Edge Chain and Line Segment Detection by Applying Helmholtz Principle on Gradient Magnitude Map

Xiaohu Lu, *Student Member, IEEE,* Jian Yao†, *Member, IEEE,* Li Li, *Student Member, IEEE,* Yahui Liu, *Student Member, IEEE,* and Wei Zhang, *Member, IEEE*

Abstract—In this paper, we develop the Helmholtz principle on the gradient magnitude map of an image in the view of probability theory and apply it on the detection of edge chains and line segments. The traditional Helmholtz principle needs a good estimation of the number of different possible configurations for an event to calculate the "number of false alarms" (NFA) of this event, which is difficult in some applications. To overcome this shortage, we propose to use the "relative number of false alarms" (RNFA) instead of the traditional NFA to validate an event. Based on this observation, a simple and efficient edge chain detection algorithm is proposed, which detects edge chains via edge pixel growing first and then validates the edge chains according to their RNFAs. In this way, not only edge chains that are weak in contrast but meaningful in vision can be detected, but also the false alarms ratio is controlled in a low level. Extending this observation onto the line segment detection, we propose a novel line segment detector which fits for straight line segments directly on the gradient magnitude map instead of the edge chains which often suffer from the influence of noise. To evaluate the proposed edge chain and line segment detectors in quantity, both an edge chain detection benchmark with 25 semi-automatically labeled images and a line segment detection benchmark with 30 images were built. The proposed edge chain and line segment detectors were tested in these two benchmarks as well as some widely used datasets. The experimental results on these benchmarks sufficiently demonstrate that the proposed edge chain and line segment detectors outperform the state-of-the-art methods.

Index Terms—Helmholtz Principle, Edge Chain Detector, Line Segment Detector, Edge Chain Detection Benchmark, Line Segment Detection Benchmark.

1 Introduction

Geometric structure detection on an image is an important and classical problem in image processing and computer vision, which has been studied for decades. Edge chains and line segments are two most widely used kinds of geometric structures, which can be used to represent the silhouettes of an image. As a low level information of an image, edge chains can be applied in line segment detection [1], [2], object recognition [3] and image segmentation [4]. Besides, line segments can be used as low-level features to assist to solve some problems such as stereo matching [5], [6], indoor scene layout recovering [7], simultaneous localization and mapping (SLAM) [8], road extraction [9], crack detection in materials, image compression, and so on.

Based on the understanding that "edge is most often defined as an abrupt change in some low-level image feature such as brightness or color" [10], traditional edge detectors usually take two steps to extract edge segments: feature image extraction and feature image thresholding. Numerous edge detectors have been proposed in the past decades [11]–[14] based on this idea. The Canny operator [11] is a widely used edge detector which finds the peak gradient magnitudes orthogonal to the edge directions by applying a non-maximum suppression. However, it uses the gradient magnitudes as information, which makes it difficult to distinguish the faint edge pixels from the noise. Wang et al. [15] proposed the "supporting range" to distinguish those weak edge pixels from their surroundings and applied a segment-based hysteresis thresholding approach to verify the edge segments, which was tested to be very useful. Edge Drawing is a recently proposed edge detector which "computes a set of anchor edge points in an image and then links these anchor points by drawing edges between them" [16]. Edge Drawing is fast and uses more direction information than Canny on its novel edge linking process. In the work of [17], the use of the Helmholtz principle gives a new view on both boundary and edge detections. However, this work mainly focuses on the geometric event that: a strong contrast along a level line of an image, thus to some extent it can not be considered as a proper edge detector. "Conversely, the detection algorithm provides a check tool to accept or reject edges proposed by any other algorithm" [17]. In the work of EDPF [18], the original work of Edge Drawing was developed into a parameter-free edge detector by applying the Helmholtz principle on the validation check of the detected edge chains.

As to line segment detection methods, in general they can be divided into two categories: gradient-orientation-based and gradient-magnitude-based. The gradient-orientation-based methods only depend on the gradient orientations [19]. In the recently proposed line segment detector LSD [20], a line segment is defined as a

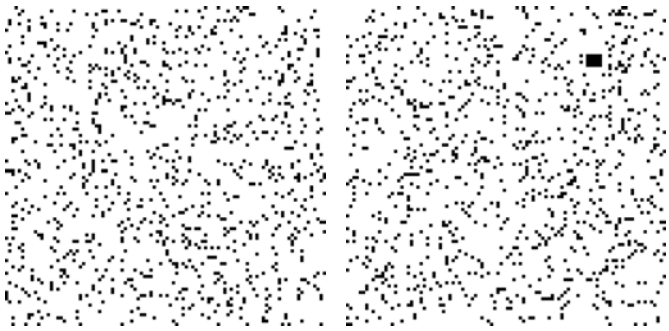
- *Xiaohu Lu, Jian Yao† (corresponding author), Li Li and Yahui Liu are with Computer Vision and Remote Sensing (CVRS) Lab, School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, China. †E-mail: jian.yao@whu.edu.cn Web: http://cvrs.whu.edu.cn/*
- *Wei Zhang is with School of Control Science and Engineering, Shandong University, Jinan, Shandong, China.*

straight image region whose points share roughly the same image gradient orientation, and the detected line segments are validated according to the Helmholtz principle. The gradient-magnitude-based methods first apply an edge detector to extract the edge map from the input image and then detect line segments based on the extracted edge map. Hough transform (HT) [21] is a traditional line detector based on an edge map but it usually extracts infinitely long lines instead of line segments and easily cause many false detections in richly-textured regions with strong edges. Recently, Akinlar and Topal [1] proposed a robust and efficient line segment detector, named as EDLines, to extract line segments from edge segments, which consists of three main steps: 1) extracting the edge segments by the Edge Drawing (ED) algorithm [16]; 2) extracting line segments from the edge segments based on the least-square line fitting method; 3) eliminating false line segments according to the Helmholtz principle.

The Helmholtz principle is popularly applied in the detection of image structures like line segments [1], [20], [22], edges and boundaries [17], [18], continuous curves [23] and vanishing points [24]. The Helmholtz principle does not use an a priori or learned model, but applies the *a contrario* uniform random assumption. The *a contrario* assumption is based on a certain background model. For example, in the line segment detection [20], the background model is the image's gradient orientation map, and the assumption is that the gradient orientations of pixels are independent and uniformly distributed in the range $(-\pi, \pi]$ on the gradient orientation map. The LSD detector [20] first applies a region growing method to obtain a line-support region, and then a rectangle is fitted as an approximation of the region, finally the Helmholtz principle is applied to validate the meaningfulness of this region by calculating the "number of false alarms" (NFA) of this region according to the number of aligned orientations in it. In the edge and boundary detection work of [17], the background model is the level lines of an image, and the assumption is that the contrast (gradient magnitude) at a point on any level line is mutually independent.

Despite that the Helmholtz principle is well studied and applied in both the gradient orientation map and the level lines, its application on the gradient magnitude map is still not well discussed yet. The main reason is that the value of N_{conf} [25], which is one of the key factors to calculate the value of NFA, is hard to be determined for the application like the edge chain detection. In this contribution, we first develop the Helmholtz principle on the gradient magnitude map of an image in the view of probability theory, and propose to use the "relative number of false alarms" (RNFA) instead of the traditional NFA to validate an event. Both an edge chain detector and a line segment detector are proposed, which detect edge chains and line segments directly on the gradient magnitude map, respectively. To evaluate the proposed edge chain and line segment detectors in quantity, we also built both an edge chain detection benchmark with 25 labeled images and a line segment detection benchmark with 30 labeled images.

The remainder of this paper is organized as follows. The proposed Helmholtz principle on the gradient magnitude map for validating edge chains and line segments is introduced in Section 2. A simple and efficient edge chain detector is described in Section 3. A line segment detector based on the gradient magnitude map instead of the edge chains is proposed in Section 4. Experimental results on our built benchmarks as well as other two widely used databases used to evaluate our proposed edge chain and line segment detectors are presented in Section 5 followed by the conclusions drawn in Section 6.



(a) randomly distributed (b) a rectangle structure

Fig. 1. Example images with the size of 100×100 pixels: (a) 1000 independent black pixels randomly distributed on the image; (b) an image with a 4×5 black rectangle structure.

2 Helmholtz Principle on Gradient Magnitude Map

2.1 Helmholtz Principle

Fig. 1 shows two 100×100 simulated images with 1000 black pixels distributed on each of them. In the image shown in Fig. 1(a) all the black pixels are randomly distributed, which means that the probability of a pixel to be black is $1000/(100 \times 100) = 1/10$, while in that of Fig. 1(b), there is a 4×5 black rectangle structure, the resting 980 black pixels are also randomly distributed, thus the probability of a pixel on Fig. 1(b) to be black is approximate to $1/10$. Comparing to the image shown in Fig. 1(a), we will sense that such a rectangle structure shown in Fig. 1(b) could not be arose just by chance. But how to measure this sense in quantity? The computational Gestalt theory and Helmholtz principle [17], [25] give a systematic solution. Before we have a deep look in the Helmholtz principle on the detection of image structures, some basic concepts are introduced as follows:

- *event*: a geometric structure on an image, for example the black rectangle shown in Fig. 1(b).
- *object*: the basic element to form an event, for example the pixels in the black rectangle shown in Fig. 1(b).
- *quality*: a common character that is shared by all the objects of an event, for example the quality that all the pixels in the black rectangle shown in Fig. 1(b) are all "in black".

