:

$$\min_{\alpha^T, D^T} \sum_{l=1}^{200000} \|x_l^T - D^T \alpha_l^T\|_2^2, \text{ s.t. } \|\alpha_l^T\|_0 \leq L \quad (5)$$

$$\min_{\alpha^F, D^F} \sum_{l=1}^{200000} \|x_l^F - D^F \alpha_l^F\|_2^2, \text{ s.t. } \|\alpha_l^F\|_0 \leq L \quad (6)$$

where all operators are identical to equation 1.

Step 3: Similar of the method of extracting patches from a set of training images described in step 1, $N/2=8$ sizes of patches (column vectors) were extracted from each test image at all locations where the centre pixel had been identified as an ‘effective’ contour by the previously proposed boundary-detection algorithm that was being refined. Afterwards, each of the $N/2=8$ sets of column vectors were encoding using the OMP algorithm [45] and each corresponding pair of dictionaries D^T and D^F learnt at the step 2 from equation 5 and 6 to generate $N=8$ sets of optimal sparse representations α^{T*} and α^{F*} as described by the following equation:

$$\alpha^{T*}(x, D^T) \equiv \arg \min_{\alpha} \|x - D^T \alpha\|_2^2, \quad (7)$$

$$\text{s.t. } \|\alpha_l\|_0 \leq L$$

$$\alpha^{F*}(x, D^F) \equiv \arg \min_{\alpha} \|x - D^F \alpha\|_2^2, \quad (8)$$

where all operators are identical to equation 2.

Afterwards, 8 sets of sparse reconstruction error $R^{T*}(x, D^T, \alpha^{T*})$ and $R^{F*}(x, D^F, \alpha^{F*})$ can be straightforwardly obtained by:

$$R^{T*}(x, D^T, \alpha^{T*}) \equiv \|x - D^T \alpha^{T*}(x, D^T)\|_2^2 \quad (9)$$

$$R^{F*}(x, D^F, \alpha^{F*}) \equiv \|x - D^F \alpha^{F*}(x, D^F)\|_2^2 \quad (10)$$

where $R^{T*}(x, D^T, \alpha^{T*})$ and $R^{F*}(x, D^F, \alpha^{F*})$ represents sparse reconstruction error obtained from computing the difference between the test image patch and the encoding of this test image patch using the dictionary D^T or D^F with its corresponding sparse representations $\alpha^{T*}(x, D^T)$ and $\alpha^{F*}(x, D^F)$ belonging to the true and false contour class.

Step 4: Each corresponding pair of sparse reconstruction errors R^{T*} and R^{F*} obtained in

equation 9 and 10 could be used to assign a test patch to a class as described in equation 4. However, the classifications produced for different patch sizes need to be combined to produce single, final, classification. To do this Mairal et al. [43] used a series of sparsity factors in the procedure of dictionary learning. Therefore each classifier can output a pair of curves of reconstruction errors as a function of the sparsity constraint. Afterwards, all curves generated by multiple sizes of classifiers were combined together to generate a feature vector for classification by a logistic regression classifier or a linear SVM.

As a fixed sparsity constraint was used in this modified version of SRBD, a new method was defined. Specifically, each corresponding pair of R^* was used to calculate a confidence metric (CM).

$$CM = \frac{R^{T*} - R^{F*}}{R^{T*} + R^{F*}} \quad (11)$$

where R^{T*} and R^{F*} represent the sparse reconstruction errors obtained by encoding test patches using the true and false contour dictionaries. There are overall $N/2=8$ CM values obtained from multiple sizes of patches and resolutions of the image. All values of CM continuously range from -1 to 1. Specifically, positive values correspond to the belief that the centre pixel of the patch is a false contour. In contrast, negative values in CM indicate the confidence that the centre pixel of the test patch is a true contour. Only positive values in CM were kept and negative values in CM were set to 0, to generate a confidence of false contour metric (CFM) as defined in:

$$v(x, y) = \begin{cases} u(x, y), & u(x, y) > 0 \\ 0, & u(x, y) \leq 0 \end{cases}$$

