EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Fast boundary extraction of color image using negative divergence of a normal compressive vector field

Samruan Wiangsamut, Nawapak Eua-Anant* and Wutthichai Phornphatcharaphong

Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand

Received 7 February 2021
Revised 5 May 2021
Accepted 6 May 2021

Abstract

Among low level gradient-based edge detection techniques, boundary extraction algorithms based on particle motion yield superior continuous edge map results. However, these methods sequentially tracing particle trajectories to obtain continuous edges can be slow in the case of images having a number of objects or spurious edges. In order to accelerate such an edge tracking process, this paper proposes the use of negative divergence of a normal compressive vector field to enable fast edge detection. By exploiting the compressive property and negative divergence of the normal compressive vector field, prominent edges can be rapidly detected in a raster scan manner. The remaining incomplete or broken boundaries are later fixed using the boundary extraction algorithm based on particle motion. Image segmentation performance of the proposed algorithm was evaluated using the BSDS500 benchmark dataset with the F-measures for ODS and OIS, with average precision and computation time used as performance measurements. Experimental results indicated that the proposed algorithm provided results comparable to those of the well-known low-level methods, while the average computation time was drastically reduced by a factor of 2 when compared to that of the original particle motion based method.

Keywords: Boundary extraction, Negative divergence, Normal compressive vector field

1. Introduction

Image segmentation partitioning of images into several regions possessing homogenous properties (color, texture, etc.) is essential for solving problems of computer vision such as: object detection/tracking /recognition, image classification and image retrieval. Among image segmentation techniques, region-based techniques focus on grouping pixels, having the same properties, into regions while edge-based techniques focus on locating the boundaries between regions where pixel properties are abruptly changed. Edge-based methods have an advantage over region-based methods as, in edge-based methods, extracting boundaries requires reduced pixel searching compared to extracting all the pixels, of the entire region, in region-based methods. In edge-based methods, boundary information is retrieved by means of edge detection or boundary extraction to locate the transitions of pixel properties such as light, color, shade, and texture. Boundary extraction can be broadly categorized into three main approaches: low-level methods, machine-learning and deep-learning [1]. The low-level methods utilize low-level features such as Canny [2], graph searching[3], and particle motion [4, 5] which are the classic boundary extraction methods to determine boundaries based on image gradient or changes of pixel properties. Extracted boundaries can be in the form of discretized paths obtained from edge pixel tracing [6], or edge weight graph searching [3], or particle trajectories in the vector fields derived from image derivatives[4-8].

In contrast to the low level methods, machine learning approaches such as probabilistic decision tree [9], support vector machine [10], random forest [11] utilize varieties of relevant features such as light, color, shade, texture and transform the coefficients to determine object boundaries. For example, the global probability of the boundary proposed in [12] combines the oriented gradients of histograms of color and texture channels, at multiple scales, to represent contour images. Complex relations between input features and extracted results are implicitly obtained by training the models with user-provided training data. The decision tree models in [9], with multi-scale input features including image gradients, difference of histograms, and Haar wavelets of image patches, are trained using the ground truths of edges provided by users, deliver output in the form of probabilities of pixels belonging to edges. The method in [10] uses the sparse coding gradient adapted from the global probability of boundary [12] as the main feature of the support vector machine classifier. Although, machine learning approaches can outperform classical boundary extraction methods, the efficiency of machine learning approaches inarguably depend on the selection of features and the separability of training data sets provided by users. In most machine learning approaches, training data sets must be composed of large numbers of feature image patches containing various edge patterns [11].

The latest approaches to boundary extraction methods, such as [13-18], are applications of deep learning networks, especially convolution neural networks (CNN), which disruptively arose in the field of computer vision a few years ago. The CNNs are one type of feature-representative artificial neural networks capable of creating their own features in the beginning layers by using adaptable convolutional masks and utilize such data in the ending layers, for detection/classification proposes, by using fully connected, artificial neural networks. Superior to classical image segmentation, CNN approaches are capable of performing semantic image segmentation,

*Corresponding author. Tel.: +08 9620 4196
Email address: nawapak@kku.ac.th
doi: 10.14456/easr.2022.19

for example, general images [13, 18, 19] and medical image segmentation [20-25], which cannot be achieved by rule based classical image segmentation techniques. To apply CNNs for contour/edge detection, training data sets are usually composed of various classes of image patches, whose centers pass through contours, or image patches containing structured edges as well as none edge image patches [14, 15]. Since structures of CNNs are naturally multi-scale encoded, multi-scale or holistic edge detection, that can resolve ambiguity of edges at multiple scales, can properly be integrated into CNN techniques by means of both network architectures [16, 17] and training data sets [15].

Despite the success of today's machine-learning and deep-learning image segmentation techniques, the main drawback of these techniques is the requirement for enormous training data sets; as an example, a set of 2,000,000 image patches [15], which is difficult to prepare and requires massive computational resources for training the models. Alternatively, this paper focuses on improving the low-level boundary extraction method based on the particle model which is still one of the most robust low-level boundary extraction methods. The particle motion based method was first proposed by Eua-Anant and Udpa [4, 5] for extracting object edges in a grayscale image and later extended to extract edges in a color/multispectral image [8]. The effects of parameters controlling particle motion in a vector image field was investigated in [7] using the c-evolute model to describe the behavior of a particle under influence from vector fields. Similar to particle models, the point flow model [6] also utilizes vector fields derived from image derivatives to force random points to flow along edges. The main disadvantage of all particle motion based methods is that the methods sequentially tracing particle positions, step-by-step, to obtain consecutive edges are time consuming. In order to reduce processing time, in this paper, the fast boundary extraction algorithm utilizing negative divergence property of a normal compressive vector field is introduced. Later, discontinuous and incomplete extracted edges can be extended to achieve close boundaries using the traditional particle motion based boundary extraction technique. Complete details are given as follows. Previous work on boundary extraction is given in Section 2. The proposed algorithm is presented in Section 3. Next, experimental results and discussion are shown in Sections 4-5. Finally, the conclusion is provided in Section 6.

2. Previous works

The original boundary extraction algorithm based on particle motion, namely the Particle Motion in a Vector Image Field (PMIVF) [4, 5], utilizes two orthogonal vector fields: the edge vector field (EVF), $\vec{e} = -\frac{\partial P}{\partial x}\hat{\imath} + \frac{\partial P}{\partial y}\hat{\jmath}$, forcing the particle to travel along object edges and the normal compressive vector field (NCVF), $\vec{n} = \frac{1}{C}\nabla P \cdot \nabla P^2$, forcing the particle to stick to object edges, to drive the particle and obtains particle trajectories as the extracted boundaries. These two vector fields have their unique features in which vectors in $\vec{e}$ form currents flowing around object edges while vectors in $\vec{n}$ are orthogonal to the closest object edges. Using $\vec{e}$ and $\vec{n}$ as driving forces, the consecutive particle position is given by

$$\vec{P}_{k+1} = \vec{P}_k + \alpha\vec{e}_k + \beta\vec{n}_k \tag{1}$$

where $\vec{P}_k$ is the k[th] particle position vector; $\vec{e}_k$ and $\vec{n}_k$ are, respectively, edge vector and normal compressive vector, interpolated at the k[th] particle position; α and β are tangential and normal stepping factors respectively.
The PMVIF algorithm is simple, effective and provides continuous extracted boundaries. In addition, by using polynomial interpolation, the extracted boundary in Equation 1 can be achieved at subpixel resolution. However, the main drawback of the original method is that it cannot be applied to a color/multichannel image since all vector fields are computed from partial derivative operations that are defined only for a scalar field.