In the Helmholtz principle, the sense "a structure could not arise just by chance" is defined as the expectation of the number of occurrences of this structure (event) under the *a contrario* uniform random assumption, which is also known as the "number of false alarms" (NFA). According to the

Helmholtz principle, an event is meaningful if the NFA of this event is very small. The NFA is formulated as follows:

$$\text{NFA} = N_{\text{conf}} \times \mathcal{B}(n, k, p), \tag{1}$$

where N_{conf} denotes the number of different possible configurations one could have for the searched event, which means that there probably are N_{conf} events in theory on the image, p represents the probability that a given object has a considered quality, and

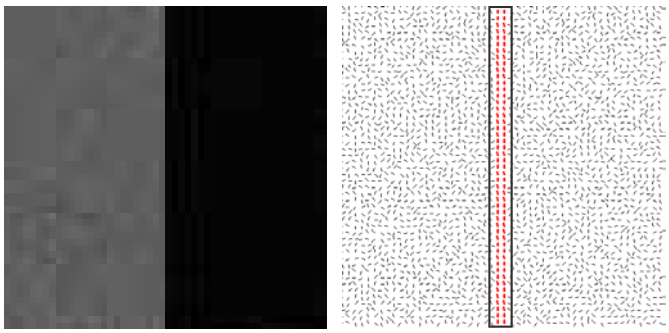
$$\mathcal{B}(n, k, p) = \sum_{i=k}^{n} \binom{n}{k} p^i (1-p)^{n-i} \tag{2}$$

is the tail of the binomial distribution which means the probability that at least k objects out of the observed n ones have this quality under the independence assumption.

As a summary, there are three key factors in the Helmholtz principle: (1) the perspective meaningful event; (2) the theoretical number N_{conf} of the event on the image; (3) the probability p of the considered quality. Take the two images in Fig. 1 for example, in Fig. 1(a) there is no perspective meaningful structure (event) observed, while in Fig. 1(b) the black rectangle is sensed as a meaningful structure. Assume that we have already obtained the black rectangle on the image shown in Fig. 1(b) by some detection methods, the rest of the problem is how to calculate the N_{conf} and $\mathcal{B}(n, k, p)$. In the case of the black rectangle, the event now is "a square made up of black pixels", the object is "pixel" and the quality is "pixel in black". The definition of an event gives the estimation of N_{conf}, considering the fact that a rectangle is determined by the left-top and right-bottom vertexes, each vertex can be any pixel on the 100×100 image region, thus an approximation of N_{conf} is $(100 \times 100)^2$. The probability of a 4×5 black rectangle is $\mathcal{B}(20, 20, \frac{1}{10}) = (\frac{1}{10})^{20}$, thus the NFA of the black rectangle is $(100 \times 100)^2 \times (\frac{1}{10})^{20} = 10^{-12}$, which is a really small value and means that this square can hardly occur in a background model where the black pixels are randomly and independently distributed with a probability of $1/10$, so according to the Helmholtz principle the 4×5 black rectangle on the image shown in Fig. 1(b) is perspective meaningful.

2.2 Helmholtz Principle on Orientation Map and Level Lines

The gradient orientation of a pixel is distributed in the range $(-\pi, \pi]$ and the pixels on a line segment share a close orientation. Such properties make it very straight forward to apply the Helmholtz principle on line segment detection from the gradient orientation map. In the LSD line segment detector [20], a line segment is defined as a straight image region whose points share roughly the same gradient orientation, so the event now is "a straight image region whose pixels are aligned with it" and the quality is "a pixel is aligned with the straight region". Given an angle tolerance $p\pi$, the probability that a pixel in the support range of a line is aligned with this line is p. Considering the fact that a straight region is determined by two terminal points and its width, there are $w \times h$ possible positions for each terminal point and $(w \times h)^{0.5}$ possible values for the width of the line segment, so the N_{conf} of this event is $(w \times h)^{2.5}$, where $w \times h$ is the size of the image. Thus, for a straight region with n pixels, among which k ones are aligned with this region, the NFA of this straight region is: $(w \times h)^{2.5} \times \mathcal{B}(n, k, p)$. Fig. 2 shows an orientation gradient magnitude map of a 50×50 image, the straight region marked in rectangle on Fig. 2(b) contains 100 aligned pixels. According to LSD, the angle tolerance is $\pi/8$, so the NFA of the red straight region is $(50 \times 50)^{2.5} \times \mathcal{B}(100, 100, 1/8) \approx 1.5 \times 10^{-82}$, which is a quite small value, thus we can assert that a very meaningful line segment is found.

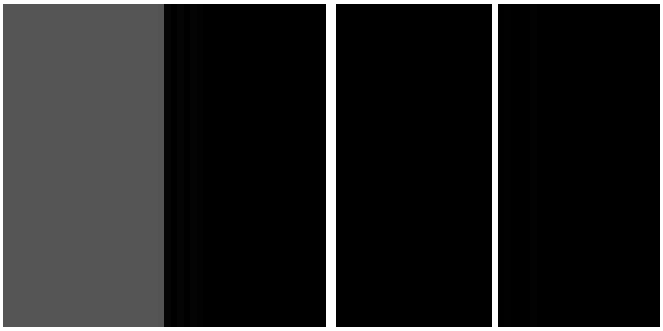


(a) gray image (b) gradient orientation map

Fig. 2. An example of the gradient orientation map and the aligned pixels of a line segment on a 50×50 gray image.

Fig. 3. An example of level lines on the Lena image with the level step of 20 where different colors of contours stand for different level values.

As to apply the Helmholtz principle on edge chain detection, the first problem we encounter is the definition of an edge chain event, because an edge chain can be anywhere with any shape and any length on the image, which makes it difficult to be expressed in a certain model. To solve this problem, in the work of [17] the "level line", as Fig. 3 shows, was introduced, and an edge chain is defined as "a piece of level line along which the contrast of the image is strong". Considering the fact that it needs two terminals to



(a) gray image (b) gradient magnitude map

Fig. 4. A 50×50 binary image with only one line segment and the corresponding gradient magnitude map.

determine an edge chain, so for a level line E_i with a length of l_i there are totally $l_i(l_i-1)/2$ possible edge chains on E_i, thus the N_{conf} of the edge chain event on an image can be formulated as:

$$N_{\text{conf}} = \sum_i l_i(l_i-1)/2, \tag{3}$$

where l_i is the pixel number of the i-th level line. The considered quality now is "each pixel of a level line E has a contrast equal or greater than u", and the probability of this quality is formulated as:

$$H(u) = \frac{1}{M}\#\{\mathbf{x} \in \mathbf{I} | g(\mathbf{x}) \geq u\}, \tag{4}$$

where $\mathbf{I}$ is the image, $g(\mathbf{x})$ denotes the contrast (gradient magnitude) of a pixel $\mathbf{x}$ and M is the number of pixels whose gradient magnitudes are not equal to zero on the image, i.e., $M = \#\{\mathbf{x} \in \mathbf{I} | g(\mathbf{x}) \neq 0\}$. So to apply the Helmholtz principle on the edge chain detection based on the level lines, first of all we should get the level lines of the image, then for an edge chain with l pixels, the smallest gradient magnitude u on this chain is found, and finally the NFA of this edge chain is defined as $N_{\text{conf}} \times H(u)^l$.

We can see that applying the Helmholtz principle on the edge chain detection is not a straight forward work because the level lines should be obtained in advance. In the work of EDPF [18], the level lines are replaced with edge chains obtained by the Edge Drawing method for convenience without convincing proofs. In fact, both the level lines based and the edge chains based N_{conf}s are just approximations of the exact value of N_{conf}, and it is still difficult to find a convincing method to calculate the value of N_{conf} for the edge chain event.

2.3 Helmholtz Principle on Gradient Magnitude Map

Let $\mathbf{I}$ be $w \times h$ image and $\mathbf{G}$ be the gradient magnitude map of $\mathbf{I}$ by applying a gradient operator on $\mathbf{I}$. In our entire work the 3×3 Sobel operator was applied, and the gradient magnitude $g(\mathbf{p})$ of a pixel $\mathbf{p}$ in $\mathbf{I}$ is calculated as follows:

$$g(\mathbf{p}) = \sqrt{(g_x(\mathbf{p}))^2 + (g_y(\mathbf{p}))^2}, \tag{5}$$

where $g_x(\mathbf{p})$ and $g_y(\mathbf{p})$ represent the gradients of the pixel $\mathbf{p}$ in $\mathbf{I}$ in the horizontal and vertical directions, respectively.

