

An Adaptive and Robust Edge Detection Method Based on Edge Proportion Statistics

Yang Liu[✉], Zongwu Xie[✉], and Hong Liu[✉]

Abstract—Edge detection is one of the most fundamental operations in the field of image analysis and computer vision as a critical preprocessing step for high-level tasks. It is difficult to give a generic threshold that works well on all images as the image contents are totally different. This paper presents an adaptive, robust and effective edge detector for real-time applications. According to the 2D entropy, the images can be clarified into three groups, each attached with a reference percentage value based on the edge proportion statistics. Compared with the attached points along the gradient direction, anchor points were extracted with high probability to be edge pixels. Taking the segment direction into account, these points were then jointed into different edge segments, each of which was a clean, contiguous, 1-pixel wide chain of pixels. Experimental results indicate that the proposed edge detector outperforms the traditional edge following methods in terms of detection accuracy. Besides, the detection results can be used as the input information for post-processing applications in real-time.

Index Terms—Adaptive threshold, edge detection, edge proportion statistics, edge segment detection, one-pixel wide, real-time.

I. INTRODUCTION

EDGE detection is a very basic and important low-level image processing operation as a critical preprocessing step for high-level tasks such as object detection, image segmentation, feature extraction, 3D reconstruction, motion analysis and so on [1]–[5]. The edges provide important visual information since they correspond to major physical, photometrical or geometrical variations on the image.

A fast and accurate edge detector can significantly improve both performance and efficiency for the whole image processing system. However, finding a universally applicable detection rule is not easy, especially on real images.

As shown in Fig. 1, it is not always possible to get a suitable gradient threshold. A high threshold value may lead to the edge fractured and discontinuous. Whereas a low threshold value may lead to mixed pseudo edge instances rather than meaningful details. Once the scene or illumination changes, the threshold needs to be adjusted manually. Thus, how to realize the automatic edge detection has become an increasingly considerable issue accordingly [6].

Manuscript received April 19, 2019; revised October 23, 2019 and January 26, 2020; accepted March 4, 2020. Date of publication March 18, 2020; date of current version March 24, 2020. This work was supported in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 51521003 and in part by the Major Research Plan of the National Natural Science Foundation of China under Grant 91848202. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Charith Abhayaratne. (Corresponding author: Zongwu Xie.)

The authors are with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150000, China (e-mail: liuyanghit@hit.edu.cn; xiezongwu@hit.edu.cn; hong.liu@hit.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.2980170

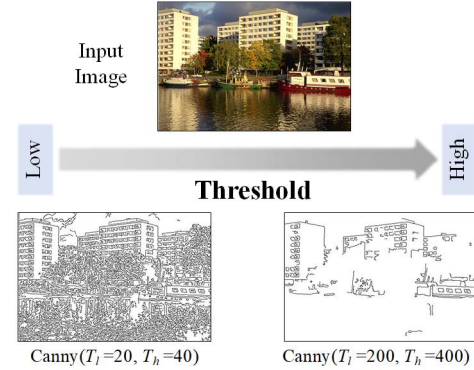


Fig. 1. Influence of the Gradient Threshold on Edge Detection.

Besides, most of the output results are binary maps where all the edge points are marked out. However, all the logical information is missing. That is, whether two given edge points come from the same object or not is unknown. In most cases, it is of great importance in the post-processing applications such as line detection, ellipse detection. Hence, how to organize edges and implement the principles effectively is still a big challenge.

A. Literature Review

Over the past 50 years, many approaches have been developed in the field of edge detection. Early detectors mainly focused on designing simple filters to detect pixels with highest gradients in their local neighborhood [7], e.g., Roberts, Sobel and Log. These detectors all based on local cues such as brightness, colors and gradients. Thus, they are easily influenced by the high-frequency noise. The Canny edge detector [8] goes a step further by employing non-maximum suppression to extract a thinner contour and applying the double threshold algorithm to get the weak edges. However, it is difficult to get a suitable threshold, especially in the dynamic environment where the background changes rapidly. Scotney and Coleman, [9] presented a design procedure for near-circular operators for image filtering. It significantly reduced angular error in comparison to other well-known gradient approximation methods.

To produce better detection results in general, many post-processing techniques have been proposed to extract edge segments from the image. A segment is a high-level feature which describes the boundary from one object to another. Edge Drawing (ED) [10] is a novel detector that extracts one-pixel wide and contiguous contours in real-time. It computes a set of anchors at first and then links them with the gradient direction. Edge Drawing Parameter Free (EDPF) [11]

is a real-time, parameter-free edge detection algorithm that validates the segments using the Helmholtz principle. Based on the Canny edge detector, CannySR [12] makes use of the smart routing step from ED to convert an edge map to a set of edge segments. Predictive Edge Linking (PEL) method [13] takes an arbitrary edge map as input and converts it to a set of edge segments with predictions from its past movements. HT-Canny[14] makes use of the gradient direction and Hough transform to connect the edge points. It is able to detect more low intensity edge and reflect the image details comprehensively. Baştan *et al.* [15] presented an edge detection and recovery framework based on open active contour models. The idea is to recover the missing pixels and joint the broken edges using the local continuity and smoothness cues. Furthermore, Flores-Vidal *et al.* [16] introduced a fuzzy segments clustering framework to differentiate the appropriate segments or true segments from the false ones.

The main problem of these filter-based methods is that they only look at the local differences between adjacent pixels but fail to provide high-level representations for natural images [17]. Martin *et al.* [18] formulated the task of cue combination as a supervised learning problem and modelled the true posterior probability of a boundary at every image location and orientation. Arbeláez *et al.* [19] combined local brightness, color and texture cues into a globalization framework using spectral clustering to boost the performance of edge detection. However, the globalization step significantly increases the computational load. Dollar and Zitnick [20] employed random decision forests to learn different local structures clustered by K-means for fast edge detection. Fu *et al.* [21] investigate automatic high-level feature construction for edge detection using genetic programming (GP) and Bayes' theorem. The evolved programs achieved a higher detection accuracy than the combination of local features in a simple Bayesian model. Benhamza and Seridi [22] proposed a new variant of ant colony optimization algorithm (ACO) to restore true broken image edges. It could detect more accurate and complete edges by diversifying the exploration and reinforcing the exploitation in the image.