In order to overcome this limitation, W. Phornphatcharaphong and N. Eua-Anant [8] proposed a new scheme, namely the particle motion in a vector image field derived from local color distance images (PMLCD), to compute the normal compressive and edge vector fields of a color/multichannel image. A local color distance image $LCD(x-i, y-i)$ is defined as an image patch centered at Pixel (i, j) in which the value of each pixel is the distance in the color space between Pixel (x, y) and Pixel (i, j). Later, the collection of the center-to-centroid vectors of all local color distance images, i.e,

$$\vec{n}(i,j) = \begin{bmatrix} n_x(i,j) \\ n_y(i,j) \end{bmatrix} = \frac{1}{C}\begin{bmatrix} \sum_x \sum_y (x-i)\cdot LCD(x-i, y-j) \\ \sum_x \sum_y (y-j)\cdot LCD(x-i, y-j) \end{bmatrix} \tag{2}$$

where C is a normalization factor making $max|\vec{n}| = 1$, is used as the new normal compressive vector field.

Figure 1 illustrates an example of $\vec{n}$ obtained Equation (2) that possesses the normal compressive property in which all vectors point to the nearest edges. In other words, a direction of a vector on one side of the object is always opposite to that of a vector on another side of the object.

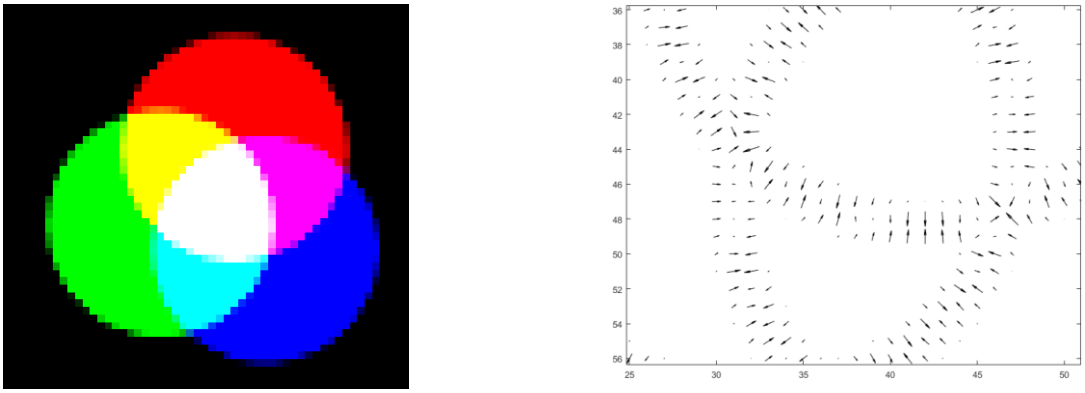
For the edge vector field, $\vec{e}$ used in the PMVIF method is derived from a Hamiltonian gradient vector field which cannot be applied to a multichannel image. Phornphatcharaphong and Eua-Anant proposed a linearization technique, based on the number base system [8], to transform an original multichannel image into a scalar auxiliary image $Aux(i, j)$. Finally, the edge vector field $\vec{e}$ is given by

$$\vec{e}(i,j) = \begin{bmatrix} e_x(i,j) \\ e_y(i,j) \end{bmatrix} = \frac{1}{C}\begin{bmatrix} -\sum_x \sum_y (y-j)\cdot sing(Aux(x,y) - Aux(i,j))\cdot LCD(x-i, y-j) \\ \sum_x \sum_y (x-i)\cdot sing(Aux(x,y) - Aux(i,j))\cdot LCD(x-i, y-j) \end{bmatrix} \tag{3}$$

where C is a normalization factor making $max|\vec{e}| = 1$.

Figure 2 illustrates an example of $\vec{e}$ obtained from Equations (3) where edge vectors form currents flowing around object edges. This new computing scheme, applicable to both scalar and multichannel images, has enabled the particle-motion based boundary

extraction algorithm to be applied to all image types. Nevertheless, the weak point of the boundary extraction algorithms based on particle motion is that the processing time is mainly consumed by the particle trajectory tracking step. These algorithms sequentially tracking particle positions step-by-step can be slow in cases where images have many objects or noisy images have many spurious edges. In order to accelerate the boundary extraction process, the boundary tracking process can be substituted by the swift raster scan method utilizing the unique property of the normal compressive vector field, described in the next section.



(**a**) (**b**)

Figure 1 (a) RGB Color image example, (b) $\vec{n}$ obtained using Equation 2

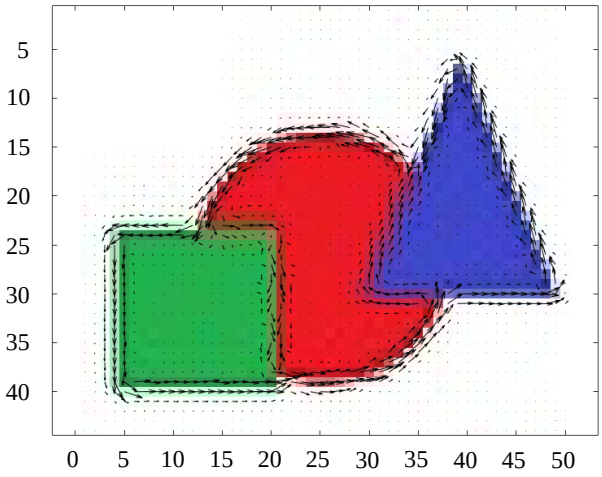


Figure 2 Example of obtained using Equation (3)

3. The proposed fast edge detection algorithm

As mention earlier, edge tracking is a time consuming process where a sequential boundary extraction computer program can track only one edge at a time. Moreover, in Equation 1, if a tangential stepping factor α is small, a particle will need several steps in order to travel across each edge pixel resulting in unnecessary lengthy computation time. On the other hand, as demonstrated in Figure 1b, in the normal compressive vector field $\vec{n}$, all vectors near object edges always point to the nearest edges. This phenomenon can be reflected by negative divergence of the field. Therefore, in this paper, some prominent edges can be obtained without requiring the boundary extraction process simply by detecting pixels of $\vec{n}$ having large negative divergence. Afterward, the obtained results may contain discontinuous edges which can be corrected by applying the boundary extraction algorithm to the remaining undetected edges.

3.1 The overall method

Figure 3 displays the overall block diagram of the proposed method. First, a color image is filtered using a smoothing filter to reduce noise. Next, $\vec{n}$ and $\vec{e}$ are computed, using Equations 2 and 3 respectively, followed by calculating $\nabla \cdot \vec{n}$. Later, the fast edge detection algorithm, based on the negative divergence, is performed to obtain an edge map. Finally, the obtained edge detection result is improved by connecting discontinuous edges by using the PMLCD algorithm.

3.2 Divergence estimation

Divergence is a vector operator performing on a vector field representing the density of the outward flux of a vector field at a given point, defined in a 2-dimensional space as $\nabla \cdot \vec{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y}$. In other words, divergence measures the quantity of the vector field which diverges from a point (positive divergence) or converges to the point (negative divergence) as demonstrated in Figure 4. For a discrete vector field $\vec{n}$, divergence of $\vec{n}$ can be estimated as follows:

$$\nabla \cdot \vec{n}(i,j) = \frac{1}{C} \sum_{(x,y)\in N(i,j)} (x-i)\cdot n_x + (y-j)\cdot n_y \quad (4)$$

where (x, y) is a pixel inside the neighboring area $N(i, j)$ of a pixel (i, j) and C is a normalization factor making $max|\nabla \cdot \vec{n}| = 1$. Figure 5 illustrates an example of $\vec{n}$ and $\nabla \cdot \vec{n}$ computed using Equations 2 and 4 respectively. As seen in Figure 5a, all vectors in $\vec{n}$ always point to the nearest edges forming the "compression" characteristic. In other words, the normal compressive vectors near object edges always converge to the closest edges resulting in a strong negative divergence at the object boundaries, as demonstrated on Figure 5b. Hence, strong negative divergence of $\vec{n}$ can be used as a criterion for screening candidates of edge pixels. Furthermore, the directions of $\vec{n}$ at edge pixels on one side of the edge are opposite to those on the opposite side of the edge. Therefore, to pinpoint exactly the locations of object edges, pairs of adjacent pixels, having large negative values of $\nabla \cdot \vec{n}$ and $\vec{n}$ having opposite direction to its counterpart, can be selected as boundary pixels.

