# Fuzzy reasoning-based edge detection method using multiple features

LI FANG, WEIREN SHI, SHUHAN CHEN
College of Automation
Chongqing University
No.174, Shazheng Street, Shapingba District, Chongqing
PEOPLE'S REPUBLIC OF CHINA
cqu.funny@yahoo.com.cn, wrs@cqu.edu.cn, c.shuhan@gmail.com

*Abstract:* - Edge detection is an indispensable part of image processing. In this paper, a novel edge detection method based on multiple features and fuzzy reasoning is proposed, in which the limitations of gradient-based edge detection methods and present fuzzy edge detection algorithms can be overcome. The new method selects trapezoid fuzzy membership functions, defines multiple features for each pixel from its neighbors, constructs two sets of fuzzy rules and applies fuzzy reasoning process to determine whether the central pixel is an edge point or not. Extensive experimental results demonstrate that the proposed method performs well in keeping low contrast and blurry edge details, noise suppression and fuzzy rules complexity.

*Key-Words:* - Edge detection, Multiple features, Fuzzy reasoning, Trapezoid fuzzy membership functions, Noise suppression

## 1 Introduction

Edges are one of the most important visual clues for interpreting images [1]. Edge points are pixels at which the intensity of an image function changes abruptly, and edges are sets of connected edge points [2]. Edge detection is by far one of the topics of most active and continuing interest because it is the basis and key issue in image processing, computer vision and pattern recognition, such as License Plate detection, iris recognition, face detection, Synthetic Aperture Radar (SAR) image detection and so on [3-6].

Many edge detection approaches are proposed in the literature. Among all existing methods, gradient-based detectors are the most classic and commonly used such as Roberts, Sobel, Prewitt and Canny [2]. They need a smoothing operation to alleviate the effect of high spatial frequency in estimating the gradient. Usually this smoothing is applied to all pixels in the image including the edge regions, and so the edge is distorted and missed in some cases, in particular at junctions or corners. The gradient magnitude alone is insufficient to determine all the meaningful edges due to the ambiguity caused by the underlying pixel pattern and noises, especially in complex natural scenes [7].

In recent years, a variety of new edge detection techniques have been explored, such as the approaches based on wavelet transform, wavelet package, mathematical morphology, cellular neural networks (CNN) and so on [8-10]. It is noted that most of the methods above can't extract edges from blurry images satisfactorily. Fuzzy set theory is an approximation tool in modeling ambiguity or uncertainty and has been applied in image processing. The traditional fuzzy edge detection (TFED) algorithm can extracts edges successfully from blurry images by introducing fuzzy enhancement [11]. But it is computationally complex because the mapping transformation involves exponential calculation and it will lead to lose some edges in low contrast regions. Recently, scholars have proposed various improved approaches with the simplified mapping transformation and optimized fuzzy enhancement operator. Yang proposes a modified Pal and King algorithm for fuzzy edge detection [12]. It needs only one parameter in this modified algorithm and it can be determined automatically. Zhang and Lian propose a new method of fuzzy edge detection based on Gauss function [13]. Gauss enhancement operator is adopted in this method. Without iterative process , it is more effective than TFED. Although these algorithms can overcome the limitations of TFED to some degree, they are sensitive to noise and usually contain matrix inversion or more complicated operations. In addition, the "min" or "max" operator has a relatively lower edge positioning accuracy in extracting edges from images. Wu and Yin propose a fast multilevel fuzzy edge detection (FMFED) algorithm which can overcome the deficiency of enhancing some edges at the expense of weakening other edges [14]. However, FMFED doesn't perform well in low contrast and noisy images.

Fuzzy reasoning based method is another novel and efficient edge detection approach. Russo and Ramponi design fuzzy rules for edge detection [15-16]. Such rules can smooth while sharpening edges, but requires a rather large rule set compared to simpler fuzzy methods. Tizhoosh proposes three fast edge detection methods to detect rough edge map by fuzzy logic [17]. Liang and Looney put forward a competitive fuzzy edge detection (CFED) method [18]. Both of these methods divide edge types into six patterns and use a fuzzy classifier to determine which pattern the edge type belongs to. In some detailed regions, CFED can't detect the delicate texture and it usually generates speckles. Eghbal G. Mansoori and Hassan J. Eghbali propose a heuristic fuzzy edge detector (HFED) [19]. The HFED extracts three features (block deviation, pixel discrepancy norm and local degree of edge) from local neighbors. Then a fuzzy rule-based classification system is designed by applying the extracted features for each pixel to classify it as part of edge or picture. However, this method is time-consuming due to its high complexity in fuzzy rules.