Recently, synthetic aperture radar (SAR) image detection has become an important part in terms of edge detection. SAR is an active sensor that offers day/night imaging using microwave bands. Therefore, SAR images are widely used in applications such as object recognition, coastline extraction and so on [23]. According to the fourth-order normalized cumulant, a new energy named KWE [24] is proposed that can be used as an efficient feature for texture discrimination in SAR images. It avoids the manually selection of the initial contour and thus can be used in automatic processes. With fewer coefficients, a new parameter estimation technique (KCE) is presented in [25]. Based on the kurtosis of the curvelet coefficients, it produces better results than KWE as it extracts much more statistical data from the image.

In the past few years, deep convolutional neural networks (CNNs) have dominated the research on edge detection. CNNs are powerful visual models that yield hierarchical features, which can generate semantic meaningful contour and achieve remarkable performance. N^4 -Fields [26] is the

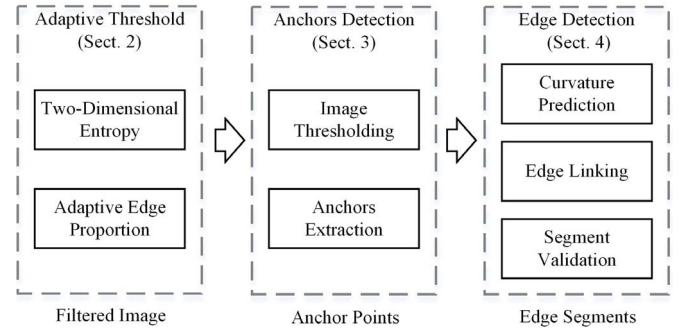


Fig. 2. Block Diagram of the Proposed Method.

first CNN-based edge detector that extracts features of image patches and uses the nearest neighbor to match extracted features to predefined features of edge patches. DeepEdge [27] is a cascaded edge detector which uses structured edge (SE) [28] to generate a feature vector for each contour point and then imports these vectors into two fully connected layers to produce a final boundary map. DeepContour [29] is also a patch-based approach which first divides one image into many patches and then redesigns the cost function for positive/negative samples to obtain more discriminative features. A hierarchical deep model is introduced in [30] to robustly fuse the edge representations. It is intended for contour detection, which produces more rich and complementary features than CNN models. Differing from these works, the HED [31] detector is an end-to-end fully convolutional neural network that takes an image as input and directly outputs the prediction. It is a multiscale network and combines multiple side outputs to fuse multi-level information. RCF [32] also utilizes the similar loss with HED and makes full use of semantic and fine detail features to carry out edge detection. CASENet [33] assigns each edge pixel to one or more semantic classes and extends the success in edge detection using an end-to-end system similar to HED and RCF. However, these works always generate much thicker contours than the classic methods [34]. Edges are blurry and do not stick to actual boundaries. LPCB [35] looks into the thickness issue of CNN-based architectures and presents an image-to-boundary prediction framework. The method can automatically learn rich hierarchical features, resolve ambiguity in prediction and predict crisp results without postprocessing. Even though, these CNN-based methods always call for large computational requirements, which weakens their applications on the embedded system or mobile devices.

B. Introduction to the Proposed Method

In this paper, an adaptive, robust and effective edge detector is proposed for real-time applications. The threshold parameters can be adjusted automatically according to the two-dimensional entropy. The detection result is a set of edge segments rather than a binary edge map, each of which is a clean, contiguous, 1-pixel wide chain of pixels. These edge segments can be used in many high-level applications such as line fitting, ellipse detection and image segmentation.

The proposed method is made up of three steps as detailed in Fig. 2. In the adaptive threshold step (Sect. 2), the 2D

image entropy is calculated to determine the reference edge proportion based on statistics. In the anchors detection step (Sect. 3), points with the biggest gradient are picked up as anchors. They are regarded as the initial candidates in the segment detection step. In the edge detection step (Sect. 4), the segment direction is taken into consideration rather than each individual gradient to predict the trend of the current segment. The edge linking process stops when the edge percentage reaches the reference value.

Compared with the other edge detectors, the proposed method has the following characteristics. Our method takes an adaptive threshold to detect edges according to the 2D image entropy. Without any prior experience, it can adapt to the environment automatically. Moreover, anchor points are jointed together with the segment direction rather than the gradient direction. On one hand, it follows a connective and smooth trajectory, on the other hand, it avoids the multi-pixel wide edges from the staircase pattern.

II. ADAPTIVE THRESHOLD

Limitations of global thresholds are typically due to poor quality of the source material, existence of multiple object classes of varying contrast and non-uniform illumination. In order to achieve the best effect of edge detection, it is significant to select different thresholds to match different images automatically.

To reduce noise and smooth away small edges, a Gaussian filter is applied on the input grayscale image. The kernel size is a user-defined parameter which balances the edge localization and noise rejection. In general, the 3×3 , 5×5 and 7×7 kernels with the standard sigma give better performance optimization than other kernels. For all results given in this paper, a standard 5×5 kernel is used [10].

A. 2D Entropy

Entropy was defined by Shannon [36] as a measure of uncertainty about an event associated with a given probability distribution. It can be served as a measure of disorder: the higher value the entropy becomes, the more information it contains. Compared with the 1D entropy model, the 2D entropy model not only considers the gray values of the image but also takes the spatial distribution into account [37].

Considering a digital gray image with the size of $M \times N$, let L be its gray level that equals to 256 in this paper. For each point, both the gray intensity I_i and the average value I_j from the neighborhood are calculated. The frequency of each gray pair (I_i, I_j) is computed then, which forms a 2D gray histogram as shown in Fig. 3.