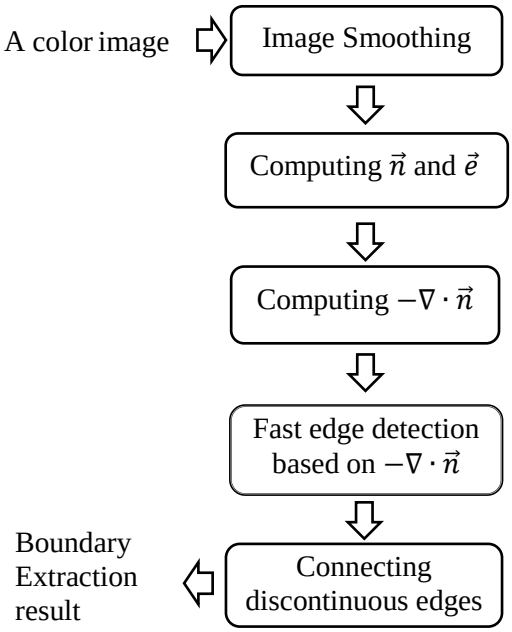
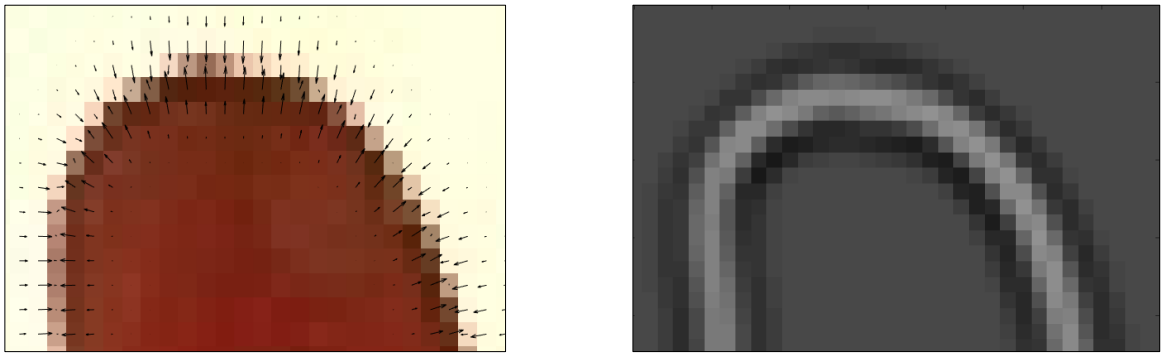


Figure 3 Block diagram of the proposed algorithm

(**a**) (**b**)

Figure 4 Divergence of a vector field: (a) positive divergence and (b) negative divergence



(**a**) (**b**)

Figure 5 (a) $\vec{n}$ and (b) $\nabla \cdot \vec{n}$ obtained using Equation 4

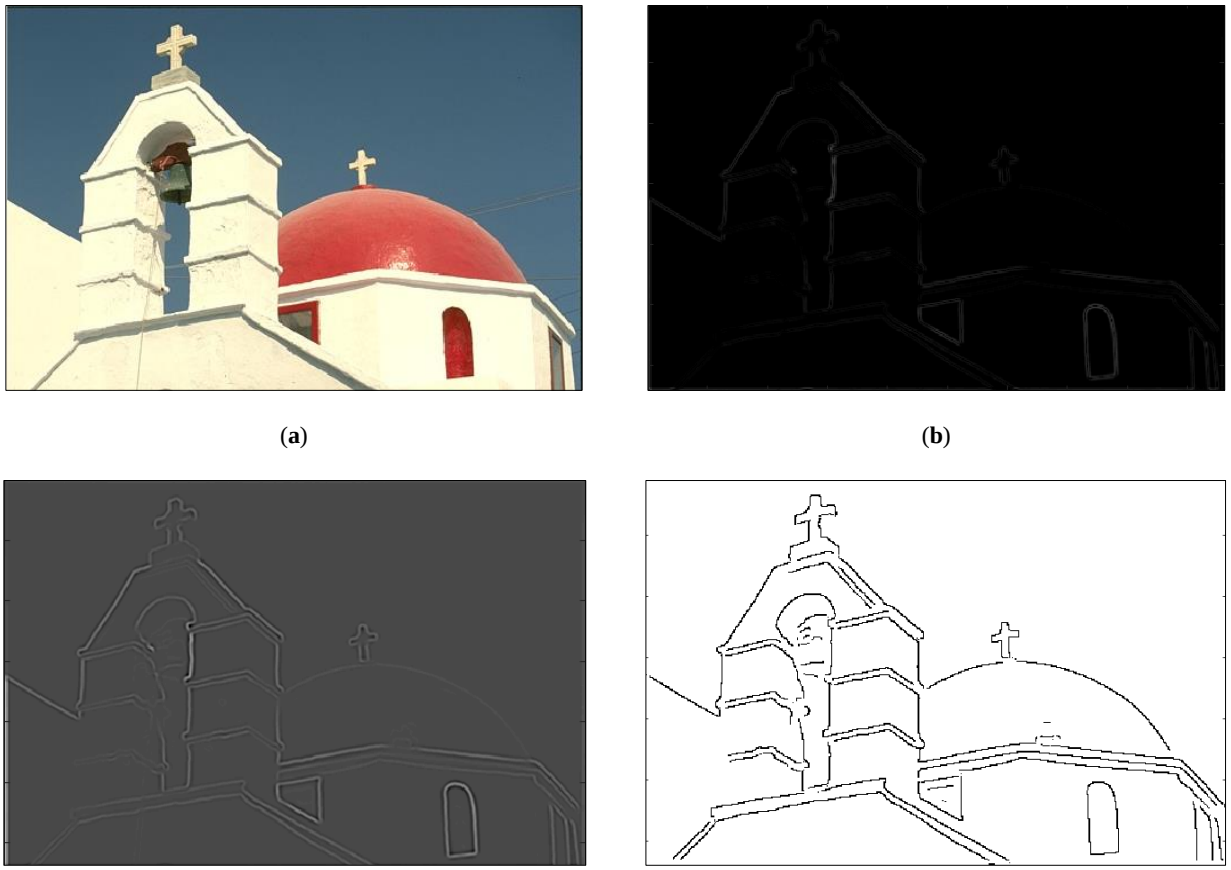
3.3 Fast edge detection based on negative divergence of $\vec{n}$

To exploit the idea described above, in this research, the algorithm for fast edge detection based on negative divergence is proposed. First, candidates of edge pixels are selected from pixels having the negative values of divergence $\nabla \cdot \vec{n}$ larger than a predefined threshold T. Such pixels represent the areas of strong edges. Nevertheless, this threshold-based method normally yields multiple-pixel-wide edges which are undesirable. In order to locate object edges more precisely, another feature of $\vec{n}$ is used. As seen in Figure 5a, since the vectors $\vec{n}$ in the proximity of object edges always converge to the closest edges, the directions of vectors $\vec{n}$ inside the object are opposite to those on the background side. This phenomenon can be reflected by the negative value of dot product between vectors $\vec{n}$ on the opposite sides of the object. Practically, for a 2-dimensional discrete image, dot products between $\vec{n}$ at adjacent pixels in both horizontal and vertical directions are evaluated for locating edge pixels. In summary, the pseudo code for the proposed fast edge detection is as follows:

For a given pixel(i,j) *with* $-\nabla \cdot \vec{n}(i,j) > T$

$$\begin{aligned} &If \cdot \vec{n}(i,j) \cdot \vec{n}(i,j+1) < 0 \; then \; mark \; Pixels \; (i,j) \; and \; (i,j+1) \; as \; edge \; pixels \\ &If \cdot \vec{n}(i,j) \cdot \vec{n}(i+1,j) < 0 \; then \; mark \; Pixels \; (i,j) \; and \; (i+1,j) \; as \; edge \; pixels \end{aligned} \tag{5}$$

The advantage of this algorithm is that it can be performed swiftly in a raster scan manner. Figure 6d illustrates the edge detection result of a color image obtained using the proposed algorithm where most prominent edges are detected.