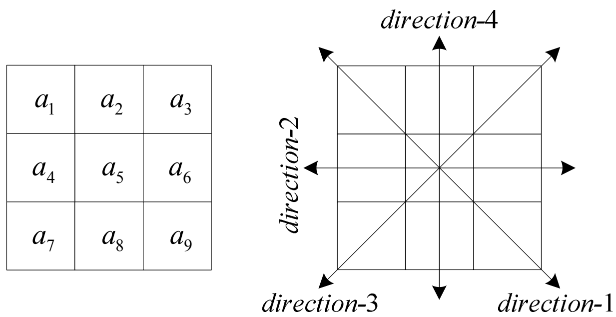
Since the nature of image data is indeterminate, fuzzy reasoning is able to extract meaningful information from approximate, incomplete and imperfect sets of data. We present a novel edge detection method based on multiple features and fuzzy reasoning to overcome the above limitations in this paper. This method chooses the trapezoid fuzzy membership functions, extracts three features for each pixel from its neighbors. These attributes are called direction indices, symmetry and similarity. Two sets of fuzzy rules based on these extracted features are designed to determine edge points. By numerous example illustrations and simulations, we have shown the superiority of the proposed method.

The remainder of this paper is organized as follows. In Section 2, we propose a new edge detection method, which is the main contribution of this paper. Our edge detection results compared with the results of some previous methods are displayed in Section 3. Finally, we present conclusions in Section 4.

# 2 The Proposed Method

## 2.1 Feature Definition

The most difficult problem in fuzzy reasoning based edge detection method is how to define edge features to describe edge points accurately and comprehensively. Meanwhile, how to develop fuzzy rules is also very important. In order to decrease the computational cost, improve efficiency and obtain better edge detection results, we select a 3×3 window in this paper which contains the central pixel $a_5$ and its eight neighbor pixels $W=\{a_1,a_2,a_3,a_4,a_6,a_7,a_8,a_9\}$ as shown in Fig. 1(a). Fig. 1(b) shows the four directions in which edges may appear.




(a) (b)

Fig. 1 (a) 3×3 window; (b) four directions of edge

All of the predefined features of the central pixel $a_5$ use nine pixels in the current 3×3 window above. These attributes are direction indices, symmetry and similarity.

### 2.1.1 Direction Indices $Q_k$:

$Q_k$ is the average diversity between pixels along some direction and the central pixel $a_5$. $S_k, k=1,2,3,4$ denote nine pixels aligning with four directions, respectively.

$$\begin{aligned} S_1 &= \{a_1,a_5,a_9\} \\ S_2 &= \{a_4,a_5,a_6\} \\ S_3 &= \{a_3,a_5,a_7\} \\ S_4 &= \{a_2,a_5,a_8\} \end{aligned} \tag{1}$$

Then, to determine whether there is any edge in the current 3×3 window, four direction indices $Q_k, k=1,2,3,4$ are introduced in equation (2):

$$\begin{aligned} Q_1 &= \frac{1}{2}\cdot\left(|a_1-a_5|+|a_9-a_5|\right) \\ Q_2 &= \frac{1}{2}\cdot\left(|a_4-a_5|+|a_6-a_5|\right) \\ Q_3 &= \frac{1}{2}\cdot\left(|a_3-a_5|+|a_7-a_5|\right) \\ Q_4 &= \frac{1}{2}\cdot\left(|a_2-a_5|+|a_8-a_5|\right) \end{aligned} \tag{2}$$