A 11×11 box filter is used to average gray values from its neighboring pixels. The frequency of occurrence is stored in f_{ij} , which counts the number of pairs (I_i, I_j) in grayscale. The probability mass function (PMF) p_{ij} is defined with respect to the total size of pixels.

$$p_{ij} = f_{ij}/MN, \quad i, j = 0, 2, \dots, 255 \quad (1)$$

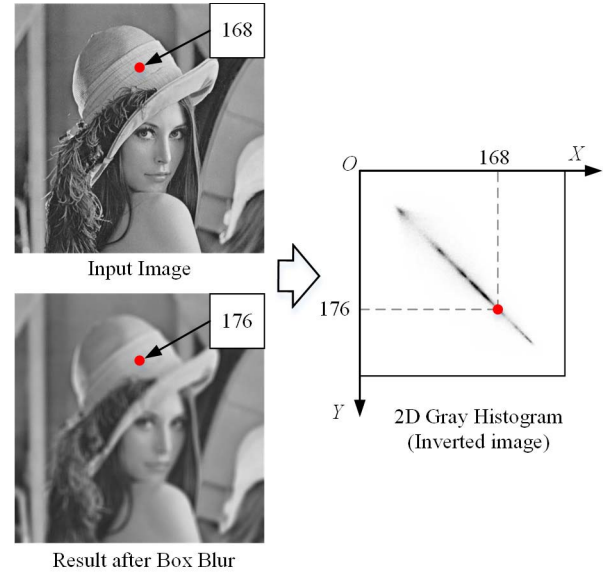


Fig. 3. 2D gray histogram of Lena. The horizontal axis is the gray-level value of pixel and the vertical axis is the average value from the neighborhood.

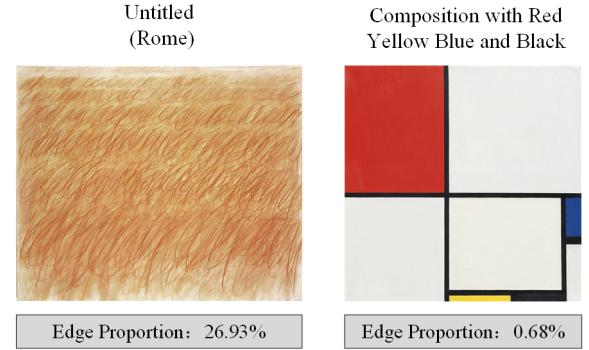


Fig. 4. Edge proportion in world famous paintings.

Then the 2D entropy is referred to

$$H = - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij} \log p_{ij} \quad (2)$$

B. Adaptive Edge Proportion

The edges provide important visual information since they correspond to stable features in terms of variations in object color and texture. It is also a critical feature to human eyes as we are easily attracted to the higher curvature points. In general, the proportion of edge pixels should be neither too big nor too small. As shown in Fig. 4, one image becomes disordered with the rise of the proportion. Whereas it contains little information when the proportion is small. Note that these two paintings are introduced to show the relationship between human perception and edge statistics. The proportions are set to extreme conditions which far from natural images.

An adaptive threshold method based on edge proportion statistics is proposed in this section. To this end, we first investigate into the Berkeley Segmentation Dataset and Benchmark (BSDS500) [19] and its precision-recall evaluation framework. It is a dataset widely used in the study of image segmentation and edge detection.

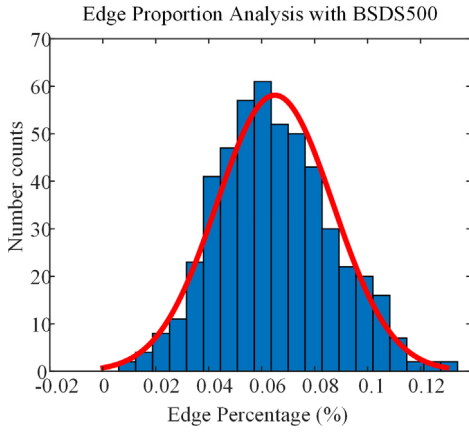


Fig. 5. Edge proportion statistics in BSDS500.

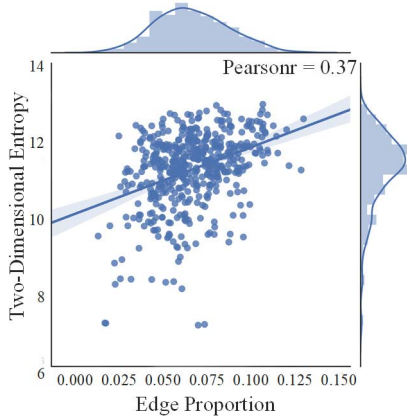


Fig. 6. Linear Correlation between Edge Proportion and 2D Entropy.

Using the human annotated ground truth boundaries, the edge percentage of each image can be figured out. From the 1D histogram over the whole dataset (Fig. 5), the proportion of the edge pixels trends to have a normal distribution, which can be referred to as $N(\mu, \sigma^2)$.

Here, the mean value is $\mu = 0.065$ and the variance is $\sigma = 0.022$. Thus, we can draw a conclusion that the edge proportion corresponds to a statistical principle despite the image diversity.

The image entropy specifies the uncertainty of the image and measures the averaged amount of information. Therefore, we assume it has something to do with the edge proportion. The linear relationship can be seen from Fig. 6. It is a multi-panel figure that shows both the bivariate relationship between the edge proportion and the 2D entropy along with the univariate distribution. The correlation coefficient indicates a positive linear relationship between these two variables: the larger the image entropy becomes, the more edge pixels the image contains.

To gain more insight, some examples are listed in Fig. 7, each attached with the edge percentage as well as the image entropy. On the basis of these results, it suggests that the edge proportion is associated with the image entropy. That is, the more information the image contains, the larger the entropy becomes as well as the number of edge pixels. Therefore, images can be clarified into three groups according to the image entropy, as illustrated in Table. I.

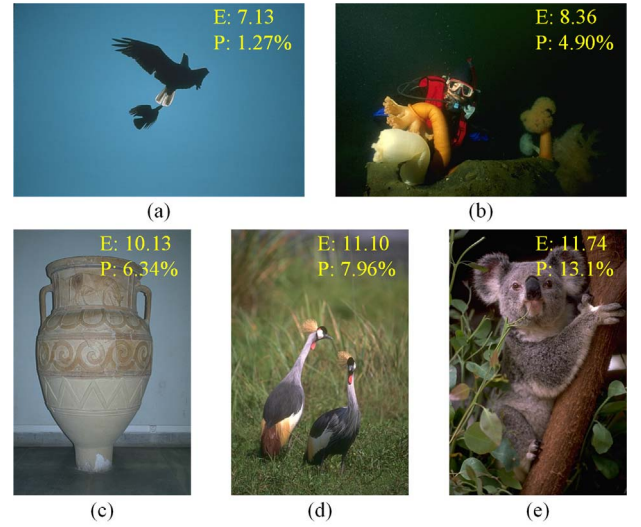


Fig. 7. Five Examples with Corresponding Parameters. (E: 2D Entropy; P: Edge Percentage).