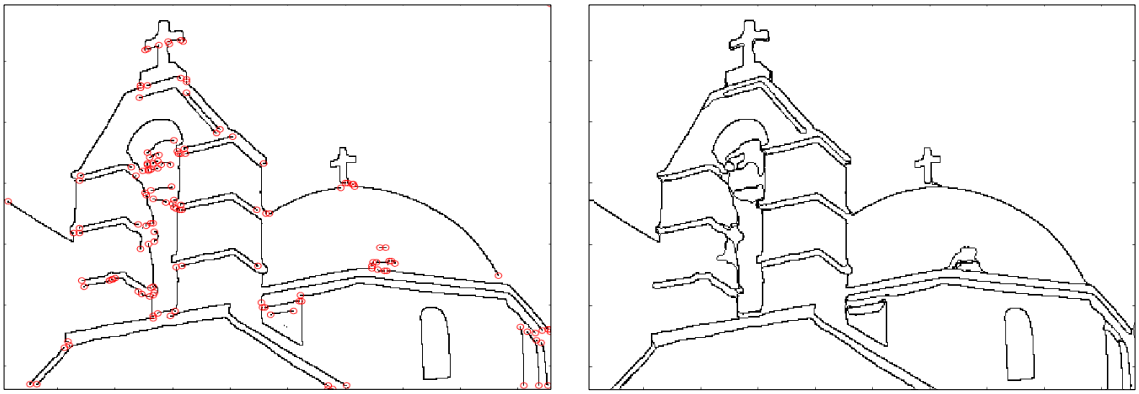
(**a**) (**b**)

(**c**) (**d**)

Figure 6 Fast edge detection result: (a) a color image from the BSDS500 data set, (b) $|\vec{n}|$, (c) $-\nabla \cdot \vec{n}(i,j)$ and (d) the edge map obtained using the proposed fast edge detection algorithm.

3.4 Connecting discontinuous edges

Although the proposed method can rapidly extract strong edges, the obtained edge map still contains several undesirable discontinuous edges, especially at object corners, as indicated by the end points of broken edges in Figure 7a. In order to connect discontinuous edges, the PMLCD algorithm [8] is subsequently employed for extracting the remaining undetected edges using the end points of broken edges, shown in Figure 7a, as the starting points of the particle trajectories. In the PMLCD algorithm, the particle motion expressed by Equation 1, containing an edge vector field forcing the particle to move along an object edge, will extend the discontinued boundaries to reach other extracted edges resulting in the completely closed boundaries as shown in Figure 7b. Since most dominant edges are already detected by using the previous fast edge detection algorithm, only a fraction of edge detection results are obtained using the PMLCD boundary extraction algorithm. Hence, the overall computational time for boundary extraction is drastically reduced.



(**a**) (**b**)

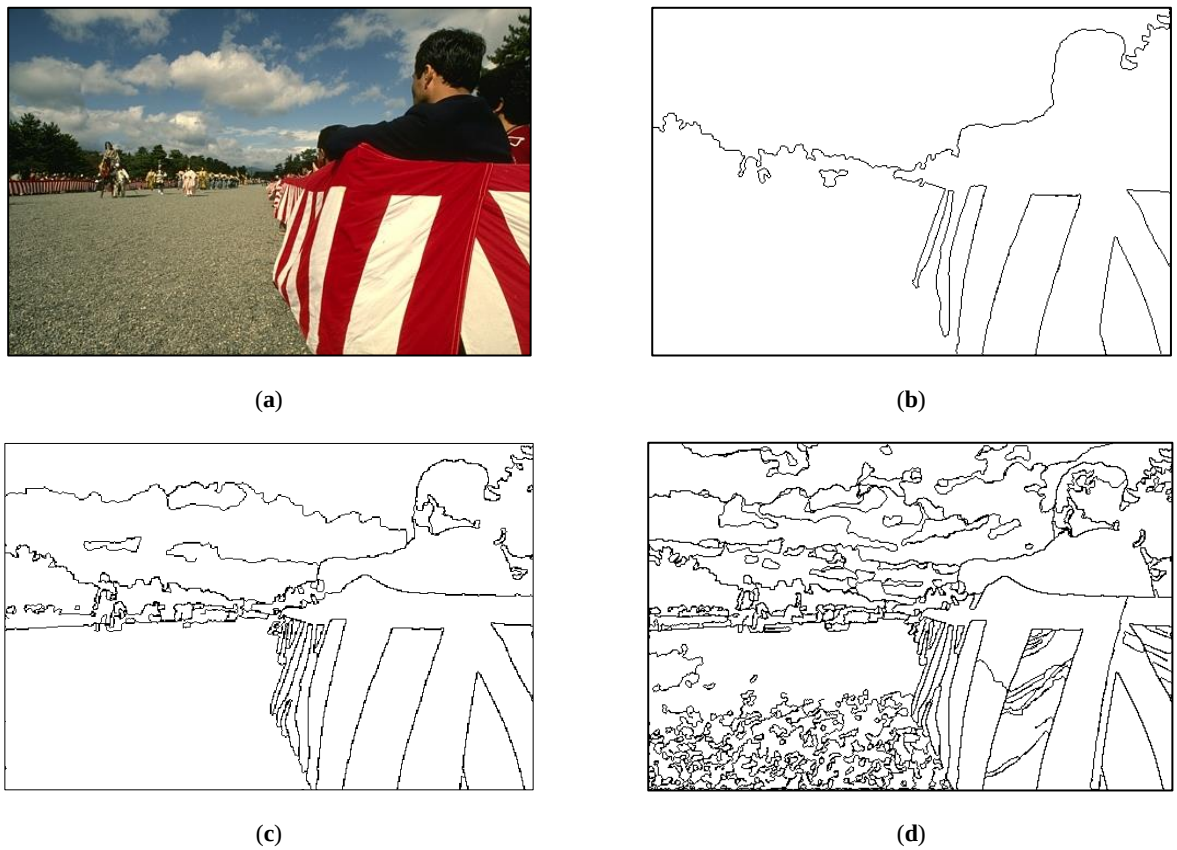
Figure 7 (a) End points of broken edges used as starting points for the boundary extraction process (b) The complete boundary extraction result.

4. Experimental results

In this paper, the BSDS500 dataset [26] containing natural images such as humans, animals, plants, nature, buildings, etc., has been used as the benchmark dataset. The BSDS500 dataset consists of a training set of 200 images, a testing set of 200 images, and a validating set of 100 images. The ground-truth boundaries are hand-drawn by four different human subjects. The parameters involved in the proposed method consist of the radius R of a circular moving window for computing $\vec{n}$, $\vec{e}$and $\nabla \cdot \vec{n}$ using Equations 2, 3 and 4 respectively, the threshold value T for comparing $-\nabla \cdot \vec{n}$ in the formula 5, α and β for the boundary extraction process, based on the particle motion expressed by Equation 1.

4.1 Selection of T

Selection of a threshold is still a challenging problem in many edge-based edge detection techniques [2, 5, 8]. In the proposed method, the threshold value T applied to $\nabla \cdot \vec{n}$ for selecting strong edge pixels is a crucial factor affecting the result. Figure 8 shows the extracted results obtained using various values of T. As seen in Figure 8(b), if the value of T is too high, more strong edges are missed. In contrast, if the value of T is too low, more spurious edges are detected as seen in Figure 8(d).



(**a**) (**b**)

(**c**) (**d**)

Figure 8 Effect of T on extracted boundaries: (a) original image results obtained using (b) T=0.4, (c) T=0.02 and (d) T=0.001.