$$\text{for } \forall v(x, y) \in CFM, \text{ and } \forall u(x, y) \in CM \quad (12)$$

where $u(x, y)$ represents each value in CM and $v(x, y)$ represents the corresponding value with same location in CFM . Afterwards, $N/2=8$ CFM were summed up and normalized by a function G , which clipped values $w(x, y)$ within the aggregated CFM at a threshold T_2 and then normalized these values by T_2 , as defined in:

$$G(w(x, y), T_2) = \begin{cases} \frac{w(x, y)}{T_2}, & w(x, y) < T_2 \\ 1, & w(x, y) \geq T_2 \end{cases}$$

$$\text{for } \forall w(x, y) \in \sum_{i=1}^8 CFM_i \text{ and } \forall G(w(x, y), T_2) \in NACFM \quad (13)$$

where $w(x, y)$ represents each value in the aggregated CFM ; $NACFM$ whose values are the normalized values in the aggregated CFM by the function G still indicates the confidence of false contour and is used to suppress false contour pixels within an image. The value of T_2 was set to 0.15, 0.65, and 1.15 for c-PC/BC, gPb-owt-ucm, and SCG respectively.

Finally, $1 - NACFM$ was multiplied by the corresponding output of the boundary-detection algorithm that was being refined. The proposed version of SRBD thus acts to suppress boundary pixels it determines to be false contours. When gPb-owt-ucm and SCG were being refined, the output that was suppressed was the final result generated by these two algorithms. However, for c-PC/BC a slightly different approach was used. The algorithm simulates neuron activity in V1, which was a convolutional neural network applied to perform contour detection by using the receptive fields of V1 neurons (modelled as first order derivative of Gaussians) to locate boundaries defined by local intensity and colour discontinuities. The output of c-PC/BC that was suppressed was the predictive neuron responses across multiple orientations which indicated discontinuities in colour and luminance. Afterwards, these predictive neuron responses were used to linearly reconstruct the final contour detection result [27].

In summary, the main difference between the SRBD method [43] used to refine Canny [6] and the modified version of the SRBD method used to refine the three previously proposed methods are in the following aspects:

- The multiple sizes of image patches extracted from multiple resolution of training and test images. The SRBD [43] used $p=5, 7, 9, 11, 15, 19, 23$ for both full scale and half scale images to refine the Canny edge detector [6]. Whereas, the proposed modified version of SRBD only employed some of these sizes of patches including $p=7, 9, 11, 15, 19$ pixels for full scale images and $p=7, 9, 11$ pixels for half scale images.
- The sparsity constraint used in the dictionary learning procedure. The SRBD [43] used to refine Canny defined a discriminative dictionary training method and used a series of sparsity constraint to learn a series of dictionaries for each different size of training image patches. In contrast, the proposed modified version of SRBD is simpler, as it does not use the discriminative dictionary learning method and only uses one sparsity constraint to generate one dictionary for each different size of training image patch.
- The sparse reconstruction error employed to separate false contours from true contours within test images. The SRBD [43] used to refine Canny can output a pair of curves for each different size of image patches plotted by two series of reconstruction errors using the corresponding series of dictionaries

belonging to the true and false contour class as a function of the corresponding series of sparsity constraint. Afterwards, all curves generated by multiple sizes of reconstruction errors were combined together to generate a feature vector for the separation of false contours from true contours by a logistic regression classifier or a linear SVM. In contrast, the proposed modified version of SRBD defined a confidence metric calculated from multiple sizes of sparse reconstruction errors. The positive part of the confidence metric indicates the confidence of a false contour at each pixel within a test image and is employed to suppress contour pixels that it determines to be false contours.

4. Result

The BSDS300 dataset [44] is a standard benchmark for evaluating the performance of contour detection and image segmentation algorithms. The BSDS300 dataset contains 100 test images and 200 training images of size 481-by-321 pixels. They are all natural images of people, animals, plants, buildings, man-made objects and some natural scenes. To quantitatively evaluate the performance of a contour detection algorithm, the BSDS300 benchmark employs the F-score. The F-score is defined as $2PR/(P+R)$ where P is the precision and R is the recall obtained when comparing the algorithm's contours with contours produced by humans. In practice, the F-score varies from 0.41, which is the performance obtained by randomly defining pixels as contours, to 0.79, which is the performance obtained by a human observer compared with other human observers.