TABLE I
EDGE PERCENTAGE DETERMINATION BASED ON THE IMAGE ENTROPY

	Low	Moderate	High
Two-Dimensional Entropy	[0, 10.8)	[10.8, 12.7]	(12.7, $+\infty$)
Recommended Edge Percentage	4.3%	6.5%	8.7%

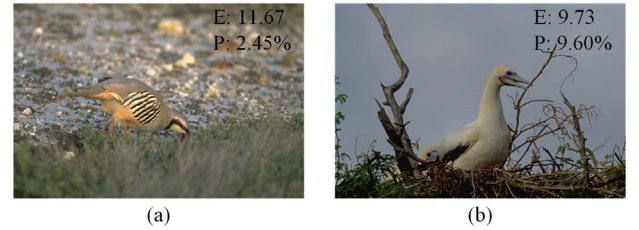


Fig. 8. Counter-Examples from BSDS500 with unexpected parameters.

In fact, it should be noted that the linear relationship between these two variables is not always true. Even though Fig. 8(a) has a larger entropy than Fig. 8(b), the edge proportion is far more less than the latter. It can be attributed to the diversity of the images. Edge structures, such as trees, grass, hairs, waves, blocks or similar anisotropic structures, will probably increase the image entropy. However, it is insignificant to extract edge contours from these geometric structures in most cases. Hence, we should decrease the reference value to get rid of non-meaningful edge pixels. On the contrary, the image entropy decreases with flat surface or smooth background. In this case, we should increase the edge percentage to get more details. In a word, an additional fine-tuning is recommended to obtain the ideal edge contours when necessary.

III. ANCHORS DETECTION

Anchors are the points whose gradient magnitudes are the biggest compared with the attached points along its gradient direction. They are peaks of the gradient map which have a high probability to be edge pixels.

TABLE II
GRADIENT DIRECTION DETERMINATION INSTEAD OF TRIGONOMETRIC FUNCTION

	$ g_x > a^1 \cdot g_y $		$ g_x < b^1 \cdot g_y $		$ g_x \leq a^1 \cdot g_y $ or $ g_x \geq b^1 \cdot g_y $			
	$g_x > 0$	$g_x \leq 0$	$g_y > 0$	$g_y \leq 0$	$g_x > 0$ and $g_y > 0$	$g_x > 0$ and $g_y \leq 0$	$g_x \leq 0$ and $g_y > 0$	$g_x \leq 0$ and $g_y \leq 0$
Direction	Right	Left	Down	Up	Down-Right	Up-Right	Down-Left	Up-Left

¹ $a = \tan 67.5^\circ$, ¹ $b = \tan 22.5^\circ$

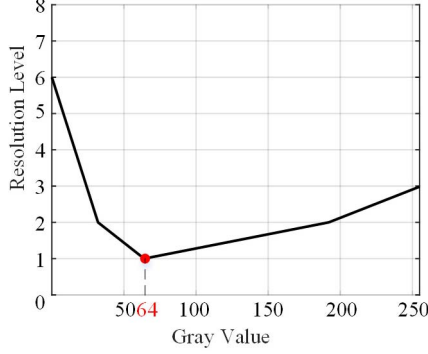


Fig. 9. Resolution of Human Eye for Gray.

In order to compute the gradient magnitude and direction, the Sobel operator is used in this section. It detects vertical and horizontal edges separately using two 3×3 convolution masks. These masks are used to perform convolution on the gray image and then obtain edge maps G_x and G_y in different directions.

A. Gradient Threshold Based on Human Resolution

First, the magnitude range is scaled to $[-255, 255]$. Then, the gradient edge map is computed by simply taking the average of the absolute values of G_x and G_y , instead of square operation.

$$G = (|G_x| + |G_y|)/2 \quad (3)$$

As it is well known, eye is the most important sense organ. Its unique transparent refractive mesenchymal that makes the retina become the only human organism which can be observed in a living body. Based on the human visual properties [38], the gray resolution function is analyzed in this step. It is a nonlinear transformation function as listed in (4).

$$R(g) = \begin{cases} -g/8 + 6 & 0 \leq g < 32 \\ -g/32 + 3 & 32 \leq g < 64 \\ g/128 + 0.5 & 64 \leq g < 192 \\ g/64 - 1 & 192 \leq g < 256 \end{cases} \quad (4)$$

According to Fig. 9, the resolution has a maximum value when the gradient tends to zero, which means we cannot tell a pixel whose intensity is 0 from 5. It drops to the minimum when the gray value approaches to 64, which means our eyes can hardly distinguish the difference between one magnitude in most cases. In this case, a threshold value $Th_r = 2$ is picked up to remove noise and smooth away edge pixels with small gradients.

B. Anchors Extraction

The anchors correspond to pixels where the gradient operator produces local maximal values. They are the peaks of

the gradient map where the edges will probably put over. For example, corners have a high possibility to be potential anchors as the intensity sharply changes around.

The gradient direction should be measured first with the edge maps G_x and G_y . Besides, it is also useful in the edge linking procedure (Sect. 4.2) that helps determine where a new segment starts. In fact, there are only eight directions (the first letter is capitalized for easy comprehension) to be figured out rather than a specific value. To speed up, several inequalities are introduced in Table II instead of trigonometric functions. Here, a and b are two temporary parameters to decide the gradient direction of each point.

Since an anchor must be a local peak of the gradient map, we simply compare one pixel with two attached points along its gradient direction. Take one pixel with “Right” direction for example, the left and right neighbor points are compared. If the current value is the biggest among these three points, it is marked as an anchor. Thus, edge pixels with unnecessary details can be filtered out regardless of how much sharp they are. Note that an anchor is just a possible location that an edge segment starts. Not all the anchors will be remained in the final edge map.

With these anchors, edge detection can be regarded as a boundary detection problem. It is much similar to the dot completion games, where a child is given a set of marked points in a picture and asked to connect the dots to reveal the hidden picture boundaries.

IV. EDGE SEGMENT DETECTION

In the edge following method, the connection among pixels is often considered rather than each independent point. In order to get high-quality edges, the edge detection result can also be described as a set of edge segments, each of which is a contiguous connected edge pixel. These obtained feature vectors can be used in detection, recognition and registration applications.