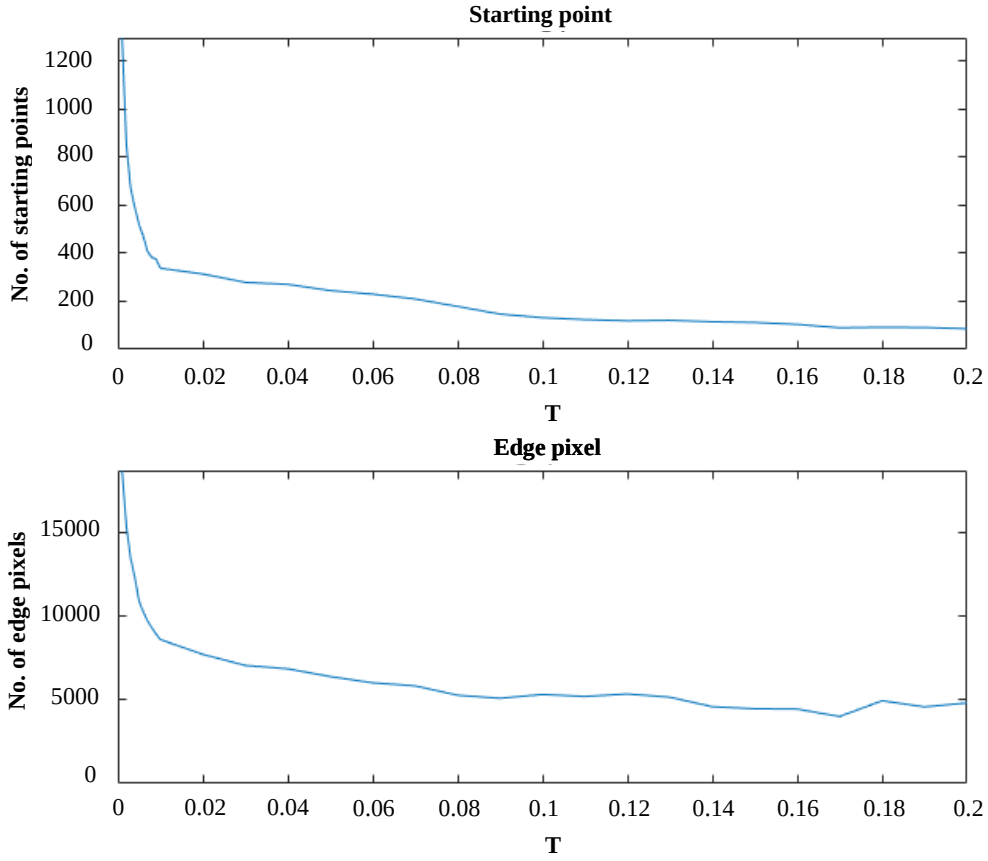


Figure 9 No of edge pixels and starting points vs *T*

Figure 9 displays graphs between no. of edge pixels and starting points obtained during the fast edge detection process versus *T* of the image in Figure 8(a). As *T* decreases from 0.2, both no. of edge pixels and starting points increase steadily. Until the value of *T* reaches the points around T=0.01, the curves start to climb rapidly indicating that a lot of spurious edges have occurred in the results. To investigate the appropriate value of *T*, 100 images from the BSDS500 database were tested to determine the optimum *T* for each image that delivers the highest F-measure score compared to its ground-truths. Figure 10 shows the histogram of the optimum *T* obtained from the experiment where 95% of the optimum values of *T* fall within the range of 0.003-0.12.

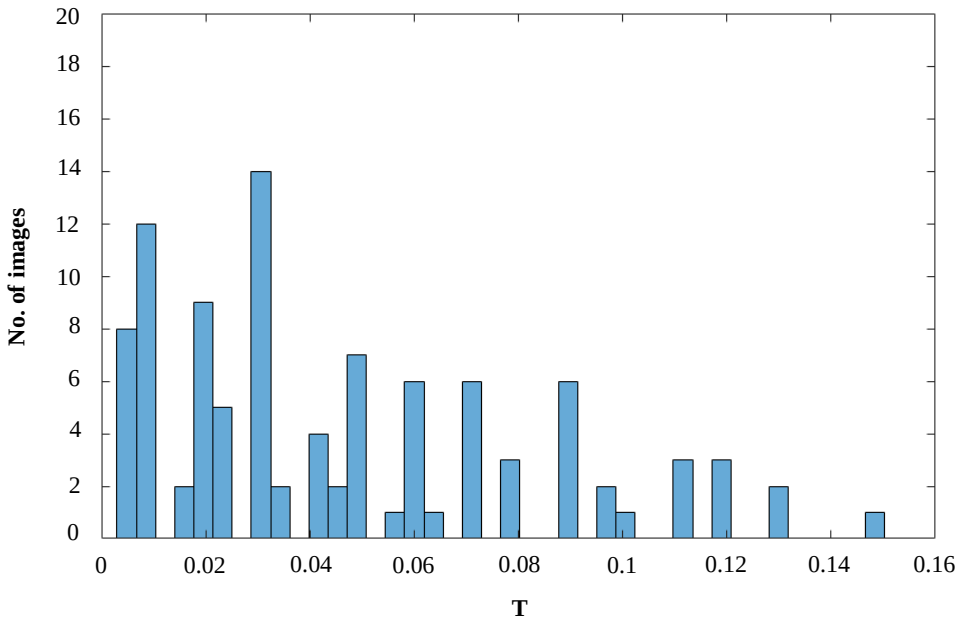


Figure 10 Histogram of the optimum *T* providing the highest F-measure score of 100 images from the BSDS500 database.

As seen, the optimum *T* varies differently, image by image, depending on details in each image. In order to avoid detecting spurious edge pixels (over segmentation) or missing important edges (under segmentation), the values of *T* used in this paper are obtained by adjusting *T* between 0.003-0.12 to achieve the highest F-measure score according to the ground-truths of each image. Examples of the boundary extraction results obtained using the proposed method are shown in Figures 11-13.