The BSDS500 dataset, which was first utilized in [2], is an extended version of the BSDS300. There are 200 fresh natural images added as test images, and the original 100 test images are altered to validation images. Three different quantities are employed for evaluation: the F-score with a fixed threshold for all test images (ODS), which is the same as the F-score used in the BSDS300 benchmark; the aggregated F-score with an optimal threshold per image (OIS); and the average precision (AP) for the full range of recall values, which is equivalent to the area under the precision-recall curve.

Contour detection results derived from c-PC/BC [27], gPb-owt-ucm [2], and SCG [40] were all generated using the code provided by the authors of these algorithms. Afterwards, the result derived from each algorithm was refined by the modified version of the SRBD method proposed in section 3. Figure 1 shows results of

contour detection for two images from each of the two BSDS datasets. These results include contours detected by human observers, and by each of the three algorithms with and without refinement by SRBD. Figure 2 and Figure 3 shows F-scores (ODS) with their precision-recall curves obtained by the different algorithms when their inputs are colour images from the BSDS300 and the BSDS500 respectively. Table 1 reports the ODS, OIS and AP obtained by these three algorithms and their refined versions.

The ODS and OIS of *c*-PC/BC [27] is slightly increased using our proposed SRBD method, and the AP is significantly increased. The result derived from one of the state-of-the-art algorithms, *gPb-owt-ucm*, is slightly improved after the refinement, with the ODS increased from 0.71 to 0.72 and the AP increased from 0.73 to 0.76, when evaluated by the BSDS300 benchmark; however, only the AP is significantly increased from 0.73 to 0.77 for the BSDS500 benchmark. The performance of the other state-of-the-art algorithm, SCG, is also slightly improved in its refined version, with the ODS increased from 0.71 to 0.72, the OIS increased from 0.73 to 0.74 and the AP increased from 0.75 to 0.76, when evaluated by the BSDS300 benchmark. Only the AP is slightly increased from 0.77 to 0.78 for the BSDS500 benchmark.

The BSDS300/500 benchmark applies a set of thresholds ranging from 0 to 1 to convert the contour detection result into a binary map and then output a precision-recall curve at different threshold values. The F-score/ODS or OIS of the method being tested is the highest F-score value in a fixed threshold for all images or a set of optimal thresholds for each image. Because the probability of boundary value of some false contour pixels (most of true contour pixels) within the contour detection results derived from different methods being refined should be significantly decreased (unchanged) by the proposed SRBD, the precision (recall) values at the majority of threshold values then should be increased (should almost have no change) compared with the result without refinement. In other words, some false contour pixels are no longer treated as a contour pixel at the majority of threshold values and most of the true contour pixels are unaffected by the proposed SRBD. As shown in Table 1, the quantitative evaluation results are consistent with the expectation. Specifically, the average precision (AP) values are all increased for the comparison between each of the three previously proposed methods with and without refinement. Also, the recall values have only a slight or no decrease at each threshold value for each of the three methods. However, the recall value may be decreased or

increased in the highest F-score of different refined results because of optimal threshold values may have been changed. Therefore, the F-score/ODS and OIS are all increased for the comparison between each of the three previously proposed methods with and without refinement, although some of these increments are negligible.

The value of each parameter in the proposed modified version of SRBD was determined by trial and error. Specifically, the modified SRBD was operated on training images with a set of manually defined default parameter values. Afterwards, each parameter was sequentially justified while all other parameter values were kept fixed according to the alteration of F-score on these training images. The sequence of the justification of each parameter value is from parameters related to training image patches extraction, dictionary learning, sparse reconstruction error computation, to suppression of contour pixels if the modified SRBD determines as false contours within results derived from the three previously proposed contour detection methods.