For each integral gradient magnitude level $u \in [1, g_{\max}]$ where $g_{\max}$ is the maximum gradient magnitude level in $\mathbf{G}$, the number of pixels whose gradient magnitude level is equal or greater than u is denoted as $k(u)$, thus the probability of the considered quality that "a pixel on $\mathbf{I}$ whose gradient magnitude level is equal or greater than u" is defined as:

$$P(u) = k(u)/M, \tag{6}$$

where $M = w \times h$ is the size of $\mathbf{I}$. This definition is similar to that of Eq. (4) in form, but different in one of the basic conceptions of the Helmholtz principle. As we have stated before, the quality is a common character that is shared by all the objects of an event, and the probability of the quality represents the distribution of the background model. By setting M as the number of pixels whose gradient magnitudes are greater than zero on the image, Eq. (4) implies that the background model is the level lines of an image, while Eq. (6) means that the objects with this quality is distributed randomly on the whole gradient magnitude map. Fig. 4(a) shows a 50×50 binary gray image with one line segment being the boundary between the light and black half regions and Fig. 4(b) illustrates the corresponding gradient magnitude map with only two values $\{0, 200\}$. According to Eq. (4), the probability $H(200) = 1$, so the NFA of the line segment is greater than 1 and thus there is no meaningful line segment on the image. While by the definition of Eq. (6), the probability $P(200) = 1/50$ and thus the line segment is a very meaningful event.

Fig. 5 shows two edge chains marked in rectangular frames on Fig. 5(a). The weak one shown in Fig. 5(b) has a minimum gradient magnitude $u = 24$ and $P(u) = 0.43$, and thus its background is dense but we can still distinguish the edge chain from the background, which means that the edge chain is meaningful. The strong one shown in Fig. 5(c) has a minimum gradient magnitude $u = 360$ and $P(u) = 0.003$, and so its background is much sparser and obviously it is a meaningful edge chain.

Definition of NFA - *Number of False Alarms*. Given an event $\mathcal{E}$ (a detected structure) made up of l pixels on an image, N_{conf} is the theoretical number of $\mathcal{E}$ on the image, u is the minimal gradient magnitude of these pixels, the NFA of $\mathcal{E}$ on the gradient magnitude map is defined as:

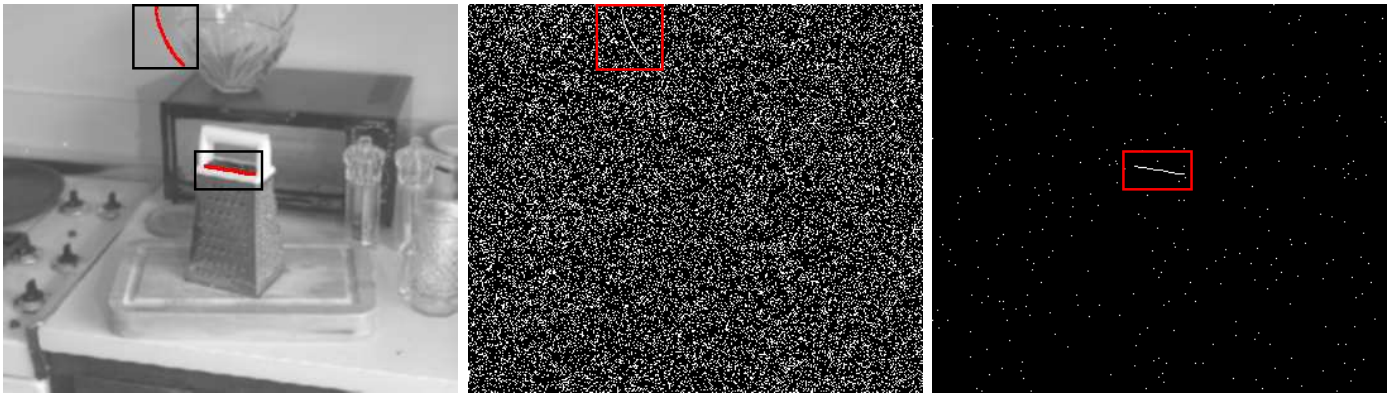
$$\text{NFA} = N_{\text{conf}} \times P(u)^l. \tag{7}$$

In some applications, it is difficult to give a good approximation of the N_{conf}, for example the value of N_{conf} for edge chain is hard to be obtained as we have discussed in Section 2.2. In this case, we propose to use the "relative number of false alarms" (RNFA) to validate edge chains.

Definition of RNFA - *Relative Number of False Alarms*. Given an event $\mathcal{E}$ (a detected structure) whose binomial probability is $\mathcal{B}(n,k,p)$, and $\mathcal{E}_r$ is a minimal meaningful event (MME) whose binomial probability is $\mathcal{B}(n_r,k_r,p_r)$. The relative number of false alarms of $\mathcal{E}$ to $\mathcal{E}_r$ is defined as:

$$\text{RNFA} = \frac{N_{\text{conf}} \times \mathcal{B}(n,k,p)}{N_{\text{conf}} \times \mathcal{B}(n_r,k_r,p_r)} = \frac{\mathcal{B}(n,k,p)}{\mathcal{B}(n_r,k_r,p_r)}, \tag{8}$$

where N_{conf} is the number of different possible configurations one could have for the searched event, and we simply say that the event is meaningful than the minimal meaningful event (MME) if RNFA < 1. As we can see from Eq. (8) that, all configurations of a given type of event on



(a) gray image (b) $u = 24$, $P(u) = 0.43$ (c) $u = 360$, $P(u) = 0.003$

Fig. 5. Two examples of edge chain structures on gradient magnitude maps with two different values of u.

the image share the same value of N_{conf}, which means that the exact value of N_{conf} can be eliminated if a reference case can be found. In this way, the problem of finding a good approximation of N_{conf} is converted into the searching for the minimal meaningful event (MME), which can be very simple in some cases.

3 Edge Chain Detection

3.1 Edge Chain Validation

We consider using the proposed RNFA to validate the meaningfulness of the edge chains. In many cases, it is difficult to find a MME reference case, but in the application of the edge chain detection, it works. The basic idea is that "a meaningful line segment on the image is also a meaningful edge chain". So, given a $w \times h$ image $\mathbf{I}$, first of all we should get the minimum length L_{mm} of a meaningful line segment, which can be very well solved according to the works of LSD [20] and CannyLines [22]:

$$L_{mm} = -2.5 \log(M) / \log(p), \tag{9}$$

where $M = w \times h$ is the size of $\mathbf{I}$ and $p = 1/8$. Thus, we can give the definition of the "minimal meaningful edge chain event" of an image:

Definition of MME$_{\text{edge}}$ - *Minimal Meaningful Edge Chain Event*. A minimal meaningful edge chain event is defined as the edge chain with a size of L_{mm} and a minimal gradient magnitude equal to $g_{\min}$.

The $g_{\min}$ is a user defined parameter which is set as a constant, in Section 5.1.2 we will demonstrate how to find the best value of $g_{\min}$ for all the applications. Thus the RNFA of an edge chain can be reformed as follows:

$$\text{RNFA}_{\text{edge}} = \frac{\mathcal{B}(n, k, p)}{P(g_{\min})^{L_{mm}}}. \tag{10}$$

If $\text{RNFA}_{\text{edge}} < 1$, we simply say that the edge chain is meaningful.

3.2 Edge Chain Detector

An edge chain should have the following qualities on a gradient magnitude map: (1) made up of edge pixels (zero-crossing pixels [11]); (2) smooth orientation deviations between consecutive edge pixels. Based on these observations, an efficient edge chain detector is proposed as follows:

(1) First, given a gray image $\mathbf{I}$, a 3×3 Gaussian filter with the standard deviation $\sigma = 1$ is applied to suppress noise and smooth out the image, then the gradient magnitude map $\mathbf{G}$ and gradient orientation map $\mathbf{O}$ of $\mathbf{I}$ are calculated by applying a certain gradient operator (a 3×3 Sobel was applied in our work).

(2) Then, the non-maximum suppression procedure is applied on $\mathbf{G}$, the gradient magnitudes of those suppressed pixels are set zero and the remaining ones are collected as edge pixels, the set of which is denoted as $\mathbf{E}$.

(3) Third, the set $\mathbf{E}$ is roughly sorted in descending order according to the gradient magnitudes. The foremost unprocessed edge pixel in $\mathbf{E}$ is selected as the initial seed pixel $\mathbf{p}_{seed}$. The 8-neighborhood of the $\mathbf{p}_{seed}$ is searched, if there exists a 8-neighbor who is an unprocessed edge pixel, we consider this pixel to be the next seed pixel, then we add it into the current edge chain and begin 8-neighbor searching from this newly added pixel. On each step, the gradient orientation deviation between $\mathbf{p}_{seed}$ and the newly added edge pixel is calculated. The seed growing of the current edge chain is conducted iteratively until all the pixels in this chain is processed or there exist two consecutive steps where the orientation deviations are both greater than θ, which means that there is no more smooth edge chains exist in front, then we add the current edge chain into the edge chain set $\mathcal{C}$ and begin with another edge chain from the rest of $\mathbf{E}$.