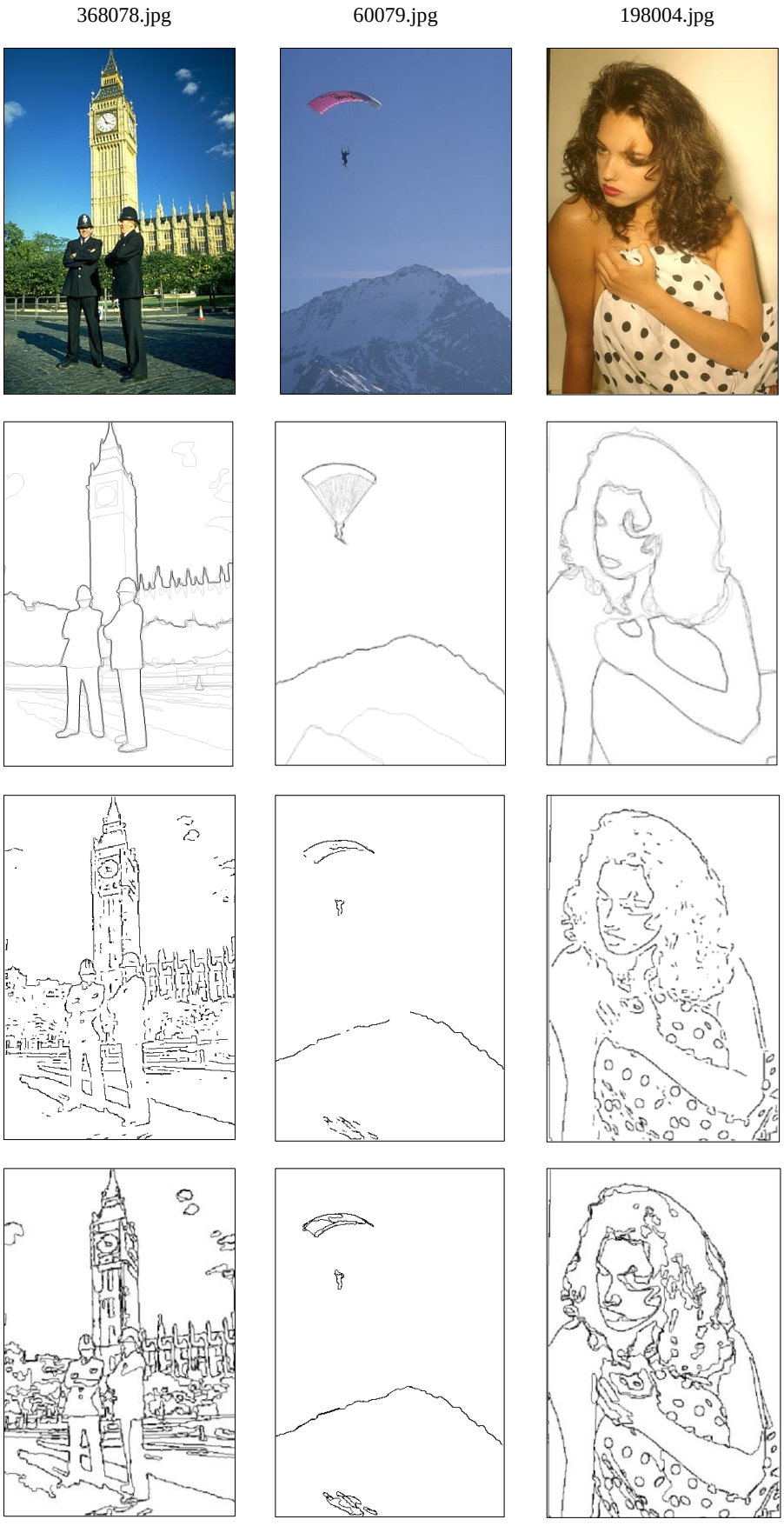


Figure 11 From top to bottom: original images, ground-truths, fast edge maps obtained using R= 2, T= 0.05, α=0.5, β=0.1 and the complete results.

36046.jpg 43051.jpg 384022.jpg

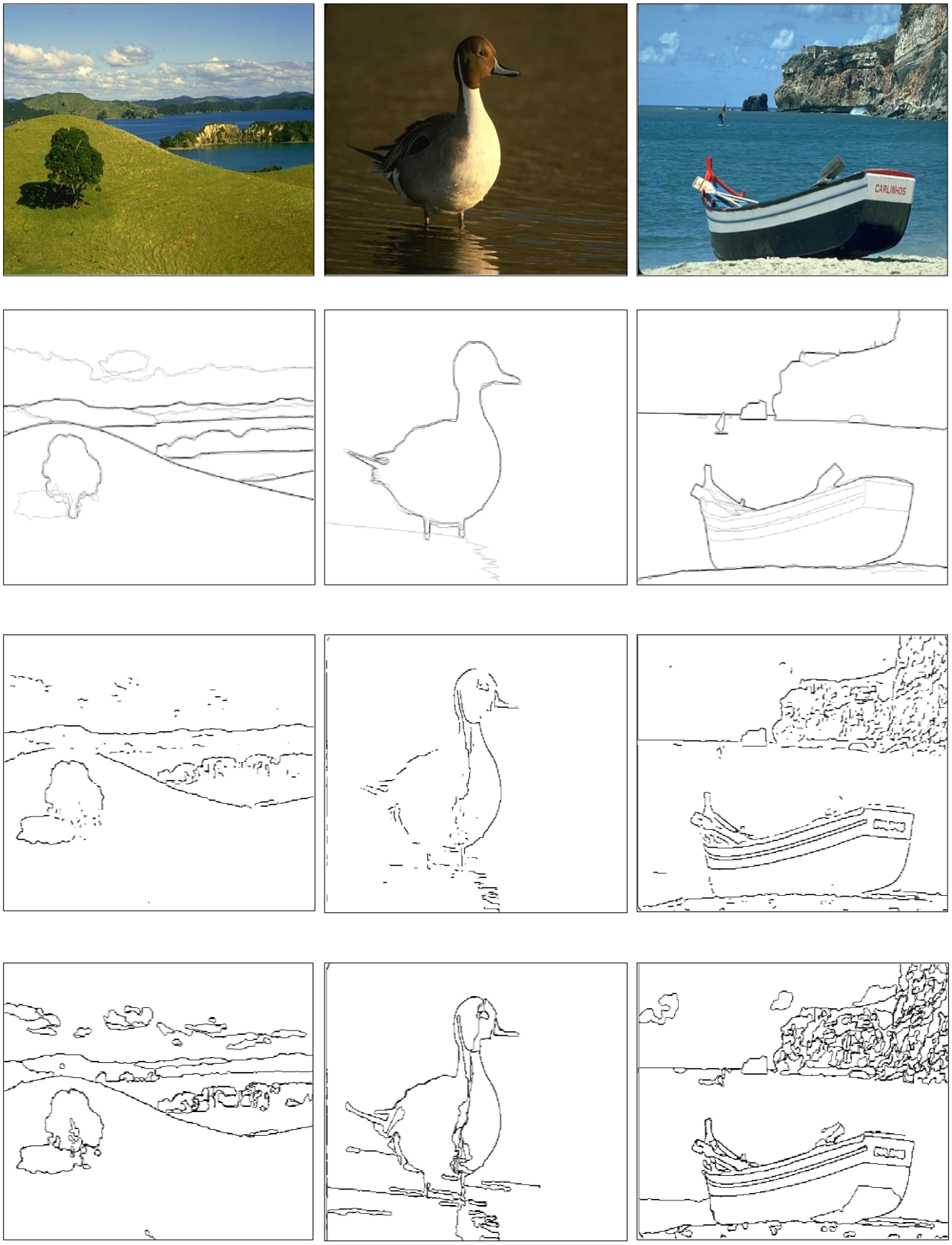


Figure 12 From top to bottom: original images, ground-truths, fast edge maps obtained using R= 2, T= 0.02, α= 0.5, β= 0.1 and the complete results.