5. Discussion

The sparse reconstruction-based discrimination (SRBD) method was previously proposed by Mairal et al. [42] and utilized two dictionaries to place image patches into two classes: edge and non-edge. Afterwards, Mairal et al. [43] applied SRBD to improve the performance of the Canny edge detector [6]. However, given the poor performance of the Canny method in locating salient boundaries, the refined method still produced results that fell far short of state-of-art contour detection methods.

In this paper, a modified version of the SRBD method is applied to refining the contour detection results generated by three different boundary-detection algorithms: the colour Predictive Coding/Biased Competition Model of V1 [27] and the two state-of-the-art algorithms: *gPb-owt-ucm* [2] and Sparse Coding Gradients method [40]. Evaluation using the BSDS300 [44] and BSDS500 [2] datasets indicate that for all three algorithms boundary detection is at least slightly improved by the proposed refinement. In other words, the two state-of-the-art algorithms refined by this proposed version of SRBD can be improved beyond the previous state-of-the-art.

Different elements in dictionaries learnt from training image patches belonging to the true or false contour classes by the SRBD can also be interpreted as different receptive fields of a neuron representing an image feature of true or false contour pixels. Hence, these elements can also be added into a neurophysiologically-

inspired contour detection method implemented by the sparse coding model (such as the PC/BC model of V1 [26]) as the receptive fields of neurons representing true or false contour pixels, which may improve its performance of contour detection.