(4) Each edge chain detected in the step (3) is validated by the Helmholtz principle on edge chain proposed in Section 3.1 to get rid of the false alarms. The same strategy as that of EDPF is applied, that is: if the RNFA of an edge chain $\mathbf{C}$ is larger than 1, i.e., RNFA > 1, the pixel $\mathbf{p}$ is the one with the minimum gradient magnitude in $\mathbf{C}$, then $\mathbf{C}$ is divided into two sub edge chains $\mathbf{C}_1$ and $\mathbf{C}_2$ at $\mathbf{p}$, then $\mathbf{C}_1$ and $\mathbf{C}_2$ are added into $\mathcal{C}$ and validated later.

Algorithm 1 describes the complete procedure in details of the proposed edge chain detector. It's worth noting that there are two internal parameters in the proposed edge chain detector: θ and $g_{\min}$. The value of θ is set as $\pi/8$ for constant, so the performance of the proposed algorithm can be adjusted by setting a customized $g_{\min}$, and the bigger the value of $g_{\min}$ is the more meaningful the final edge chains are. In Section 5.1.2 we will demonstrate how to find the

best value of $g_{\min}$ for all the applications.

Algorithm 1 Our Proposed Edge Chain Detector

Require: The gray image $\mathbf{I}$.
Ensure: The set of meaningful edge chains $\mathcal{C}$.
1: $[\mathbf{G}, \mathbf{O}] \leftarrow$ GetImageInfo($\mathbf{I}$)
2: $\mathbf{E} \leftarrow$ NonMaximumSuppression($\mathbf{G}$)
3: $\mathcal{C} \leftarrow$ EdgeChainCollection($\mathbf{G}, \mathbf{E}, \theta$)
4: $\mathcal{C} \leftarrow$ EdgeChainValidation($\mathcal{C}, g_{\min}$)

4 Line Segment Detection

4.1 Line Segment Validation

It is much easier to apply the NFA on validation of line segments than on validation of edge chains, because the value of N_{conf} of the line segment event on the gradient magnitude map is the same as that on the gradient orientation map. Thus, assume that there is a line segment made up of l pixels, the minimal gradient magnitude of these pixels is u, the NFA of this line segment is:

$$\text{NFA} = M^{2.5} \times P(u)^l, \tag{11}$$

where $M = w \times h$ is the size of the image. If NFA < 1, we simply say that the event is meaningful.

It's worth noting that we can also apply the RNFA on the line segment validation if a minimal meaningful line segment event is given. The choice of applying NFA or RNFA depends on the estimation of N_{conf}. If we can't give out a convincing estimation of N_{conf}, like that of the edge chain detection, we can pick out a minimal meaningful event (MME) and apply RNFA for the Helmholtz principle. In fact, both N_{conf} and MME are representations of the background model. We recommend to apply both NFA and RNFA to double check the line segments, in this way the performance of the validation procedure can be adjusted by setting a customized $g_{\min}$. In Section 5.2.2 we will demonstrate the difference between the NFA, RNFA and NFA+RNFA based methods.

4.2 Line Segment Detector

A line segment should have the following qualities on a gradient magnitude map: (1) made up of edge pixels; (2) those edge pixels should roughly on a straight line. Based on this observation, a new line segment detector is proposed which detects line segments directly on the gradient magnitude map instead of the edge chains as the EDLines [1] and CannyLines [22]. The first two steps are the same as those of the edge chain detector proposed in Section 3.2 and the resting steps are described as follows:

(1) First, the set $\mathbf{E}$ is roughly sorted in descending order according to the gradient magnitudes. The foremost unprocessed edge pixel in $\mathbf{E}$ is selected as the initial seed pixel $\mathbf{p}_{seed}$, and the tangent line $\mathbf{l}$ of $\mathbf{p}_{seed}$ (passing $\mathbf{p}_{seed}$ and orthogonal to the gradient orientation of $\mathbf{p}_{seed}$) is set as the initial line segment. Then the 8-neighborhood of $\mathbf{p}_{seed}$ is searched and any edge pixel in the 8-neighborhood whose orthogonal distance to $\mathbf{l}$ is less than 2.0 is collected as the "support edge pixels". All the collected support edge pixels are as a set $\mathbf{E}_{line}$. The growing procedure of the current line segment is conducted iteratively until all the pixels in $\mathbf{E}_{line}$ are processed, and then we begin with another line segment from the rest of $\mathbf{E}$. To efficiently fit the line segment in the growing procedure, we adopt the following simple strategy. Whenever the number of newly added support edge pixels is greater than $1/5$ of the size of $\mathbf{E}_{line}$, the tangent line $\mathbf{l}$ is updated by applying the least square fitting on $\mathbf{E}_{line}$.

(2) Each line segment detected in the above step is validated by the Helmholtz principle on line segment proposed in Section 4.1. The same validation strategy is applied as that in Section 3.2.

Algorithm 2 describes the complete procedure in details of the proposed line segment detector.

TABLE 1
Comparison between Level Lines, Edge Chain and RNFA.

	Level Lines [17]			Edge Chain [18]			RNFA		
Measurements	R	P	F	R	P	F	R	P	F
Our Benchmark	0.90	0.60	0.71	**0.92**	0.53	0.66	0.82	**0.75**	**0.77**

Algorithm 2 Our Proposed Line Segment Detector

Require: The gray image $\mathbf{I}$.
Ensure: The set of meaningful line segments $\mathcal{L}$.
1: $[\mathbf{G}, \mathbf{O}] \leftarrow$ GetImageInfo($\mathbf{I}$)
2: $\mathbf{E} \leftarrow$ NonMaximumSuppression($\mathbf{G}$)
3: $\mathcal{L} \leftarrow$ LineSegmentCollection($\mathbf{G}, \mathbf{E}$)
4: $\mathcal{L} \leftarrow$ LineSegmentValidation($\mathcal{L}$)

5 Experimental Results

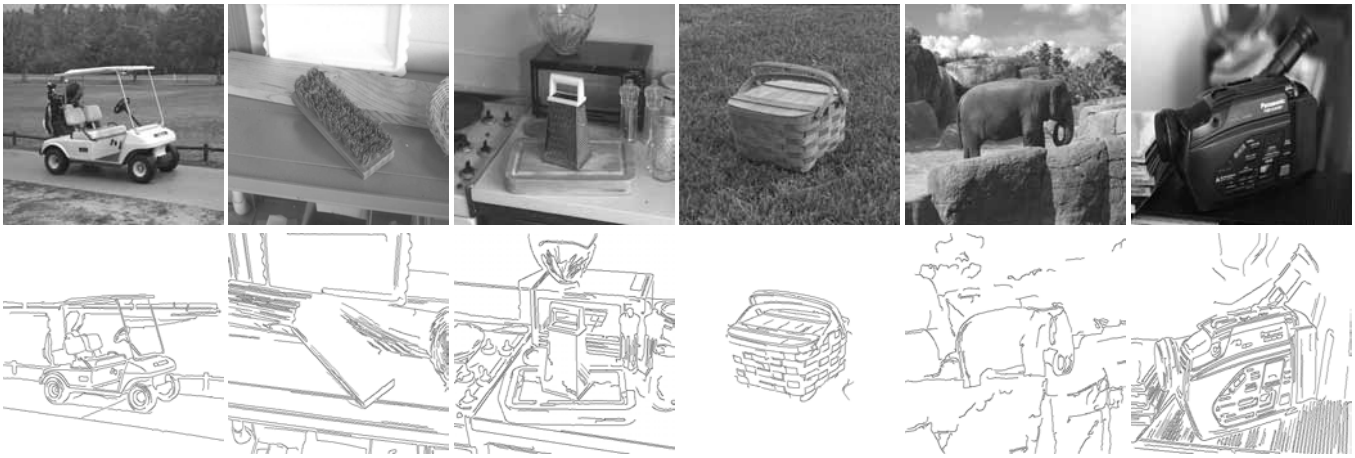
In this section, we built two benchmarks for comprehensively evaluating our proposed edge chain and line segment detectors, respectively. In addition, other two widely used databases were also used to verify that our proposed detectors outperform the state-of-the-art methods. Our built benchmarks, more experimental results and source codes of our proposed methods are publicly available at http://cvrs.whu.edu.cn/projects/RNFA/.

5.1 Evaluation on Edge Chain Detector

5.1.1 Edge Chain Benchmark

To evaluate the performance of the proposed edge chain detector, we built a benchmark with ground truth edge chains labeled in a semi-automatic way. The reason why we don't directly use two widely used public datasets: the BSDS dataset [1] [26] and the RUG dataset [2] [27] is that the BSDS dataset focuses mainly on the boundaries of objects, while the RUG dataset concentrates on the edges of objects on the textured background. Both of these two datasets are more or less based on objects instead of edge chains, on the contrary, the proposed benchmark is a specific edge chain benchmark. Considering the fact that there are similarities between object boundaries and edge chains, in this work we still tested the proposed edge chain detector on both

1. Available at https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
2. Available at http://www.cs.rug.nl/~imaging/databases/contour_database/contour_database.html



(a) Golfcart (b) Brush (c) Grater (d) Picnic Basket (e) Elephant (f) Videocamera

Fig. 6. Six representative images and the corresponding ground truth edge chains on the proposed benchmark.