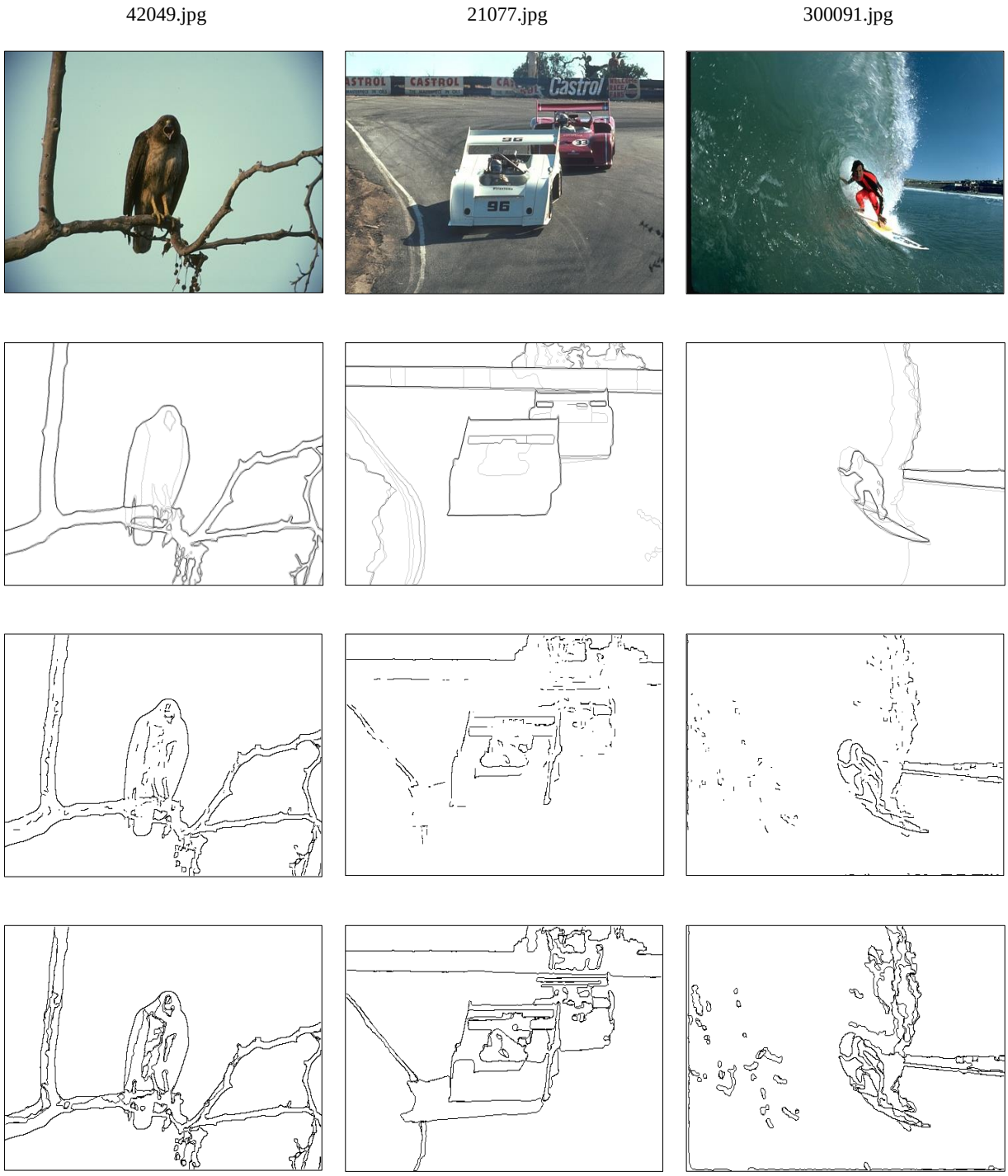


Figure 13 From top to bottom: original images, ground-truths, fast edge maps obtained using R= 2, T= 0.07, α= 0.5, β= 0.1 and the complete results.

4.2 Performance evaluation

For performance evaluation, 100 images from the BSDS500 dataset for segmentation algorithm ranking [27] are used. Figure 14 shows examples of image segmentation results along with F-measure (recall, precision) obtained using four boundary detectors and the proposed method. Three performance indicators [12]: the best F-measure on the data set for fixed scale (ODS), the aggregate F-measure on the dataset for the best scale in each image (OIS) and average precision (AP) of seven algorithms are provided in Table 1. The average computation times of these algorithms, performed using MATLAB codes running on a computer with an Intel Core i7 CPU (2.5GHz) with 8GB of RAM are given in Table 2. As seen, while the segmentation performance of the proposed algorithm is comparable to, or slightly better than, those of other methods, the obvious advantage of the proposed algorithm is that its computation time is much faster. Among the methods based on particle motion, the average computation time of the proposed algorithm is faster than those of PMVIF [5] and PMLCD [8] by factors of 1.93 and 18.93 respectively.

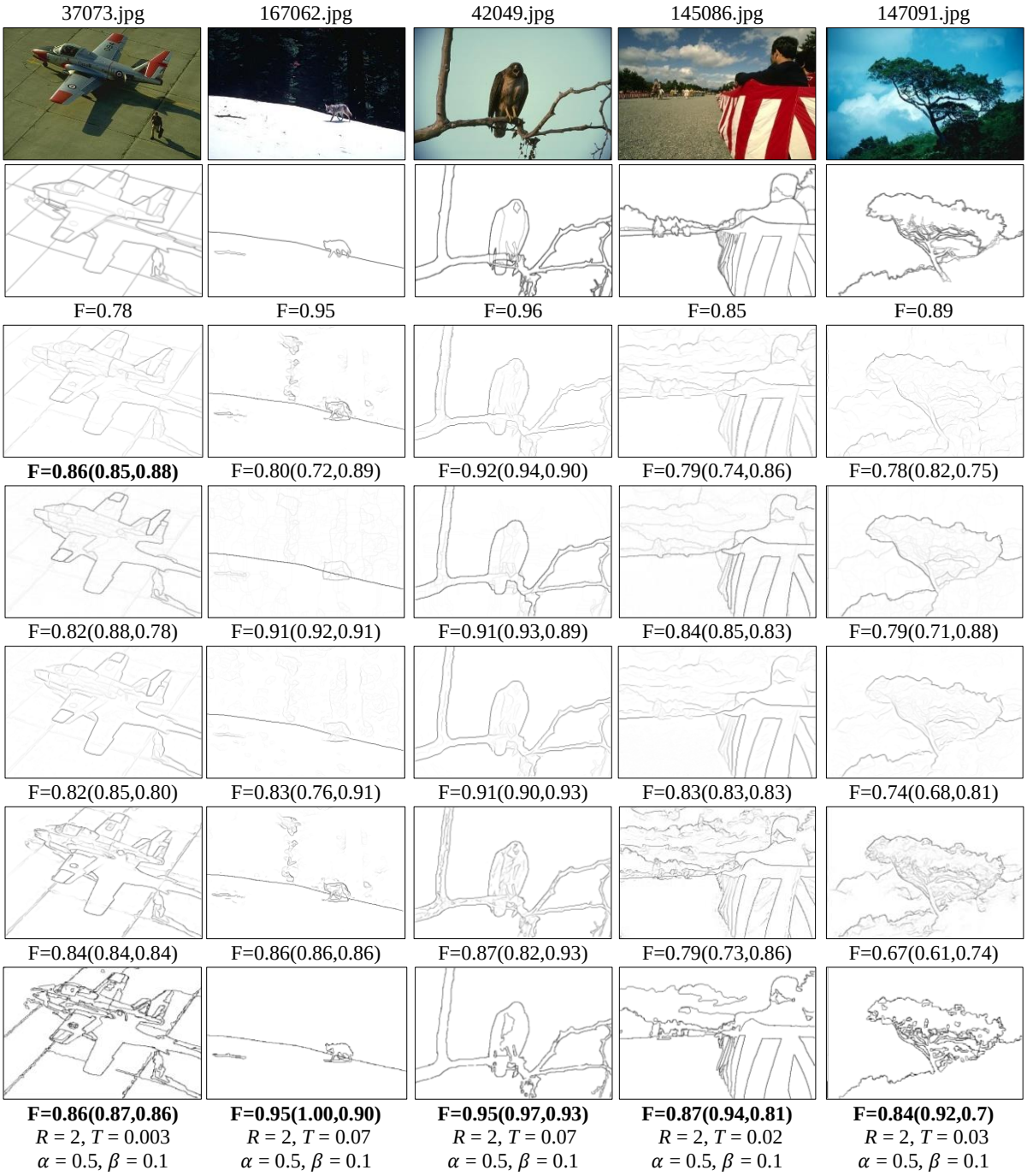


Figure 14 From top to bottom: examples of validating images, ground-truths, results obtained using Xiaofeng et al. [10], Arbelaez et al.[12], Maire et al. [26], Ren [28], and the proposed method.

Table 1 Performances of low-level boundary extraction algorithms evaluated using the BSDS500 dataset.

Methods	ODS	OIS	AP
Human	0.80	0.80	-
Canny [2]	0.60	0.63	0.58
Graph-Based [29]	0.61	0.64	0.56
Mean Shift [30]	0.64	0.68	0.56
NCuts [31]	0.64	0.68	0.45
PMVIF[5]	0.58	0.62	0.54
PMLCD [8]	0.61	0.64	0.55
Proposed method	0.61	0.64	0.60

Table 2 Average computational times of low-level boundary extraction algorithms tested using the BSDS500 dataset performed using MATLAB codes run on an Intel Core i7 CPU (2.5GHz) with 8GB of RAM.