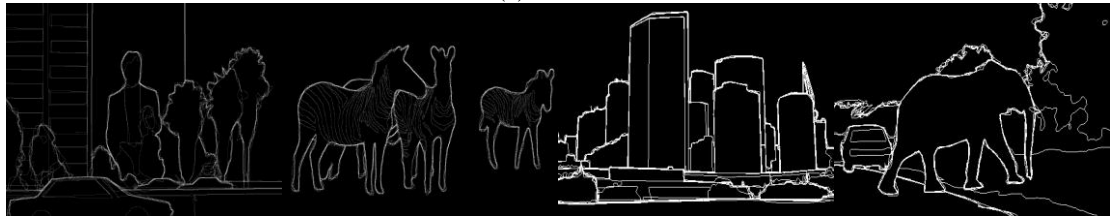
Reference

1. Walther, D. B., Chai, B., Caddigan, E., Beck, D. M., Fei-Fei, L.: 'Simple line drawings suffice for functional MRI decoding of natural scene categories', *Proceedings of the National Academy of Sciences*, April 2011, vol. 108, no. 23, pp. 9661-9666.
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: 'Contour detection and hierarchical image segmentation', *IEEE Tran. Pattern. Anal. Mach. Intell.*, 2011, vol. 33, pp. 898-916.
3. Lowe, D. G.: 'Object recognition from local scale-invariant features', In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, 1999, vol. 2, pp. 1150-1157.
4. Comport, A., Marchand, É., Chaumette, F.: 'Robust model-based tracking for robot vision', In *Intell. Robots and Systems, 2004. The proceedings of IEEE/RSJ International Conference on*, 2004, vol.1, pp. 692-697.
5. Chalana, V., Linker, D. T., Haynor, D. R., Kim, Y.: 'A multiple active contour model for cardiac boundary detection on echocardiographic sequences', *IEEE Tran. Medical Imaging*, Jun 1996, vol. 15, pp. 290-298.
6. Canny, J.: 'A Computational approach to edge detection', *IEEE Tran. Pattern. Anal. Mach. Intell.*, 1986 vol. 8, pp. 679-698.
7. Jain, A. K.: 'Fundamentals of digital image processing', Prentice-Hall, Englewood Cliffs., 1989.
8. Marr, D., Hildreth, E.: 'Theory of edge detection', *Proc. R. Soc. Lond. B: Bio. Sci.*, 1980, vol. 207, pp. 187-217.
9. Papari, G., Petkov, N.: 'Edge and line oriented contour detection: State of the art', *Image. Vision. Comput.*, 2011, vol. 29, pp. 79-103.
10. Comaniciu, D., Meer, P.: 'Mean shift: A robust approach toward feature space analysis', *IEEE Tran. Pat. Anal. Mach. Intell.*, 2002, vol. 24, pp. 603-619.
11. Tomasi, C., Manduchi, R.: 'Bilateral filtering for gray and color images', *Proceedings of the Sixth International Conference on Computer Vision*, Jan 1998, pp. 839-846.
12. Pitas, I., Venetsanopoulos, A. N.: 'Order statistics in digital image processing', *Proceedings of the IEEE*, 1992, vol. 80, pp.1893-1921.
13. Lee, J. S. J., Haralick, R. M., Shapiro, L. G.: 'Morphologic edge detection', *IEEE Journal Robot. Auto.*, 1987, vol. 3, pp. 142-156.
14. Huang, J. S., Tseng, D. H.: 'Statistical theory of edge detection', *Comput. Vision Graph Image Process*, 1988, vol.43, pp. 337-346.
15. Ojala, T., Pietikainen, M.: 'Unsupervised texture segmentation using feature distributions', *Pattern. Recog.*, March 1999, vol.32, pp.477-486.
16. Kovese, P.: 'Image features from phase congruency', *Videre: Journal Comput. Vision Research.*, 1999, vol.1, pp. 1-26.
17. Mallat, S. G.: 'A theory for multiresolution signal decomposition: the wavelet representation', *IEEE Trans. Pat. Anal. Mach. Intell.*, 1989, vol.11, pp. 674-693.
18. Kass, M., Witkin, A., Terzopoulos, D.: 'Snakes: Active contour models', *Intern. Journal Comput. Vision*, 1988, vol. 1, pp. 321-331.
19. Shi, J., Malik, J.: 'Normalized cuts and image segmentation', *IEEE Trans. Pat. Anal. Mach. Intell.*, 2000, vol. 22, pp. 888-905.
20. Najman, L., Schmitt, M.: 'Geodesic saliency of watershed contours and hierarchical segmentation', *IEEE Trans. Pat. Anal. Mach. Intell.*, 1996, vol. 18, pp.1163-1173.
21. Arbelaez, P.: 'Boundary extraction in natural images using ultrametric contour maps', *IEEE Conference on Computer Vision Pattern Recognition Workshop*, 2006, pp. 182.
22. Yadollahi, M., Procházka, A., Kašparová, M., Vyšata, O.: 'The use of combined illumination in segmentation of orthodontic bodies', *Signal, Image Video Process.*, 2015, vol. 9, pp. 243-250.
23. Nadernejad, E., Sharifzadeh, S.: 'A new method for image segmentation based on fuzzy c-means algorithm on pixonal images formed by bilateral filtering', *Signal, Image Video Process.*, 2013, vol. 7, pp. 855-863.
24. Ladgham, A., Hamdaoui, F., Sakly, A., Mtibaa, A.: 'Fast MR brain image segmentation based on modified shuffled frog leaping algorithm', *Signal, Image Video Process.*, 2015, vol. 9, pp. 1113-1120.
25. Spratling, M. W.: 'Image segmentation using a sparse coding model of cortical area V1', *IEEE Trans. Image. Process.*, Feb 2013, vol. 22, no. 4, pp. 1631-1643.
26. Spratling, M. W.: 'Predictive coding as a model of response properties in cortical area V1', *The Journal of Neuroscience*, March 2010, vol. 30, pp. 3531-3543.
27. Wang, Q., Spratling, M. W.: 'Color Image Segmentation Using the Predictive Coding/Biased Competition Model of V1', submitted to the *Journal: Cognitive Computation*.
28. Johnson, E. N., Hawken, M. J., Shapley, R.: 'The orientation selectivity of color-responsive neurons in macaque V1', *The Journal of Neuroscience*, 2008, vol. 28, pp. 8096-8106.
29. Mignotte, M.: 'A biologically inspired framework for contour detection', *Patt. Analy. App.*, 2015, pp. 1-17.
30. Benoit, A., Caplier, A., Durette, B., Herault, J.: 'Using human visual system modeling for bio-inspired low level image processing', *Comput. vision Image understand.*, 2010, vol. 114, pp. 758-773.
31. Grigorescu, C., Petkov, N., Westenberg, M. A.: 'Contour and boundary detection improved by