#### 2.1.2 Symmetry $P_k$:

Excluding 3 pixels on each direction, the other 6 pixels in the current 3×3 window can be divided into two sets. For example, in the edge pattern of *direction*-1, two sets are $\{a_2, a_3, a_6\}$ and $\{a_4, a_7, a_8\}$. $P_k$ is the average symmetry between two sets. $P_k, k = 1,2,3,4$ for directions 1, 2, 3 and 4, respectively, are defined as:

$$
\begin{aligned}
P_1 &= \frac{1}{3}\cdot\left(|a_2 - a_4| + |a_3 - a_7| + |a_6 - a_8|\right) \\
P_2 &= \frac{1}{3}\cdot\left(|a_1 - a_7| + |a_2 - a_8| + |a_3 - a_9|\right) \\
P_3 &= \frac{1}{3}\cdot\left(|a_1 - a_9| + |a_2 - a_6| + |a_4 - a_8|\right) \\
P_4 &= \frac{1}{3}\cdot\left(|a_1 - a_3| + |a_4 - a_6| + |a_7 - a_9|\right)
\end{aligned} \tag{3}
$$

#### 2.1.3 Similarity $R_{k1}$, $R_{k2}$:

As mentioned in section 2.1.2, there are two sets in the edge pattern of some fixed direction. $R_{k1}, R_{k2}$ are the similarity between these two sets and the central pixel $a_5$, respectively. They are calculated by equation (4)-(7):

$$
\begin{aligned}
R_{11} &= \left|\frac{1}{3}\cdot(a_2 + a_3 + a_6) - a_5\right| \\
R_{12} &= \left|\frac{1}{3}\cdot(a_4 + a_7 + a_8) - a_5\right|
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
R_{21} &= \left|\frac{1}{3}\cdot(a_1 + a_2 + a_3) - a_5\right| \\
R_{22} &= \left|\frac{1}{3}\cdot(a_7 + a_8 + a_9) - a_5\right|
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
R_{31} &= \left|\frac{1}{3}\cdot(a_1 + a_2 + a_4) - a_5\right| \\
R_{32} &= \left|\frac{1}{3}\cdot(a_6 + a_8 + a_9) - a_5\right|
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
R_{41} &= \left|\frac{1}{3}\cdot(a_1 + a_4 + a_7) - a_5\right| \\
R_{42} &= \left|\frac{1}{3}\cdot(a_3 + a_6 + a_9) - a_5\right|
\end{aligned} \tag{7}
$$

### 2.2 Edge Detection Method

The proposed approach in this paper consists of two sets of fuzzy rules. Firstly, we apply fuzzy reasoning process on direction indices to extract candidate edge points from the whole image. Then, let symmetry and similarity, corresponding to candidate edge points above, be the inputs of the second fuzzy reasoning process to remove interferential pixels and exact edge points from candidate edge points. Particularly, if one pixel isn't classified as a candidate edge point (CED) in the first fuzzy reasoning, it won't be taken into account in the second fuzzy reasoning. As a result, it reduces the computational loading of our method.

#### 2.2.1 The First Fuzzy Reasoning

As mentioned above, edge points can be thought of as pixel locations of abrupt gray-level change. Gray level of pixels along edge direction changes slightly but sharply on the other three directions. Hence, we apply the first fuzzy reasoning on direction indices $Q_k$ to extract candidate edge points from the whole image. If one pixel is a CED, we compute its edge direction. The details are as follows:

Step1: select membership functions. We select fuzzy membership functions $small(u)$ and $big(u)$ shown in equation (8a) and (8b), respectively. Both are trapezoid shapes and illustrated in Fig. 2. The determination of the two parameters $m$ and $n$ depends on the considered image and will be discussed in section 3.

$$
small(u) = \begin{cases} 1, & u < m \\ \dfrac{u-n}{m-n}, & m \le u < n \\ 0, & u \ge n \end{cases} \tag{8a}
$$

$$
big(u) = \begin{cases} 0, & u < m \\ \dfrac{u-m}{n-m}, & m \le u < n \\ 1, & u \ge n \end{cases} \tag{8b}
$$