TABLE 2
Comparison of CannyPF, ED, SREdge, EDPF and RNFA on the proposed edge chain benchmark and the RUG dataset.

	CannyPF [22]			ED [16]			SREdge [15]			EDPF [18]			RNFA		
Measurements	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Our Benchmark	**0.88**	0.49	0.59	0.83	0.55	0.62	0.79	0.61	0.65	0.81	0.65	0.71	0.82	**0.75**	**0.77**
RUG Dataset	0.82	0.16	0.26	**0.83**	0.19	0.29	0.81	0.23	0.35	0.72	0.26	0.37	0.60	**0.44**	**0.49**

BSDS and RUG datasets to demonstrate the performance of the proposed algorithm as an object boundary detection method.

The edge chain labeling is much more difficult than the object boundary labeling, because there usually exists mass and faint edge chains within an object, which are difficult to distinguish directly on the gray-level image in vision. In order to solve this problem, we developed a software to track the edge chains on the edge map in a semi-automatic way. First, the gray-level image was converted into a binary map by applying the non-maximum suppression, and those pixels whose gradient magnitudes are less than 5.0 were discarded to decrease the influence of noise. Then, an experienced subject was asked to pick out a single point on each edge chain which is "meaningful" in vision on the binary map, and then an edge tracking procedure was executed automatically to extract the edge chain from the binary map starting based on the picked point. Also, the subject was asked to draw a polyline by hand to fit for an edge chain if this chain was not automatically extracted well, and then the edge pixels were collected around the polyline to fit for an edge chain.

There are 25 labeled images in our benchmark, most of which were selected from the EDC dataset [3] [28], [29] despite of several natural images that are too difficult for human to label. The images cover a range of textured and non-textured man-made and natural scenes. Fig. 6 shows six representative images and the corresponding labeled edge chains.

To evaluate the accuracy of the edge chain detection result, we use the same F-score metric as EDPF [18]. Let DC be the set of edge pixels detected by a certain method, GT denotes that of the ground truth data, the precision (P) and recall ratio (R) are defined as follows:

$$P = \frac{\#\{\text{DC} \cap \text{GT}\}}{\#\{\text{GT}\}} \quad \text{and} \quad R = \frac{\#\{\text{DC} \cap \text{GT}\}}{\#\{\text{DC}\}}. \tag{12}$$

The F-score is defined as $F = 2PR/(P+R)$.

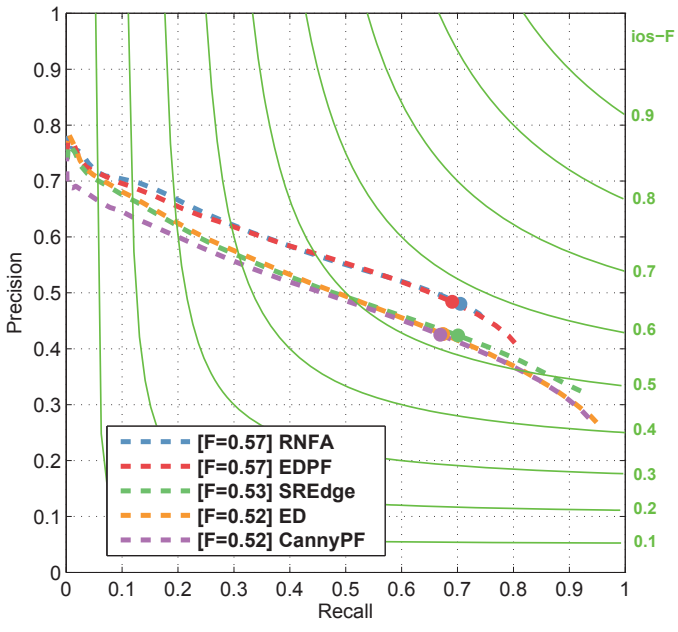
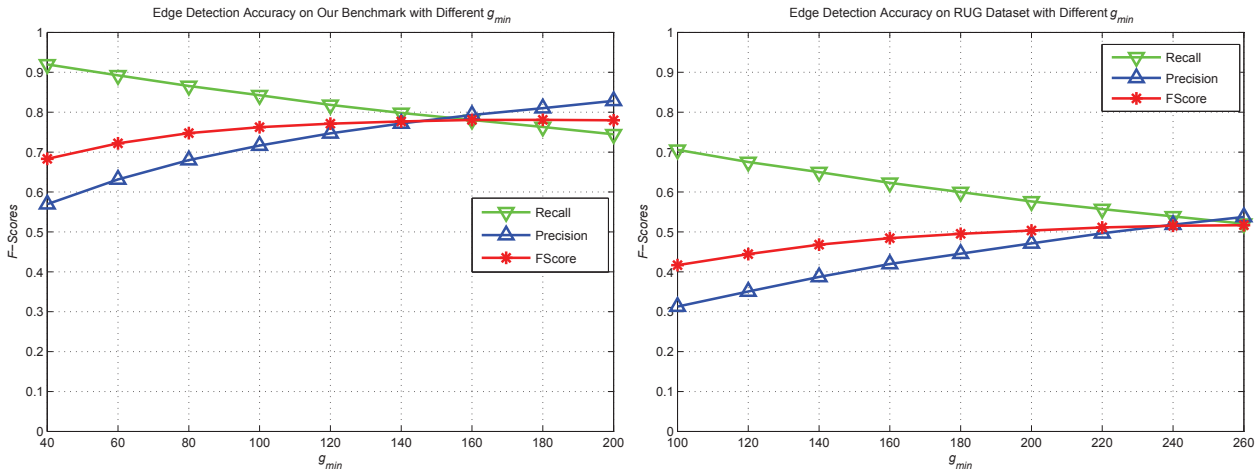


Fig. 9. PR curve of the five tested algorithms on the BSDS dataset.

3. Available at http://marathon.csee.usf.edu/edge/edge_detection.html



(a) our benchmark (b) RUG dataset

Fig. 7. Different performances of our proposed edge chain detection algorithm with different values of $g_{\min}$ on the proposed benchmark and the RUG dataset.

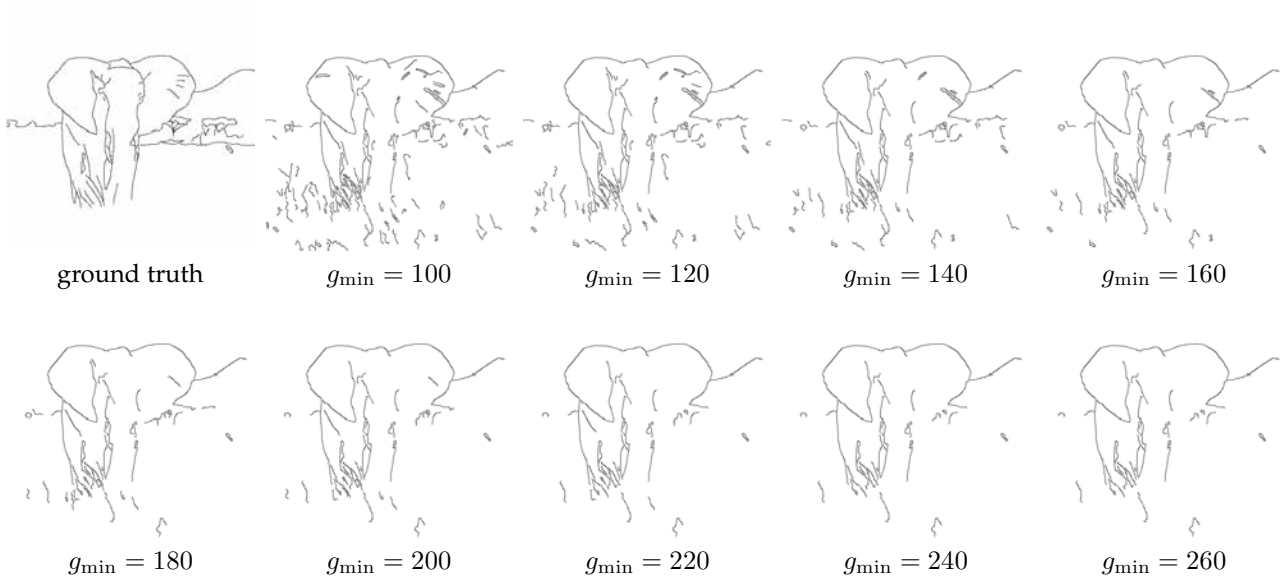


Fig. 8. Comparison of the edge chain detection results with different $g_{\min}$ on the *elephant_2* image in RUG dataset.