Methods	Image type	Computational time (s)
Canny [2]	Grayscale	0.040816385
PMVIF[5]	image	0.078083054
Graph-Based [29]		10+
Mean Shift [30]	Color	3.141358
NCuts [31]	image	10+
PMLCD [8]		0.76450303
Proposed method		0.040385

5. Discussion

When looking into details in Table 2, both the graph-based [29] and NCuts [31] methods employ the time-consuming graph searching techniques that literally merge pixels that match the criteria into regions resulting in the longest computation times. Also, the mean shift algorithm [30] estimates the density function of pixels in the feature space by iteratively computing centers of clusters of pixel attributes using a kernel function as weights. Such an approach requires several rounds of computation to allow the centers of clusters to converge to actual means before the data can be taken into account resulting in a long computation time. Due to their long computation time, these methods are hence impractical for real-time implementation. On the other hand, edge tracking methods, i.e., Canny [2], PMVIF [5] and PMLCD [8], search for edge pixels only in the transition areas instead of searching pixels in the whole image and hence significantly reduce computation times. Compared to Canny and PMVIF algorithms, PMLCD and the proposed method suffer the high overhead preprocessing computation cost for computing local color distance images while Canny and PMVIF methods, considering an image as a monochrome image, need only gradient information which can be quickly estimated by using spatial filtering with appropriate kernels. Nevertheless, since edge tracking process, used in Canny, PMVIF and PMLCD, is a sequential process that requires prior knowledge of previously extracted edge pixel location, only one edge can be extracted at a time. In contrast, in the proposed method, finding adjacent pairs of pixels having negative values of dot products between the normal compressive vectors can be performed in parallel without requiring any prior knowledge about edge pixel locations. Thus, strong edge detection using the proposed method can be swiftly conducted resulting in less overall computation time.

6. Conclusion

The method presented in this paper utilizes the unique normal compressive property of NCVF to allow fast edge detection to be performed. The negative divergence of the normal compressive vector field is used as the indicator for screening areas of object edges. Strong edge pixels are later pinpointed by using the negative dot products between the normal compressive vectors at pairs of adjacent pixels which can be rapidly calculated by a raster scan manner. Most strong edges are obtained during the fast edge detection step. Only a fraction of weak edges, mostly at object corners, remain undetected resulting in discontinuous edges. Finally, the discontinuous edges are extended by using the PMLCD algorithm to achieve completely closed boundaries. Image segmentation experiments were conducted using the BSDS500 benchmark dataset. Segmentation performance was measured using the F-measures for ODS and OIS, average precision and computation time. It was shown that the proposed algorithm provided results comparable to those of well-known methods, while the average computation time is drastically reduced when compared to that of the particle motion based PMVIF and PMLCD methods. Moreover, the computation of NCVF and EVF, based on local color distance images proposed in the PMLCD [8], allows the method to be applicable to all types of images.

7. References

[1] Mutneja V. Methods of image edge detection: a review. J Electr Electron Syst. 2015;4(2):1000150.
[2] Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell. 1986;8(6):679-98.
[3] Cour T, Benezit F, Shi J. Spectral segmentation with multiscale graph decomposition. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); 2005 Jun 20-25; San Diego, USA. New York: IEEE; 2005. p. 1124-31.
[4] Eua-Anant N, Upda L. Boundary extraction algorithm based on particle motion in a vector image field. Proceedings of International Conference on Image Processing; 1997 Oct 26-29; Santa Barbara, USA. New York: IEEE; 1997. p. 732-5.
[5] Eua-Anant N, Udpa L. Boundary detection using simulation of particle motion in a vector image field. IEEE Trans Image Process. 1999;8(11):1560-71.
[6] Yang F, Cohen LD, Bruckstein AM. A model for automatically tracing object boundaries. 2017 IEEE International Conference on Image Processing (ICIP); 2017 Sep 17-20; Beijing, China. New York: IEEE; 2017. p. 2692-6.
[7] Lu C, Chi Z, Chen G, Feng D. Geometric analysis of particle motion in a vector image field. J Math Imag Vis. 2006;26(3): 301-7.
[8] Phornphatcharaphong W, Eua-Anant N. Edge-based color image segmentation using particle motion in a vector image field derived from local color distance images. J Imag. 2020;6(7):72.
[9] Dollar P, Zhuowen T, Belongie S. Supervised learning of edges and object boundaries. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06); 2006 Jun 17-22; New York, USA. New York: IEEE; 2006. p. 1964-71.
[10] Xiaofeng R, Bo L. Discriminatively trained sparse code gradients for contour detection. Adv Neural Inform Process Syst. 2012;1:584-92.

[11] Hallman S, Fowlkes CC. Oriented edge forests for boundary detection. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7-12; Boston, USA. New York: IEEE; 2015. p. 1732-40.
[12] Arbelaez P, Maire M, Fowlkes C, Malik J. Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell. 2011;33(5):898-916.
[13] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. IEEE Conf Comput Vis Pattern Recogn. 2015;1:3431-40.
[14] Shen W, Wang X, Wang Y, Bai X, Zhang Z. Deep contour: a deep convolutional feature learned by positive-sharing loss for contour detection. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7-12; Boston, USA. New York: IEEE; 2015. p. 3982-91.
[15] Bertasius G, Shi J, Torresani L. Deep edge: a multi-scale bifurcated deep network for top-down contour detection. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7-12; Boston, USA. New York: IEEE; 2015. p. 4380-9.
[16] Xie S, Tu Z. Holistically-nested edge detection. 2015 IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7-13; Santiago, Chile. New York: IEEE; 2015. p. 1395-403.
[17] Wang Y, Zhao X, Li Y, Huang K. Deep crisp boundaries: from boundaries to higher-level tasks. IEEE Trans Image Process. 2019;28(3):1285-98.
[18] Zou N, Xiang Z, Chen Y, Chen S, Qiao C. Boundary-aware CNN for semantic segmentation. IEEE Access. 2019;7:114520-8.
[19] Lee HS, Kim K. Simultaneous traffic sign detection and boundary estimation using convolutional neural network. IEEE Trans Intell Transport Syst. 2018;19(5):1652-63.
[20] Varghese R, Sharma S, Premalatha M. Transforming auto-encoder and decoder network for pediatric bone image segmentation using a state-of-the-art semantic segmentation network on bone radiographs. 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS); 2018 Oct 21-24; Bangkok, Thailand. New York: IEEE; 2018. p. 251-6.
[21] Kustner T, Muller S, Fischer M, Weiβ J, Nikolaou K, Bamberg F, et al. Semantic organ segmentation in 3D whole-body MR images. 2018 25th IEEE International Conference on Image Processing (ICIP); 2018 Oct 7-10; Athens, Greece. New York: IEEE; 2018. p. 3498-502.
[22] Tran T, Kwon O, Kwon K, Lee S, Kang K. Blood cell images segmentation using deep learning semantic segmentation. 2018 IEEE International Conference on Electronics and Communication Engineering (ICECE); 2018 Dec 10-12; Xi'an, China. New York: IEEE; 2018. p. 13-6.
[23] Nguyen N, Lee S. Robust boundary segmentation in medical images using a consecutive deep encoder-decoder network. IEEE Access. 2019;7:33795-808.
[24] Turan S, Bilgin G. Semantic nuclei segmentation with deep learning on breast pathology images. 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT); 2019 Apr 24-26; Istanbul, Turkey. New York: IEEE; 2019. p. 1-4.
[25] Pemasiri A, Ahmedt-Aristizabal D, Nguyen K, Sridharan S, Dionisio S, Fookes C. Semantic segmentation of hands in multimodal images: a region new-based CNN approach. 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019); 2019 Apr 8-11; Venice, Italy. New York: IEEE; 2019. p. 819-23.
[26] Maire M, Arbelaez P, Fowlkes C, Malik J. Using contours to detect and localize junctions in natural images. 2008 IEEE Conference on Computer Vision and Pattern Recognition; 2008 June 23-28; Anchorage, USA. New York: IEEE; 2008. p. 1-8.
[27] Berkeley.edu [Internet]. 2007 [updated 2007 June; cited 2021 Jan 9]. Available from: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/images.html.
[28] Ren X. Multi-scale improves boundary detection in natural images. 10th European Conference on Computer Vision: Part III; 2008 Oct 12-18; Marseille, France. Berlin: Springer; 2008. p. 533-45.
[29] Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. Int J Comput Vis. 2004;59:167-81.
[30] Comaniciu D, Meer P. Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell. 2002;24(5):603-19.
[31] Jianbo S, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell. 2000;22(8):888-905.