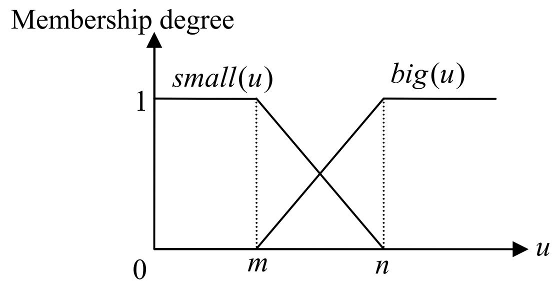



Fig. 2 The trapezoid fuzzy membership functions

Step2: Let the inputs of the first fuzzy reasoning be represented as $Q^l, l = 1,2,3,4$. The definition of $Q^l, l = 1,2,3,4$ is introduced as follows. Sort the direction indices $Q_k, k = 1,2,3,4$ in ascending order as $Q^1 \le Q^2 \le Q^3 \le Q^4$. The superscript $l$ in $Q^l, l = 1,2,3,4$ is the order number of $Q_k$, $k = 1,2,3,4$. Then, let $\Gamma(Q^l)$ denotes the index $k$ of $Q_k, k = 1,2,3,4$, which is at the ascending order $l$ after sorting. Significantly, the original fuzzy rule base has $2^4 = 16$ rules due to four inputs $Q^l, l = 1,2,3,4$ and two membership functions $small(u)$ and $big(u)$. To simplify the fuzzy rule base and reduce the computational loading, four inputs have been sorted such that only five fuzzy rules are fired.

The following five fuzzy rules shown in Table 1 are used to determine which type of the central pixel $a_5$ should be. The two possible types of the central pixel are CED and non-edge point (NED).

Table 1 Fuzzy rules I

| Rule number | $Q^1$ | $Q^2$ | $Q^3$ | $Q^4$ | Pixel Type |
|---|---|---|---|---|---|
| 1 | *big* | *big* | *big* | *big* | NED |
| 2 | *small* | *big* | *big* | *big* | CED |
| 3 | *small* | *small* | *big* | *big* | NED |
| 4 | *small* | *small* | *small* | *big* | NED |
| 5 | *small* | *small* | *small* | *small* | NED |

In the Table 1, for example, Rule 3 can be interpreted as:

If $Q^1$ is *small* and $Q^2$ is *small* and $Q^3$ is *big* and $Q^4$ is *big*, then the central pixel $a_5$ is a NED.

Step3: Let $F_t$, where $t = 1,2,3,4,5$ is the rule number in Table 1, be defined below.

$$\begin{aligned}
F_1 &= big(Q^1)\cdot big(Q^2)\cdot big(Q^3)\cdot big(Q^4)\\
F_2 &= small(Q^1)\cdot big(Q^2)\cdot big(Q^3)\cdot big(Q^4)\\
F_3 &= small(Q^1)\cdot small(Q^2)\cdot big(Q^3)\cdot big(Q^4)\\
F_4 &= small(Q^1)\cdot small(Q^2)\cdot small(Q^3)\cdot big(Q^4)\\
F_5 &= small(Q^1)\cdot small(Q^2)\cdot small(Q^3)\cdot small(Q^4)
\end{aligned} \tag{9}$$

where the product inference engine is adopted in equation (9) to realize the first fuzzy reasoning [20-21]. After all $F_t, t = 1,2,3,4,5$ are obtained, the following five cases are processed.

Case 1: If $\max\{F_1,F_2,F_3,F_4,F_5\} = F_1$, the central pixel $a_5$ is a NED.

Case 2: If $\max\{F_1,F_2,F_3,F_4,F_5\} = F_2$, the central pixel $a_5$ is a CED, and its direction is $\Gamma(Q^1)$.

Case 3: If $\max\{F_1,F_2,F_3,F_4,F_5\} = F_3$, the central pixel $a_5$ is a NED.

Case 4: If $\max\{F_1,F_2,F_3,F_4,F_5\} = F_4$, the central pixel $a_5$ is a NED.

Case 5: If $\max\{F_1,F_2,F_3,F_4,F_5\} = F_5$, the central pixel $a_5$ is a NED.

It's obvious that only when $\max\{F_1,F_2,F_3,F_4,F_5\} = F_2$ is the central pixel $a_5$ a CED and its direction is $\Gamma(Q^1)$.