5.1.2 *Choice of Best* $g_{\min}$

As an edge chain detector, we consider to detect as many meaningful edge chains as possible. In this condition, we should find the value of $g_{\min}$ corresponding to the minimal meaningful edge chain event. To find out the best value of $g_{\min}$ for edge chain detection, we tested the proposed edge chain detector on our edge chain benchmark with $g_{\min}$= 40, 60, 80, 100, 120, 140, 160, 180, and 200, respectively. Fig. 7(a) shows the edge chain detection results on our benchmark with different values of $g_{\min}$, we can see that with the increment of $g_{\min}$, the recall is decreasing while the precision and the F-score are increasing. When $g_{\min} < 120$, the increment of precision is greater than the decrement of recall, while when $g_{\min} > 120$ they are generally in the same magnitude, so the increment of F-score is close to 0. Thus to detect the edge chains of an image, we recommend to set $g_{\min} = 120$ as a balance between the precision and recall.

As an object boundary detector, we consider to detect as many boundary as possible. In this condition, the value of $g_{\min}$ should be greater than that of edge chain detection because the boundaries are usually with greater contrast. To find out a general value of $g_{\min}$ for boundary detection, we tested the proposed edge chain detector on the RUG dataset with $g_{\min}$= 100, 120, 140, 160, 180, 200, 220, 240, and 260, respectively. Fig. 7(b) shows the edge chain detection results on the RUG with different values of $g_{\min}$, we can observe the same configurations as that on Fig. 7(a) that the F-score is increasing with the increment of $g_{\min}$ with the cost of the decrement of recall. A higher F-score may sacrifice the completeness of detection result. Fig. 8 is a demonstration of the edge chain detection results on the *elephant_2* image in the RUG dataset, we can see that when $g_{\min} = 260$ some ground truth edge chains were not detected out, but due to the high value of precision, the F-score of this image is still high. Thus to detect the boundaries of objects on an image, we recommend to set $g_{\min} = 180$ as a balance between the precision and recall.

In conclusion, as we have stated in Section 3.2 that the

performance of the proposed algorithm can be adjusted by setting a customized $g_{\min}$, which has already been demonstrated according to Figs. 7 and 8, and the bigger the value of $g_{\min}$ is the more meaningful the final edge chains are.

5.1.3 Comparison of Level Lines, Edge Chain and RNFA

As we have mentioned in Section 2.2 that in the works of [17] and EDPF [18], the level lines and the edge chains were used to calculate the value of N_{conf}, respectively. In this section, we will compare the performance of these two methods with our proposed RNFA method on our built benchmark. The level lines were created with the level quantization step equal to 1, the edge chains were detected by the method proposed in Section 3.2, and the value of $g_{\min}$ was set as 120. Table 1 shows the average accuracies of these three methods on all the 25 images in our benchmark. From Table 1, we can see that the proposed RNFA method achieves the best scores on precision and F-score, which are better than those of both the level lines and the edge chain based method. In fact the values of N_{conf} of the six images in Fig. 6 calculated based on the level lines and edge chains are $\{1280942827, 982019578, 868959894, 1368890674, 951913875, 887204182\}$ and $\{1034672, 1731733, 1071010, 736707, 598523, 1582839\}$, respectively. In average, the values of N_{conf} calculated based on level lines are around 1100 times bigger than those calculated based on the edge chains, which is the reason why the edge chain based method gains the higher recall ratios but lower precisions than the other two methods. As a conclusion, the proposed RNFA method can achieve better accuracy than the level lines and edge chain based methods, however we don't have to obtain the level lines of an image in advance.

5.1.4 Comparison with State-of-the-Art Methods

To sufficiently evaluate the performance of our proposed RNFA based edge chain detection method, we compared it with other four state-of-the-art edge detection methods, including: EDPF [18], ED [16], SREdge [15] and CannyPF [22]. The source codes of ED and EDPF can be obtained from the Edge Drawing library [30], the source code of our previously proposed CannyPF is publicly available [4] and the source code of SREdge was implemented by us according to the original paper. All the algorithms were tested on the proposed edge chain benchmark, the RUG dataset and the BSDS dataset.

Table 2 shows the average accuracies of these algorithms on the proposed benchmark and the RUG dataset. We can see in this table that the proposed RNFA method achieves the highest values on both precision and F-score, which are much better than the EDPF on the second place. The edge chain based method SREdge also performed very well, better than both ED and CannyPF, considering the fact that it applied the saliency instead of the Helmholtz principle to validate the edge chains. We can also find out that the algorithms EDPF and RNFA, that apply the Helmholtz principle as a validation procedure, achieve higher precisions than those of CannyPF, ED and SREdge that do not apply the Helmholtz principle, which proves the effectiveness of the Helmholtz principle. What should be noticed is that if we keep $g_{\min} = 120$ for both the proposed benchmark and the RUG dataset, the recall, precision and F-score on the RUG dataset is 0.68, 0.35 and 0.44, respectively, which is still much better than that of EDPF and the resting methods. Fig. 9 shows the precision recall curve of the five tested algorithms on the BSDS dataset, in which we set $g_{\min} = 180$ for the proposed RNFA method. We can see that the performances of the RNFA and EDPF are close to each other with the same F-score equals 0.57, which are better than that of the SREdge (0.53), ED (0.52) and CannyPF (0.52). This performance on the BSDS dataset is consistent with those of the RUG dataset and our benchmark, which verifies the robustness and effectiveness of the proposed RNFA method as an edge chain and boundary detector.

Fig. 11 shows the edge detection results of these five algorithms on six test images in our benchmark. We can observe from Fig. 11 that the proposed RNFA method achieves the best performance, EDPF and RNFA generated less false alarms than the other three methods, which is consistent with the conclusion drawn above.

5.2 Evaluation on Line Segment Detector

5.2.1 Line Segment Benchmark

To evaluate the performance of the proposed line segment detector, we built a benchmark with ground truth line segments labeled in the same way as the edge chain benchmark in Section 5.1. The reason why we don't use the YorkUrbanDB dataset [31] is that the line segments were labeled for vanishing points detection instead of line segment detection. In our benchmark, the line segments were labeled by edge pixels tracking and the least square fitting on the edge map which was obtained in the same way as the edge chain benchmark. There are 30 labeled images in our benchmark, all of them were selected from the YorkUrbanDB dataset [5]. These images cover a range of indoor and outdoor scenes. The first two rows on Fig. 12 show six representative images and the corresponding labeled line segments of our line segment benchmark.

5.2.2 Comparison of NFA and RNFA

As we have stated in Section 2.3 that both NFA and RNFA can be used in the validation of line segments on the gradient magnitude map. In this section we will show the difference of these two methods. Fig. 10 shows the line segment detection results of NFA, RNFA and NFA+RNFA based methods on a low-contrast image and a high-contrast one, respectively. We can observe from Fig. 10 that the NFA based method tends to detect false alarms on the low-contrast image, while the RNFA based method is apt to detect false alarms on the high-contrast image. The average recall ratio, precision and F-score in Table 3 show that the RNFA based method tends to gain higher recall ratio but lower precision than the NFA based method, but the average F-scores of them on all 30 test images are very close. The recall ratio and precision of the NFA+RNFA based method are in-between these of the NFA and RNFA methods, but the F-score of the NFA+RNFA based method is the highest among these three methods. In general, the NFA and RNFA