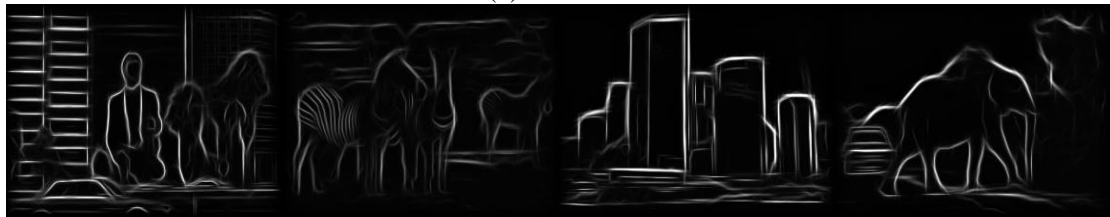
- surround suppression of texture edges', *Image Vision Comput.*, 2004, vol. 22, pp. 609-622.
32. Martin, D. R., Fowlkes, C. C., Malik, J.: 'Learning to detect natural image boundaries using local brightness, color and texture cues,' *IEEE Trans. Pattern Anal. Mach. Intell.*, May 2004, vol. 26, no. 5, pp. 530-549.
 33. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: 'Using contours to detect and localize junctions in natural image', *Comput. Vision. Pattern. Recog.*, June 2008, pp. 1-8.
 34. Ren, X.: 'Multi-scale improves boundary detection in natural image', *Computer Vision-ECCV 2008*, Oct 2008, pp. 533-545.
 35. Song, Y. Z., Arbelaez, P., Hall, P., Li, C., Balikai, A.: 'Finding semantic structures in image hierarchies using laplacian graph energy', *Comput. Vision-ECCV 2010*. Springer Berlin Heidelberg, 2010, pp. 694-707.
 36. Elad, M., & Aharon, M.: 'Image denoising via sparse and redundant representations over learned dictionaries', *IEEE Trans. Image. Process.*, 2006, vol. 15, pp. 3736-3745.
 37. Mairal, J., Elad, M., Sapiro, G.: 'Sparse representation for color image restoration', *IEEE Trans. Image. Process.*, 2008, vol. 17, pp. 53-69.
 38. Skretting, K., Husøy, J. H.: 'Texture classification using sparse frame-based representations', *EURASIP. Jour. Applied. Signal. Process.*, Jan. 2006, pp. 102-102.
 39. Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., Ma, Y.: 'Robust face recognition via sparse representation', *IEEE Trans. Patt. Anal. Mach. Intell.*, 2009, vol. 31, pp. 210-227.
 40. Ren, X., Bo, L.: 'Discriminative trained sparse code gradients for contour detection', In *Advances. Neur. Inform. Process. System.*, 2012, pp. 584-592.
 41. Aharon, M., Elad, M., Bruckstein, A.M.: 'The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations', *IEEE Trans. Signal. Process.*, 2006, vol. 54, pp. 4311-4312.
 42. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: 'Discriminative learned dictionaries for local image analysis', *IEEE Confer. Comput. Vision. Pattern. Recog.*, 2008, pp. 1-8.
 43. Mairal, J., Leordeanu, M., Bach, F., Hebert, M., Ponce, J.: 'Discriminative sparse image models for class-specific edge detection and image interpretation', In *Computer Vision-ECCV 2008*, 2008, pp. 43-56.
 44. Martin, D., Fowlkes, C., Tal, D., Malik, J.: 'A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,' in *Proc. Int. Conf. Computer. Vision*, Jul 2001, vol. 2, pp. 416-423.
 45. Rubinstein, R., Zibulevsky, M., Elad, M.: 'Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit,' *CS Technion*, 2008, vol. 40, pp. 1-15.