### 2.2.2 The Second Fuzzy Reasoning

In section 2.2.1, if the central pixel $a_5$ is classified as a CED and $\min\{Q_1,Q_2,Q_3,Q_4\} = Q^1 = Q_i$, $i = \Gamma(Q^1)$, we apply symmetry $P_i$ and similarity $R_{i1}$, $R_{i2}$ to the second fuzzy reasoning to exact edge points from candidate edge points. As is known to all, attributes (gray level in this paper) in two sets divided by edge points are much different. At the same time, attributes of pixels along edge direction should be close to one set, and obviously different from the other set. So we construct the following eight fuzzy rules shown in Table 2 to finally determine which type of the CED $a_5$ should be. The two possible types of the CED $a_5$ are edge point (ED) and NED.

Table 2 Fuzzy rules II

| Rule number | $P_i$ | $R_{i1}$ | $R_{i2}$ | Pixel Type |
|---|---|---|---|---|
| 1 | *small* | *small* | *small* | NED |
| 2 | *small* | *small* | *big* | NED |
| 3 | *small* | *big* | *small* | NED |
| 4 | *small* | *big* | *big* | NED |
| 5 | *big* | *small* | *small* | NED |
| 6 | *big* | *small* | *big* | ED |
| 7 | *big* | *big* | *small* | ED |
| 8 | *big* | *big* | *big* | NED |

Let $E_j$, where $j = 1,2,...,8$ is the rule number in Table 2, be expressed as equation (10). We also use the product inference engine in equation (10) to realize the second fuzzy reasoning [20-21].

$$\begin{aligned}
E_1 &= small(P_i)\cdot small(R_{i1})\cdot small(R_{i2})\\
E_2 &= small(P_i)\cdot small(R_{i1})\cdot big(R_{i2})\\
E_3 &= small(P_i)\cdot big(R_{i1})\cdot small(R_{i2})\\
E_4 &= small(P_i)\cdot big(R_{i1})\cdot big(R_{i2})\\
E_5 &= big(P_i)\cdot small(R_{i1})\cdot small(R_{i2})\\
E_6 &= big(P_i)\cdot small(R_{i1})\cdot big(R_{i2})\\
E_7 &= big(P_i)\cdot big(R_{i1})\cdot small(R_{i2})\\
E_8 &= big(P_i)\cdot big(R_{i1})\cdot big(R_{i2})
\end{aligned} \tag{10}$$

Similarly, after all $E_j, j = 1,2,\ldots,8$ are obtained, we can get eight cases.

Case 1: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_1$, the CED $a_5$ is a NED.

Case 2: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_2$, the CED $a_5$ is a NED.

Case 3: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_3$, the CED $a_5$ is a NED.

Case 4: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_4$, the CED $a_5$ is a NED.

Case 5: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_5$, the CED $a_5$ is a NED.

Case 6: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_6$, the CED $a_5$ is an ED, and its direction is $\Gamma(Q^1)$.

Case 7: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_7$, the CED $a_5$ is an ED, and its direction is $\Gamma(Q^1)$.

Case 8: If $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_8$, the CED $a_5$ is a NED.

We can get that Only when $\max\{E_1, E_2,\ldots, \ldots, E_7, E_8\} = E_6$ or $\max\{E_1, E_2,\ldots, E_7, E_8\} = E_7$ is the CED $a_5$ an ED and its direction is $\Gamma(Q^1)$.

The above analysis in section 2 is summarized as the following procedure:

Step1: Select two membership functions, exact three edge features for each pixel from its neighbors and generate fuzzy rules I and II.

Step2: Calculate $Q_k, k = 1,2,3,4$ in the current 3×3 window from equation (2).

Step3: Realize the first fuzzy reasoning based on $Q_k$ and fuzzy rules I to classify the central pixel $a_5$ as a CED or a NED. If it is a CED, we obtain its edge direction $\Gamma(Q^1)$, otherwise, turn Step2.

Step4: Compute $P_i$ and $R_{i1}, R_{i2}$ along edge direction $\Gamma(Q^1)$ from equation (3)-(7).

Step5: Apply the second fuzzy reasoning process based on $P_i$, $R_{i1}, R_{i2}$ and fuzzy rules II to finally determine whether the CED $a_5$ is an ED or not. Turn Step2.

# 3 Experimental Results

To demonstrate the efficiency of the proposed approach, several experiments based on some standard test images were conducted. We selected a few standard images: Couple, low contrast Lena, blurry Lena, Rice with salt & pepper noise and Rice with speckle noise. For comparison, the simulation results were compared with the results by other methods such as Sobel, method in paper [12], method in paper [13] and HFED. The values of parameters $m$ and $n$ in the fuzzy membership functions must be set in advance. By trial and error, the better performance occurs with the parameters $m \in [5,25]$; $n = 2*m$.