A. Curvature Predictive Method

As we have described above, the Canny algorithm makes use of double threshold to achieve image segmentation. The edge points can be divided into three classes: strong edge pixels, weak edge pixels and non-edge pixels depend on the gradients. The non-edge pixels are suppressed at first, then blob analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels. When there is one strong edge pixel in the blob, the weak edge point can be identified as one that should be preserved.

The ED edge detector, as well as the improved methods such as EDPF and CannySR, connects the anchors by smart routing to extract the edge segments. It makes use of the neighbor pixels' gradient directions and walks to the next anchor by

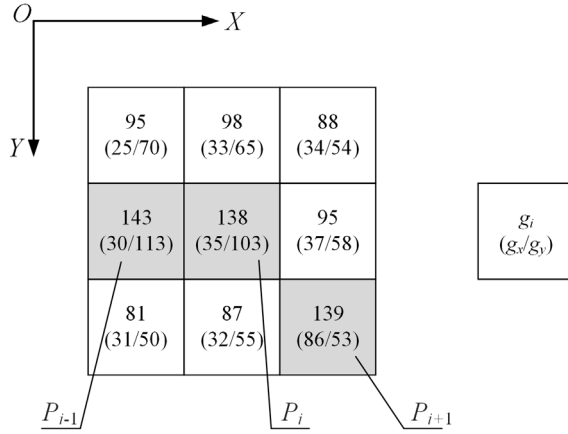


Fig. 10. Illustrations of the Curvature Predictive Method.

proceeding over the cordillera peak of the gradient map mountain. Nevertheless, the edge pixels are only divided into two groups (vertical/horizontal edge) based on G_x and G_y , which leads to some useless walks within the same edge segment. Moreover, the edge direction as well as the neighbor points is decided by its own gradients, which can be easily affected by the noise.

A property named segment direction is introduced in this step, which represents the trend of the segment. For example, if the next maximum gradient neighbor is on the left of the current one, its direction is left. Obviously, there are eight possible directions: up, up-right, right, down-right, down, down-left, left and up-left. Although they have the same names as listed in Table II, the definitions are totally different as the former is based on the horizontal and vertical gradients, and the latter is related to the connection of the neighbor pixels. It helps connect the edge points together with the trajectory history rather than the individual gradients.

The curvature predictive method is based on the hypothesis that an edge segment always has continuous curvature till to the sharp corners, where we will split the segment into different parts. In this case, we assume the segment direction of the current point is equal to the former one, then we check the next point along this direction as well as two linked neighbors. If the former segment direction is right, we will take the right, up-right and down-right pixels into consideration. After that, the pixel with largest gradient value is regarded as the next edge point. Then, the segment direction can be figured out as well.

According to Fig. 10, there are some edge pixels marked with the correct gradients. The lower values indicate the horizontal and the vertical gradients separately. The upper number is the final gradient comes from the absolute values below. P_{i-1} , P_i , and P_{i+1} are three neighbor points from one segment marked with gray in a close-up view. Take the middle point P_i for example, the segment direction is defined based on the curvature predictive method proposed in this paper. As the segment direction of the former point P_{i-1} is right, we suppose P_i has the same direction at first. Then, those three immediate neighbors on the third column are checked to get the maximum gradient. Thus, the segment direction changes to down-right at last. A new trial step then starts along this direction.

Algorithm 1 Edge Linking Method

Input: Anchor Points $P_i = \{P_1, P_2, \dots, P_m\}$, edge map
 1: Sort the anchor points by the gradient in descending order
 2: **for** $i = 1$ to m **do**
 3: Generate two short segments S_p and S_q with opposite directions based on the gradient direction from P_i
 4: **while** $EndFlag \neq \text{True}$ **do**
 5: Add the next edge point on S_p/S_q based on the Curvature Predictive Method
 6: **end while**
 7: **if** $length(S_p) + length(S_q) \geq Th_l$, **then**
 8: Joint S_p and S_q into one edge segment S_j
 9: **if** $EdgePointNum > Th_n$, **then**
 10: break
 11: **end if**
 12: **end if**
 13: **end for**
Output: Edge Segments $S = \{S_1, S_2, \dots, S_n\}$

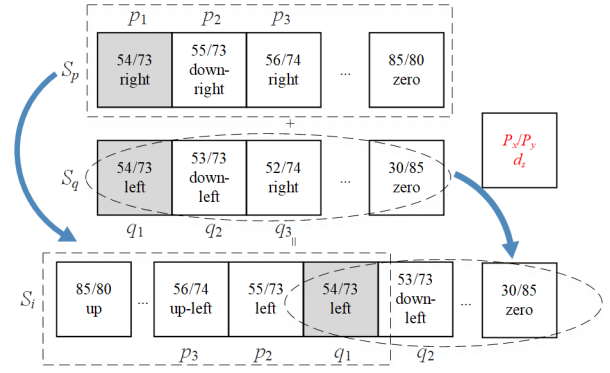


Fig. 11. Connection between Two Short Edge Segments.

B. Edge Linking With the Detected Anchors

In this section, we will connect the anchors together into consecutive edge segments. Each segment is a chain of pixels connected one another. The coordinates as well as the segment directions are collected for the post-processing applications.

First, the anchor points are sorted by the gradient values in descending order. Then, we start two opposite steps based on the gradient direction of each anchor point. If the gradient direction is down, we start a proceeding to the left/right, both perpendicular to the current gradient. They are also the initial segment directions where two short segments start. Next, each segment stretches with the curvature predictive method till it moves out of the edge areas. Once an edge pixel is detected, it will be removed from the edge map. Notice that the segment direction of the end point is indeterminate, which is referred to 0 in this paper. After that, these two short segments will be connected into a longer segment if they meet the length threshold Th_l . Edge segments shorter than eight pixels are considered as noisy artifacts and then removed in this paper [12]. Finally, the number of the edge points is counted to compare with the threshold Th_n , which is related to the recommended edge proportion in Sect. 2.2.