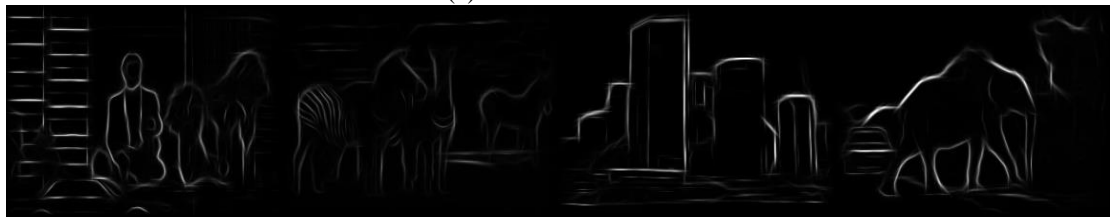
(a) Human



(b) c-PC/BC



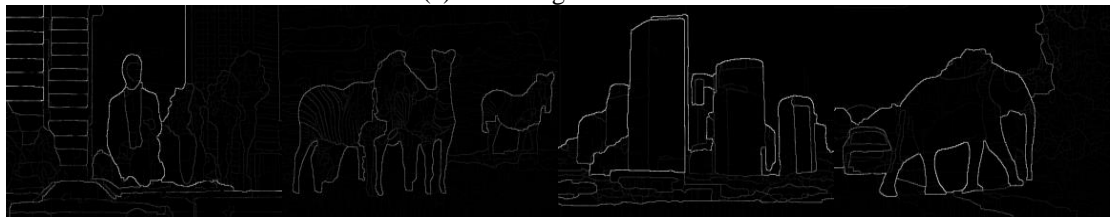
(c) refined c-PC/BC



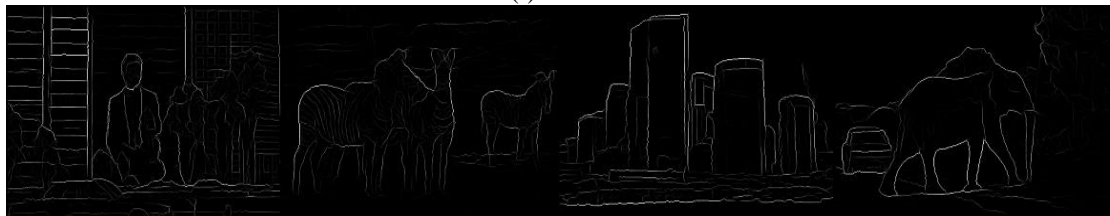
(d) gPb-owt-ucm



(e) refined gPb-owt-ucm



(f) SCG



(g) refined SCG

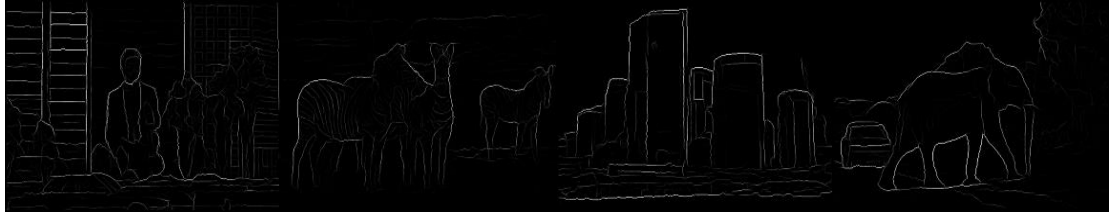


Figure 1 Results of Contour Detection Derived from Three Previously Proposed Contour Detection Methods with and without Refinement by the Proposed SRBD for Four Images from BSDS300 [44] and BSDS500 [2] Dataset.

Contour detection results are produced by (from top to bottom): (a) human, (b) c-PC/BC, (c) refined c-PC/BC, (d) gPb-owt-ucm, (e) refined gPb-owt-ucm, (f) SCG, and (g) refined SCG. Some contours within the results obtained from three previously proposed methods that are not identified by human observers are accurately suppressed by the proposed modified version of SRBD in their refined contour detection results.