4. Available at http://cvrs.whu.edu.cn/projects/cannyLines/

5. Available at http://www.elderlab.yorku.ca/YorkUrbanDB/

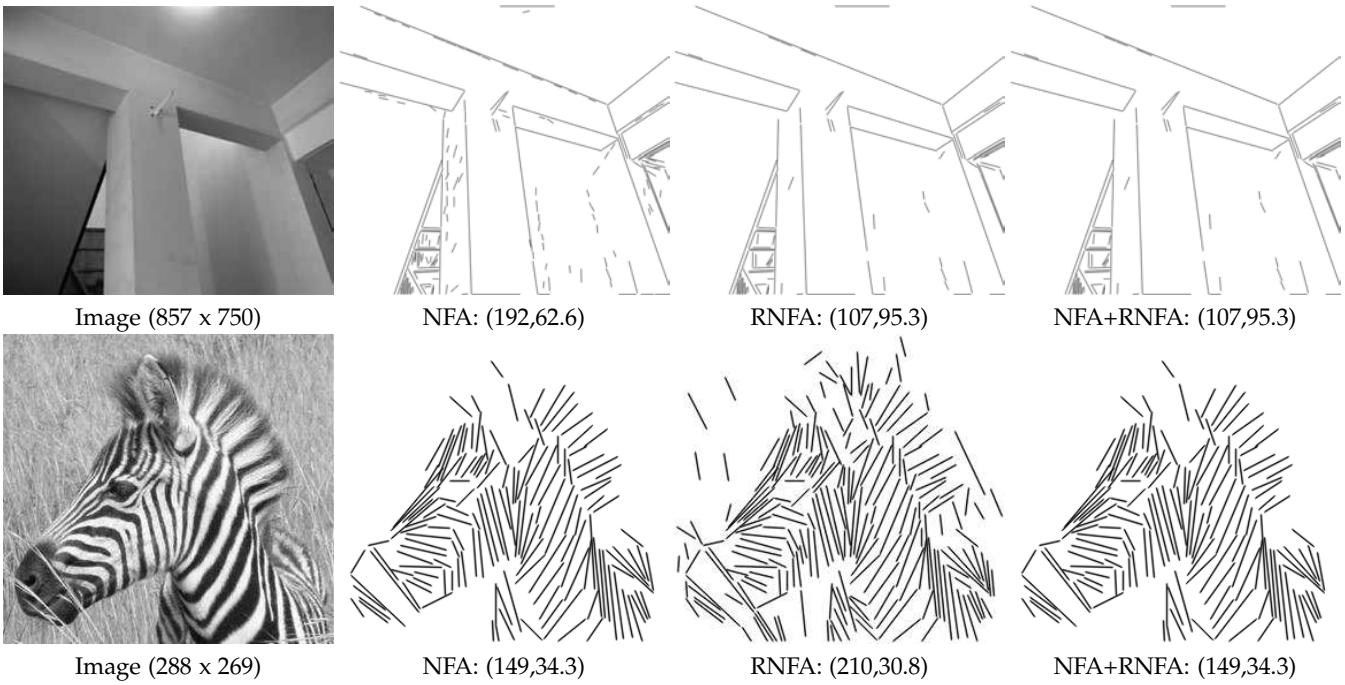


Fig. 10. The line segment detection results of the NFA, RNFA and NFA+RNFA based methods on a low-contrast image in the first row and a high-contrast one in the second row, respectively. $(\cdot,\cdot)$ denotes the line number and average line length of an image.

TABLE 3
Comparison of EDLines, LSD, CannyLines, NFA, RNFA and NFA+RNFA.

	EDLines [1]			LSD [20]			CannyLines [22]			NFA			RNFA			NFA+RNFA			RNFAEdge		
Measurements	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Average	0.638	0.724	**0.675**	0.729	0.796	**0.757**	0.792	0.819	**0.802**	0.882	0.825	**0.848**	0.858	0.846	**0.849**	0.862	0.841	**0.851**	0.827	0.865	**0.843**

based method may have different performance on a certain image, but their average performance are close. Combining both NFA and RNFA to double check the line segments is a good balance between NFA and RNFA. The statistic result in Table 3 is a good testament to this explanation.

5.2.3 Comparison with State-of-the-Art Methods

To sufficiently evaluate the performance of the proposed NFA+RNFA line segment detector, we compared it with other three state-of-the-art line segment detectors, including: CannyLines [22], LSD [20] and EDLines [1]. Considering that the edge chains can be directly applied on line segment detection via the least square fitting, we also tested the line segment detection result based on the edge chains detected by the proposed RNFA based edge chain detector. For each edge chain, we applied the least square method to fit for a line segment from the very beginning of the edge chain, if a line segment was well fitted, the following pixels on the edge chain was added consecutively into this line segment when the distance from this pixel to the current line segment was smaller than 1.0 pixel, otherwise the current line segment was preserved and a new line segment will be fitted from the resting pixels. Then the same gradient orientation line validation procedure as LSD and EDLines is applied to get rid of false alarms. We denote this line segment detector as RNFAEdge.

Fig. 12 shows the line segment detection results of NFA+RNFA, RNFAEdge, CannyLines, LSD and EDLines (from the third row to the last one) on six test images in our benchmark. We can see that the NFA+RNFA method performed best and generated the line segments with less in amount but longer in average length than the other detectors, which means the NFA+RNFA method can generate more complete line segments without loss in accuracy. The RNFAEdge method detected more short line segments than the NFA+RNFA method, and the reason is that the RNFAEdge is based on the edge chains which may suffer more from the noisy effect than the NFA+RNFA method who detects line segments directly from the gradient magnitude image. However, the differences between RNFAEdge and NFA+RNFA are small, which can be seen in Table 3, the F-score of NFA+RNFA and RNFAEdge are 0.851 and 0.843, respectively, which is much better than the other three methods. The EDLines and LSD usually miss the line segments that are week in contrast but long enough to be recognized by vision, and also tend to detect a long line segment into small fragments in the low contrast area for the reason that the noise is relatively salient when the contrast is low. The CannyLines performed better than both the EDLines and LSD methods, but it can still not detect the weak line segments very well.

Fig. 13 shows the line segment detection results of EDLines, LSD, CannyLines, RNFAEdge and NFA+RNFA on four test images with 0%, 5%, 10% and 15% Gaussian white noise added, respectively. We can see that the NFA+RNFA method performed best in different noise conditions with

little false alarms, the RNFAEdge method also performed well with most of the line segments which were detected out unbroken, the CannyLines method detected false alarms with the increment of noise, the LSD method suffered most from the influence of noise with most of its detected line segments in the 15% noisy image were broken into fragments, the EDLines method gained as little false alarms as the NFA+RNFA method in all the noisy conditions, but the amounts of line segments are much less than those of the NFA+RNFA method, besides there are false detections in the third test image of the EDLines. As a conclusion, the proposed NFA+RNFA line segment detection method can detect more line segments with less false alarms in different noisy images.

6 Conclusion

In this paper we developed the Helmholtz principle on the gradient magnitude map of an image in the view of probability theory, and proposed a new conception named "relative number of false alarms" (RNFA) as a supplement of the "number of false alarms" (NFA) when the value of N_{conf} is difficult to be estimated. An edge chain detector which applies the RNFA to validate the meaningfulness of the detected edge chains was proposed, along with a line segment detector which detects line segments directly from the gradient magnitude map instead of edge chains like the EDLines and CannyLines. An edge chain benchmark and a line segment one were built to evaluate the proposed edge chain and line segment detectors in quantity, respectively. The experimental results on our built benchmarks and some widely used datasets sufficiently demonstrate that the proposed edge chain and line segment detectors outperform the state-of-the-art methods.

References

[1] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1633–1642, 2011.

[2] L. A. Fernandes and M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," *Pattern Recognition*, vol. 41, no. 1, pp. 299–314, 2008.

[3] S. Ullman and R. Basri, "Recognition by linear combinations of models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 992–1006, 1991.

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[5] H. Bay, V. Ferraris, and L. Van Gool, "Wide-baseline stereo matching with line segments," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[6] H. Kim and S. Lee, "Simultaneous line matching and epipolar geometry estimation based on the intersection context of coplanar line pairs," *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1349–1363, 2012.

[7] J. Chu, A. GuoLu, L. Wang, C. Pan, and S. Xiang, "Indoor frame recovering via line segments refinement and voting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[8] T. Lemaire and S. Lacroix, "Monocular-vision based SLAM using line segments," in *IEEE International Conference on Robotics and Automation*, 2007.

[9] R. Stoica, X. Descombes, and J. Zerubia, "A Gibbs point process for road extraction from remotely sensed images," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 121–136, 2004.

[10] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[11] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[12] V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 699–714, 1986.

[13] L. A. Iverson and S. W. Zucker, "Logical/linear operators for image curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 982–996, 1995.

[14] C. A. Rothwell, J. Mundy, W. Hoffman, and V.-D. Nguyen, "Driving vision by topology," in *Proceedings of International Symposium on Computer Vision*, 1995.

[15] L. Wang, S. You, and U. Neumann, "Supporting range and segment-based hysteresis thresholding in edge detection," in *IEEE International Conference on Image Processing (ICIP)*, 2008.

[16] C. Topal and C. Akinlar, "Edge Drawing: A combined real-time edge and segment detector," *Journal of Visual Communication and Image Representation*, vol. 23, no. 6, pp. 862C–872, 2012.

[17] A. Desolneux, L. Moisan, and J.-M. Morel, "Edge detection by Helmholtz principle," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 271–284, 2001.

[18] C. Akinlar and C. Topal, "EDPF: A real-time parameter-free edge segment detector with a false detection control," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 01, p. 1255002, 2012.

[19] J. B. Burns, A. R. Hanson, and E. M. Riseman, "Extracting straight lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 4, pp. 425–455, 1986.

[20] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[21] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 87–116, 1988.

[22] X. Lu, J. Yao, K. Li, and L. Li, "CannyLines: A parameter-free line segment detector," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[23] F. Cao, "Good continuations in digital image level lines," in *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[24] A. Almansa, A. Desolneux, and S. Vamech, "Vanishing point detection without any a priori information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 4, pp. 502–507, 2003.

[25] A. Desolneux, L. Moisan, and J.-M. Morel, *From gestalt theory to image analysis: a probabilistic approach*. Springer Science & Business Media, 2007, vol. 34.

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE International Conference on Computer Vision (ICCV)*, 2001.