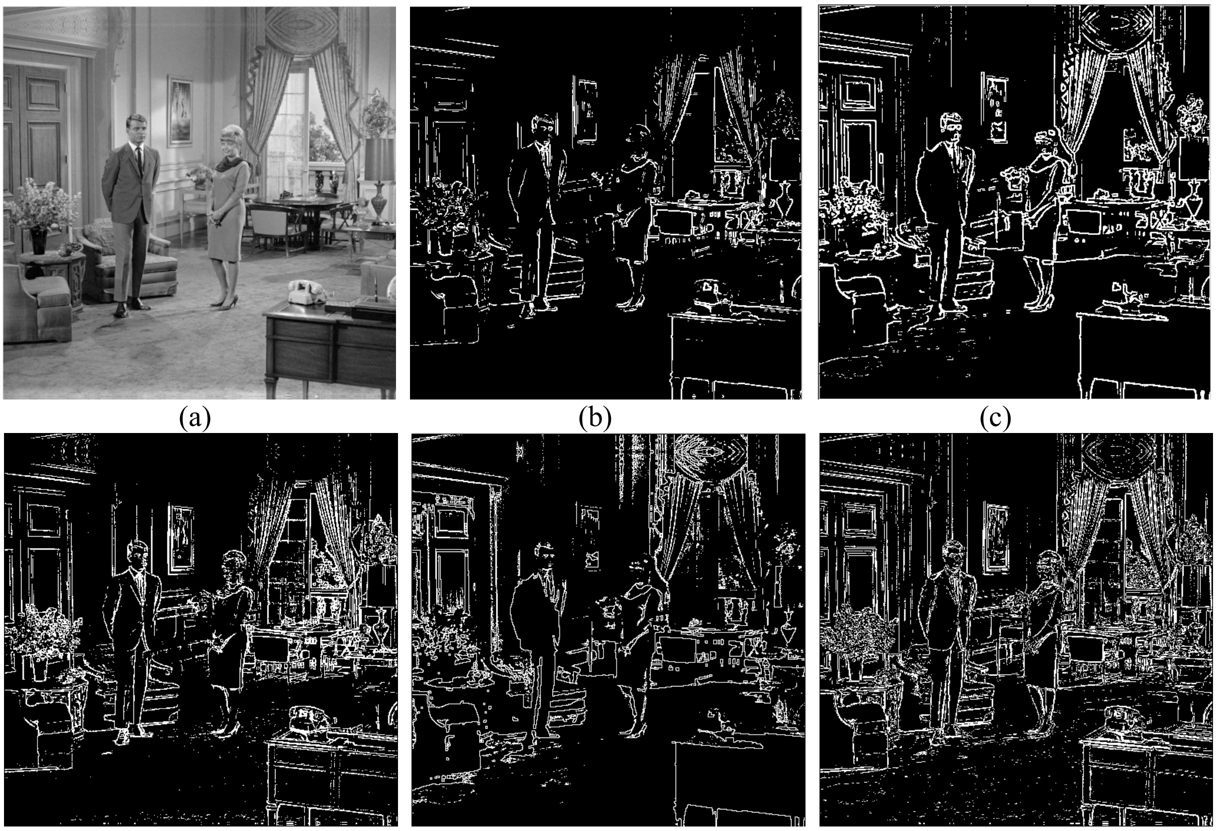
## 3.1 Edge Results

Example 1: Consider the image "Couple" and low contrast "Lena" shown in Fig. 3(a) and Fig. 4(a) which are of the size 512×512 and with 8-bit per pixel. The edge results are illustrated in Fig. 3(b)-(f) and Fig. 4(b)-(f), respectively. Their parameters are set as below.

Fig. 3: (b) Sobel: $T = 28$; (c) method in paper [12]:Threshold is generated by Otsu [22]; (d) method in paper [13]: $\delta = 0.15, t_1 = 0.23, t_2 = 0.7$ and threshold is generated by Otsu; (e) HFED: $\alpha = 1$; (f) our method: $m$=10; $n = 2*m$.