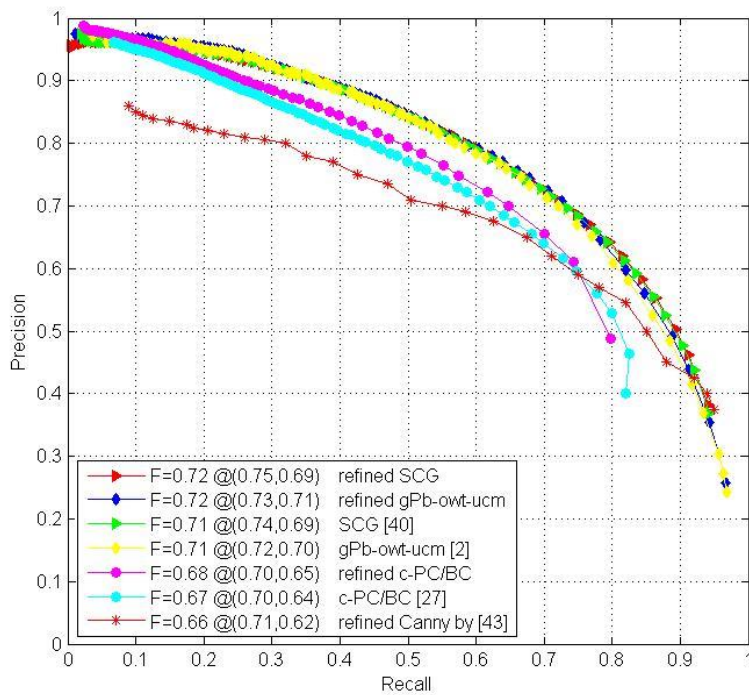


Figure 2 Performances of Contour Detection Algorithms Evaluated by the Precision-recall Curve of the F-score from the BSDS300 Benchmark [44].

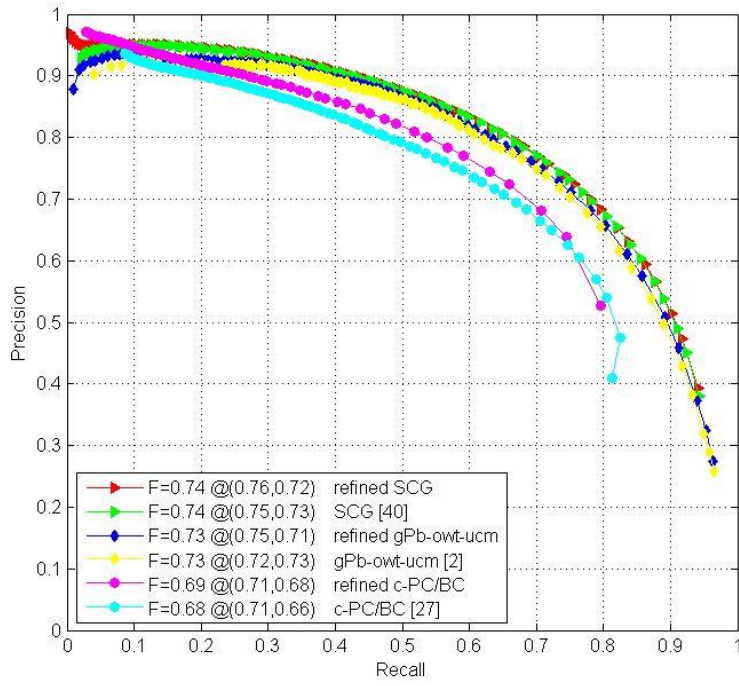


Figure 3 Performances of Contour Detection Algorithms Evaluated by the Precision-recall Curve of the ODS (F-score) from the BSDS500 Benchmark.

Methods	BSDS300			BSDS500		
	ODS	OIS	AP	ODS	OIS	AP
Human	0.79	0.79	—	0.80	0.80	—
Refined SCG	0.72	0.74	0.76	0.74	0.76	0.78
SCG [39]	0.71	0.73	0.75	0.74	0.76	0.77
Refined gPb-owt-ucm	0.72	0.74	0.76	0.73	0.76	0.77
gPb-owt-ucm [2]	0.71	0.74	0.73	0.73	0.76	0.73
Refined c-PC/BC	0.68	0.69	0.63	0.69	0.71	0.64
c-PC/BC [24]	0.67	0.68	0.59	0.68	0.70	0.58

Table 1 The ODS, OIS and AP of Different Contour Detection Methods