As illustrated by Fig. 11, S_p and S_q are two opposite edge segments that start from the same anchor point (54, 73). The upper values are the image coordinates of each edge point. The lower value shows the corresponding segment direction. The segment S_p is transferred into a reverse order

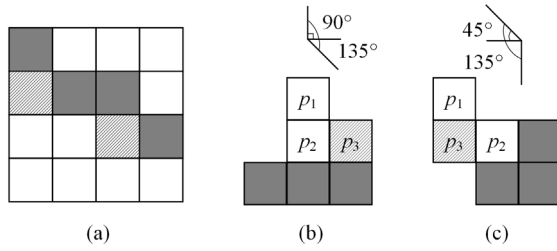


Fig. 12. Illustrations of the Multi-Pixel Wide Condition.

at first. Note that the segment direction of each point has been changed. In the forward case, the next point of p_2 is (56, 74), whereas it turns to (54, 73) in the inverse situation. As a result, the segment direction of p_2 changes to inverse direction of the former point p_1 . Furthermore, these two curves are connected into one longer segment with the point p_1 omitted.

Due to the quantization errors, traditional edge operators always generate multi-pixel edges in a staircase pattern as shown in Fig. 12(a). On the contrary, the detection result of our method is a set of edge segments, each of which is a clean, contiguous, 1-pixel wide chain of pixels. Based on the curvature predictive method, the angle between three consecutive points will be either 135° or 180° in this paper. However, the angle in a staircase pattern is 90° or 45° , which means there is no staircase structure in our result.

V. EXPERIMENTAL RESULTS

In this section, we test the performance of our algorithm by contrast with the state-of-the-art methods. All the following experiments are executing on a PC (I3-3240 at 3.4GHz, 4GB of Ram) with Win10.

A. Implementation Details

To evaluate the quality of the proposed method, we make use of the BSDS500 dataset once more. It is designed for evaluating natural edge detection that includes not only object contours but also background boundaries. The dataset contains 200 training, 100 validation and 200 testing images, each attached with human annotated ground truth results. Prior to evaluation, the marked boundaries must be thinned to one-pixel wide segments before running the benchmark code. In this case, the non-maximum suppression algorithm from [20] is adopted for that.

Besides, we also test our method on the NYUD dataset [39]. It is composed of 1449 RGB-D images with the same resolution of 480×640 . Unlike BSDS500, this dataset is a large depth benchmark for indoor scenes. Here we use the same 60%/40% training/testing split setting to make the comparison with the previous approaches.

To investigate the performance of proposed method, we then compare our method with several up-to-date edge detectors, i.e., OSTU-based [40], EDPF [11], PEL [13], SE [28] and N^4 -fields [26].

Selecting an appropriate threshold is one of the biggest problems in the study of edge detection. Assuming that the magnitudes from the meaningful edges are always greater than the non-meaningful ones, a gradient histogram is constructed

and the auto-thresholding algorithms can be then applied to determine the threshold parameters automatically. By iterating through all the possible threshold values and calculating the spread on both sides of the threshold, Otsu method can determine a suitable value using the interclass variance. Fang's work [40] is introduced then to make a fair comparison.

However, the detection result from OSTU-based method is a binary edge map that cannot decide whether two given edge points come from the same object or not. Considering that the main purpose of this paper is to extract edge segments for the post-processing applications, we then take the threshold values from OSTU as the input parameters of CannySR [12]. Using the smart routing step from ED [10], CannySR converts the edge map to a set of edge segments and fills in one-pixel gaps to clean up the low quality artifacts.

EDPF [11] and PEL [13] are two kinds of edge following methods. The detection results are all one-pixel wide and contiguous segments that hold the main frame of scene. EDPF works by running ED [10] with ED's parameters set at their extremes. The input image is first smoothed with a 5×5 Gaussian filter. In addition, it eliminates all pixels whose gradient values are smaller than 8.48 and sets the anchor threshold to zero. PEL first runs Canny to obtain edge maps with the low/high threshold value set to 20 and 40 respectively. Then it starts at an arbitrary point in the binary edge map and walks over the neighbor pixels until the end of a segment.

The above-mentioned edge detectors are all traditional filter-based methods that rely on the gradient magnitudes. Thus, these detectors are tested on the whole dataset directly without training or fine-tuning. Besides, we also present experimental comparisons with recent learning-based methods. SE [28] employs a random forest framework to learn different local structures clustered by K-means. The number of decision trees is set to 4 in this paper with a stride of 2 pixels. N^4 -fields [26] is a CNN-based method that combines the convolutional neural networks with the nearest neighbor search. The maximum number of comparisons in K-D trees is set to 30.

In contrast, our algorithm is a parameter-free edge detector with adaptive threshold values. Without any prior experience, our method has the ability to adapt to the environment according to the 2D entropy.

B. Performance Analysis

In order to quantitatively evaluate the performances of different edge detectors, the famous ODS F -measure metric is introduced as the harmonic mean of precision (P) and recall (R).

Results in Table III report the detection accuracy and the execution time of different edge detectors. In terms of running time, OSTU achieves a high speed which is much faster than the others. In fact, EDPF, PEL, as well as the proposed edge detector, can be regarded as the improvements of traditional binary edge map detection methods. These detectors focus on the segment extraction step that link all the edge pixels into different segments for further study (e.g., shape analysis). Considering that the detection process of OSTU

Image Name	Leopard	Building	Boat	Printing Shop	Bedroom	Living Room
Original Image						
Ground Truth						
OSTU	 F: 0.09	 F: 0.49	 F: 0.74	 F: 0.40	 F: 0.37	 F: 0.36
OSTU+SR	 F: 0.06	 F: 0.49	 F: 0.77	 F: 0.39	 F: 0.38	 F: 0.38
EDPF	 F: 0.76	 F: 0.48	 F: 0.75	 F: 0.40	 F: 0.38	 F: 0.35
PEL	 F: 0.27	 F: 0.46	 F: 0.67	 F: 0.33	 F: 0.39	 F: 0.32
SE	 F: 0.84	 F: 0.70	 F: 0.75	 F: 0.68	 F: 0.78	 F: 0.75
N^4 -field	 F: 0.82	 F: 0.74	 F: 0.73	 F: 0.71	 F: 0.79	 F: 0.77
Proposed	 F: 0.82	 F: 0.51	 F: 0.78	 F: 0.45	 F: 0.42	 F: 0.41

Fig. 13. Edge Detection Results on BSDS500 and NYUD.

terminates before edge linking, we cannot judge the speed simply on these results. Since the edge linking process is more time-consuming, CannySR is applied additionally for a fair comparison. In this case, PEL performs the best efficiency with respect to the edge following methods. Our method is in the second place due to the edge percentage determination step, where the 2D entropy is calculated in detail. On the contrary, the learning-based methods take an enormous amount of time, making them unsuitable for the real-time applications.