[27] G. Cosmin, P. Nicolai, and M. A. Westenberg, "Contour detection based on nonclassical receptive field inhibition." *IEEE Transactions on Image Processing*, vol. 12, no. 7, pp. 729–739, 2003.

[28] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1338–1359, 1997.

[29] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of edge detectors: a methodology and initial study," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 38–54, 1998.

[30] C. Akinlar and C. Topal, "Edge drawing library," http://ceng.anadolu.edu.tr/cv/EdgeDrawing/.

[31] P. Denis, J. H. Elder, and F. J. Estrada, *Efficient edge-based methods for estimating Manhattan frames in urban imagery*. Springer, 2008.

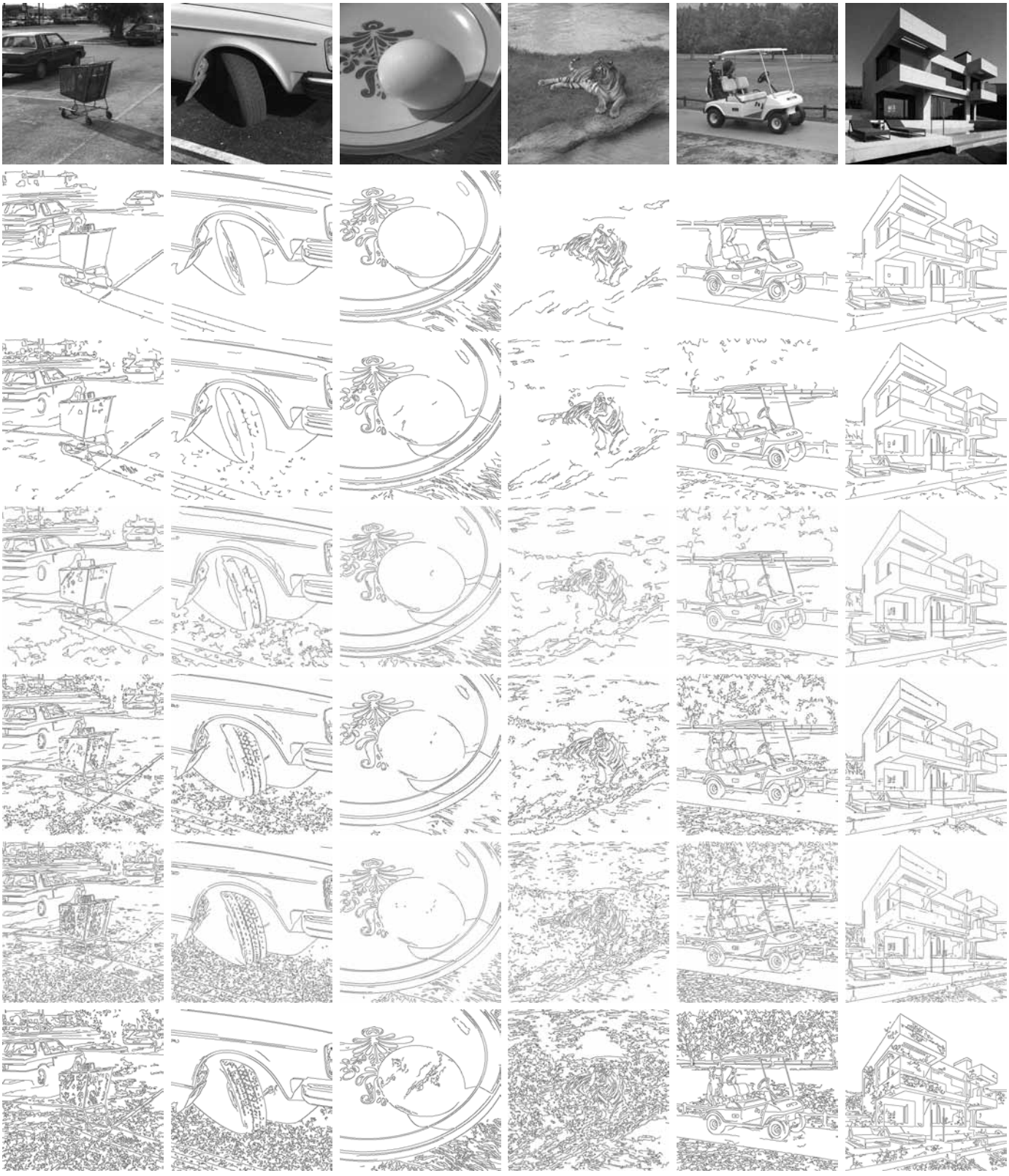


Fig. 11. Six test images in our edge chain benchmark in the first row, ground truth edge chains in the second row, and edge chains detected by RNFA, EDPF, SREdge, ED and CannyPF from the third row to the last one, respectively.

Fig. 12. Six test images in our line segment benchmark in the first row, ground truth line segments in the second row, and line segments detected by NFA+RNFA, RNFAEdge, CannyLines, LSD and EDLines from the third row to the last one, respectively. $(\cdot, \cdot)$ denotes the line number and average line length of an image.

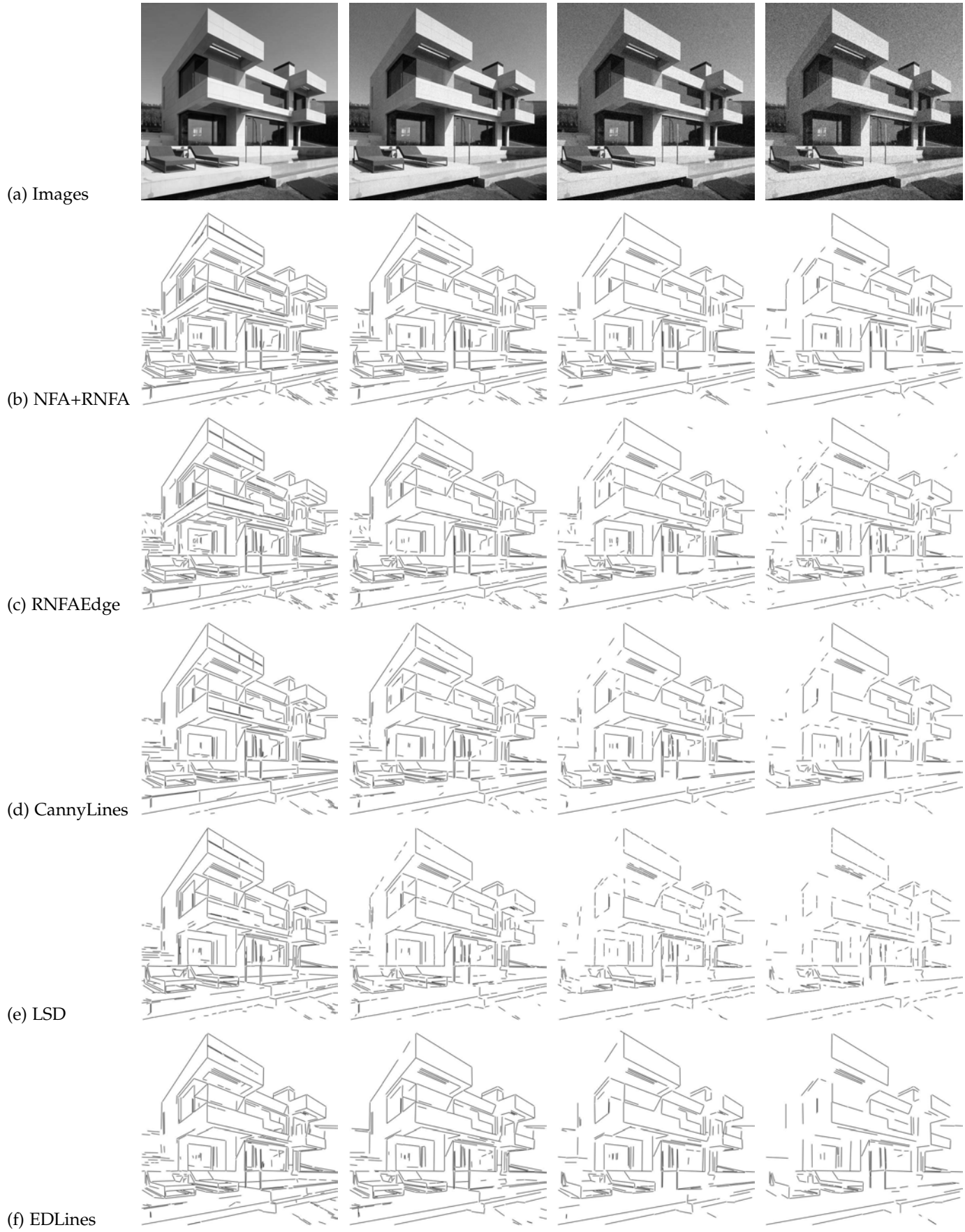


Fig. 13. Four test images with 0%, 5%, 10% and 15% Gaussian white noise added, respectively, in the first row, and line segments detected by NFA+RNFA, RNFAEdge, CannyLines, LSD and EDLines from the second row to the last one, respectively.