As to the detection accuracy, the learning-based methods present much better performance than the others. However, the detection results are a little blurry with multiple edges. Besides, these methods take longer times due to the large computational requirements. Apart from these learning-based methods, our method achieves the best performance on both datasets. Thus, it is a robust and practical edge detector for the real-time applications such as line detection, ellipse detection. On the one hand, it takes a little time with an adaptive threshold, on the other hand, it describes the image using a series of edge segments that are much more suitable

for the post-processing technique. In contrast, OSTU and PEL provide low F-measures even though they take small execution time. The main reason is that PEL takes a fixed threshold to detect edge maps and the low threshold of the OSTU comes from the higher one directly. As the edge contours are easily influenced by the environment, it is hard to find a proper threshold to adapt the whole dataset. EDPF also achieves a good detection accuracy. However, the edge validation procedure is a little time demanding.

The qualitative comparisons in Fig. 13 show consistent performance with those of the experiments on both datasets. The first three images are random samples from BSDS500 and the others are from NYUD. The test images include but not limited to: animals, buildings and sceneries. The first column is the input image and the second one lists the ground truth. Experimental results have demonstrated that learning-based methods have the capacity for automatically learn the high-level representations for natural images, which helps them generate semantic meaningful contour and achieve remarkable performance. However, the hardware requirement and the running time limit the application of these methods.

TABLE III
PERFORMANCE EVALUATION OF DIFFERENT EDGE DETECTORS

Dataset	BSDS500		NYUD	
	F-measure	Time (ms)	F-measure	Time (ms)
OSTU	0.437	3.17	0.335	4.95
OSTU+SR	0.440	11.72	0.338	14.24
EDPF	0.495	8.98	0.353	12.66
PEL	0.469	8.05	0.328	11.68
SE	0.743	436.46	0.751	285.31
N^4 -field	0.635	5.28×10^5	0.603	6.57×10^5
Proposed	0.614	8.17	0.427	11.73






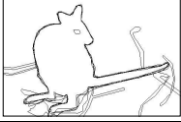


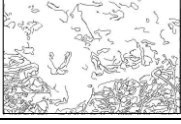
Name	Swan	Sofa	Kangaroo
Source Image			
Ground Truth			
Results			
Remarks	Spurious Edges	Missing Area	Meaningless

Fig. 14. Different Kinds of False Detections.

The detection result of PEL contains a lot of mixed pseudo edge instances due to the inappropriate threshold in the Canny edge process. Similarly, OSTU also suffers from the spurious and undesirable edge fragments from the irrelevant structures. In contrast, the detection results of our method are relatively clean and coherent with contiguous edge segments.

From the first image of Fig. 14, we can see that our method sometimes detects excessive edge pixels from the background. The main reason is that we take the upper limit of the recommended edge percentage rather than the average value. A larger percentage will result in more edge pixels while keeping the basic framework of the image as specific as possible. Considering that the detection results are the input information for post-processing applications, we would rather reserve more specific contours than seek for a clean background. Besides, some missing areas appear when there are large flat or smooth surfaces on the image. As we have discussed in Sect. 2.2, these objects will decrease the image entropy as well as the recommended edge percentage. The edge detection procedure terminates before it gets enough edge segments, leading to the blank areas on the detection result. In addition, the proposed method focuses on the local cues (e.g., colors, gradients) that are difficult to represent high-level information. According to the “Kangaroo” image, our method fails to retrieve the sketch of the animal. Therefore, we will try to solve these problems with the local information as well as the image content in the further work.

VI. CONCLUSION

In this paper, an adaptive and robust edge segment detector for real-time application was proposed. The edge proportion statistics was analyzed on BSDS500 at first, which led to a normal distribution as a result. Based on the 2D entropy, different images were classified into three classes, attached with a reference percentage value. According to the gradient direction, anchor points were picked up from the edge gradient map. Then, these anchors were jointed into different edge segments using the proposed curvature predictive method, each of which is a clean, contiguous, 1-pixel wide chain of pixels.

To evaluate the accuracy and efficiency of the proposed method, we have compared our detector with the state-of-the-art methods on the BSDS500 and NYUD datasets. In general, these results suggest that our method advances the other edge following methods in terms of accuracy. Based on the image entropy, the gradient threshold can be adjusted automatically. The edge pixels are stored into different segments for a better understanding of the image. A major limitation of our method is that it has a weak capability to learn the high-level representations compared with the learning-based methods. However, these learning-based methods always call for large computational requirements, which weakens the real-time performance on the embedded system or mobile devices.

To the best of our knowledge, this is the first time that edge proportion statistics is introduced to the edge detection techniques. Moreover, the “segment direction” is used to predict the next point on the segment instead of “gradient direction”, which helps joint the edge points together with the trajectory history rather than the individual gradients. As further improvements, our method can be used as a pre-processing step in many high-level applications such as line fitting, ellipse detection and image segmentation.

ACKNOWLEDGEMENT

The authors would especially like to thank Dr. Akinlar for providing us with the experimental executables as well as the insights. They would also like to thank the authors of [19] for sharing their database and the reviewers for their fruitful comments.

REFERENCES

- [1] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, “Multi-scale combinatorial grouping,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 328–335.
- [2] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille, “Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4545–4554.
- [3] Z. Zhang *et al.*, “Sequential optimization for efficient high-quality object proposal generation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1209–1223, May 2018.
- [4] M. Modava and G. Akbarzadeh, “Coastline extraction from SAR images using spatial fuzzy clustering and the active contour method,” *Int. J. Remote Sens.*, vol. 38, no. 2, pp. 355–370, Jan. 2017.
- [5] Y. Liu *et al.*, “DEL: Deep embedding learning for efficient image segmentation,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 864–870.
- [6] Y.-K. Huo, G. Wei, Y.-D. Zhang, and L.-N. Wu, “An adaptive threshold for the canny operator of edge detection,” in *Proc. Int. Conf. Image Anal. Signal Process.*, 2010, pp. 371–374.

- [7] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London. B, Biol. Sci.*, vol. 207, pp. 187–217, Feb. 1980.
- [8] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [9] B. W. Scotney and S. A. Coleman, "Improving angular error via systematically designed near-circular Gaussian-based feature extraction operators," *Pattern Recognit.*, vol. 40, no. 5, pp. 1451–1465, May 2007.
- [10] C. Topal and C. Akinlar, "Edge drawing: A combined real-time edge and segment detector," *J. Vis. Commun. Image Represent.*, vol. 23, no. 6, pp. 862–872, Aug. 2012.
- [11] C. Akinlar and C. Topal, "EDPF: A real-time parameter-free edge segment detector with a false detection control," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 26, no. 1, Feb. 2012, Art. no. 1255002.
- [12] C. Akinlar and E. Chome, "CannySR: Using smart routing of edge drawing to convert binary edge maps to edge segments," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Sep. 2015, pp. 1–6.
- [13] C. Akinlar and E. Chome, "PEL: A predictive edge linking algorithm," *J. Vis. Commun. Image Represent.*, vol. 36, pp. 159–171, Apr. 2016.
- [14] R. Song, Z. Zhang, and H. Liu, "Edge connection based canny edge detection algorithm," *Pattern Recognit. Image Anal.*, vol. 27, no. 4, pp. 740–747, Oct. 2017.
- [15] M. Baştan, S. S. Bukhari, and T. Breuel, "Active canny: Edge detection and recovery with open active contour models," *IET Image Process.*, vol. 11, no. 12, pp. 1325–1332, Dec. 2017.
- [16] P. A. Flores-Vidal, D. Gómez, P. Olaso, and C. Guada, "A new edge detection approach based on fuzzy segments clustering," in *Advances in Fuzzy Logic and Technology*, vol. 2. Cham, Switzerland: Springer, 2017, pp. 58–67.
- [17] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 7–27, 2001.
- [18] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.
- [19] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [20] P. Dollar and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1841–1848.
- [21] W. Fu, M. Zhang, and M. Johnston, "Bayesian genetic programming for edge detection," *Soft Comput.*, vol. 23, no. 12, pp. 4097–4112, Jun. 2019.
- [22] K. Benhamza and H. Seridi, "Canny edge detector improvement using an intelligent ants routing," *Evolving Syst.*, pp. 1–10, Aug. 2019.
- [23] M. Farbod, G. Akbarizadeh, A. Kosarian, and K. Rangzan, "Optimized fuzzy cellular automata for synthetic aperture radar image edge detection," *J. Electron. Imag.*, vol. 27, no. 1, p. 1, Feb. 2018.
- [24] G. Akbarizadeh, "A new statistical-based kurtosis wavelet energy feature for texture recognition of SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4358–4368, Nov. 2012.
- [25] Z. Tirandaz and G. Akbarizadeh, "A two-phase algorithm based on kurtosis curvelet energy and unsupervised spectral regression for segmentation of SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 3, pp. 1244–1264, Mar. 2016.
- [26] Y. Ganin and V. Lempitsky, " N^4 -fields: Neural network nearest neighbor fields for image transforms," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 536–551.
- [27] G. Bertasius, J. Shi, and L. Torresani, "DeepEdge: A multi-scale bifurcated deep network for top-down contour detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4380–4389.
- [28] P. Dollar and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1558–1570, Aug. 2015.
- [29] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3982–3991.
- [30] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe, "Learning deep structured multi-scale features using attention-gated CRFs for contour prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3962–3971.
- [31] S. Xie and Z. Tu, "Holistically-nested edge detection," *Int. J. Comput. Vis.*, vol. 125, nos. 1–3, pp. 3–18, Dec. 2017.
- [32] Y. Liu *et al.*, "Richer convolutional features for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1939–1946, Aug. 2019.
- [33] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "CASNet: Deep category-aware semantic edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5964–5973.
- [34] Y. Wang, X. Zhao, and K. Huang, "Deep crisp boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1724–1732.
- [35] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 11210, Oct. 2018, pp. 570–586.
- [36] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.
- [37] H. D. Cheng, Y. H. Chen, and X. H. Jiang, "Thresholding using two-dimensional histogram and fuzzy entropy principle," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 732–735, Apr. 2000.
- [38] Q. Jia, L. V. Xu-Liang, W. U. Chao, and H. C. Tang, "Research on infrared image enhancement based on human visual system," *Infr. Technol.*, vol. 32, no. 12, pp. 708–712, 2010.
- [39] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
- [40] M. Fang, G. Yue, and Q. Yu, "The study on an application of Otsu method in canny operator," in *Proc. Int. Symp. Inf. Process. (ISIP)*, 2009, p. 109.



Yang Liu was born in Handan, Hebei, China, in 1990. He received the B.S. and M.S. degrees from the Harbin Institute of Technology, Harbin, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include image processing, computer vision, pattern recognition, and robotics.



Zongwu Xie received the B.S. degree in electrical engineering and automation from the Harbin University of Science and Technology, Harbin, China, in 1996, and the M.S. and Ph.D. degrees in mechanical engineering from the Harbin Institute of Technology, Harbin, in 2000 and 2003, respectively.

He is currently a Professor with the Harbin Institute of Technology. His current research interests include the design and control of robotics. He has presided over the National Natural Science Foundation and the State Key Development Program of Basic Research of China (863 Program) more than 20 times.



Hong Liu was born in Hefei, Anhui, China, in 1966. He received the bachelor's and Ph.D. degrees from the Harbin Institute of Technology, Harbin, in 1986 and 1993, respectively.

In 1991, he joined DLR (the German Aerospace Center) as a joint Ph.D. candidate and worked on three-finger robot hand. Since 1999, he has been the Leader of the HIT-DLR Joint Robotics Laboratory, HIT. He has published more than 200 articles in robotics, mainly on robot sensing, sensory feedback, mechatronics, medical, and space robotics. He was

the Prime Investigator of first Chinese Space Robot Arm, which has been successfully tested in 2013.

Dr. Liu was awarded the Chang Jiang Scholars Program in 1999 and received the Professorship of the Harbin Institute of Technology (HIT). He received the First Prize of Technology Transfer Award, which is the most important EU-Wide Award for robotics research to raise the profile of technology transfer between science and industry, in 2007, the Second Prize of National Award of Technological Invention of China in 2008, and the Chinese Recruitment Program of Global Experts in 2010. Up to date, it is the most important recruitment program in China.