Fig. 4: (b) Sobel: $T = 18$; (c) method in paper [12]:Threshold is generated by Otsu; (d) method in paper [13]: $\delta = 0.15, t_1 = 0.23, t_2 = 0.7$ and threshold is generated by Otsu; (e) HFED: $\alpha = 0.9$; (f) our method: $m$=10; $n = 2*m$.

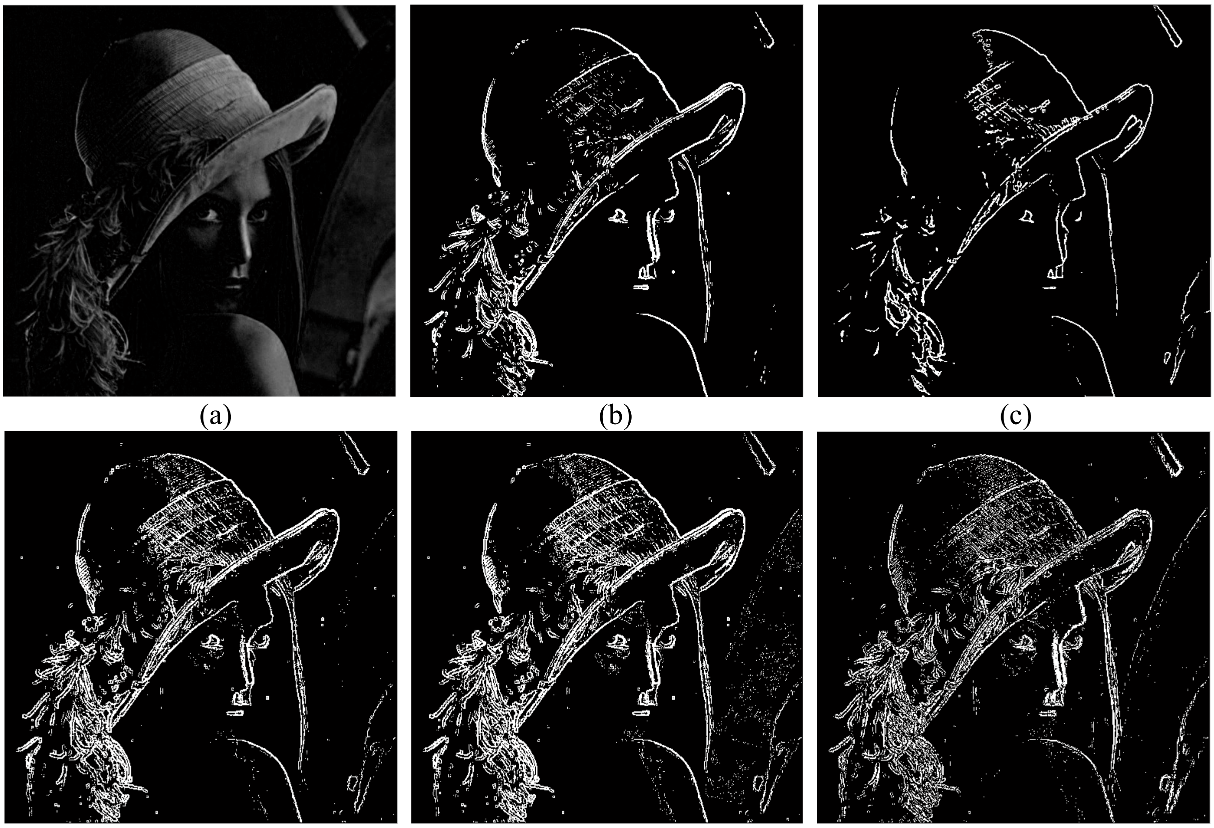
From Fig. 3 we can clearly observe that our method detects most, more accurate and clearly meaningful edges. Furthermore, the details in our method are preserved better than others. For the other four methods, they all lose some edges especially in detailed regions, such as the desk on the bottom right of the image, the man and the picture on the wall in Fig. 3(b), (c), (e) and the top of the image in Fig. 3(b), (d). In addition, from Fig. 4 we can also find that our method obtains the best result in low contrast image.



(a) (b) (c)

(d) (e) (f)

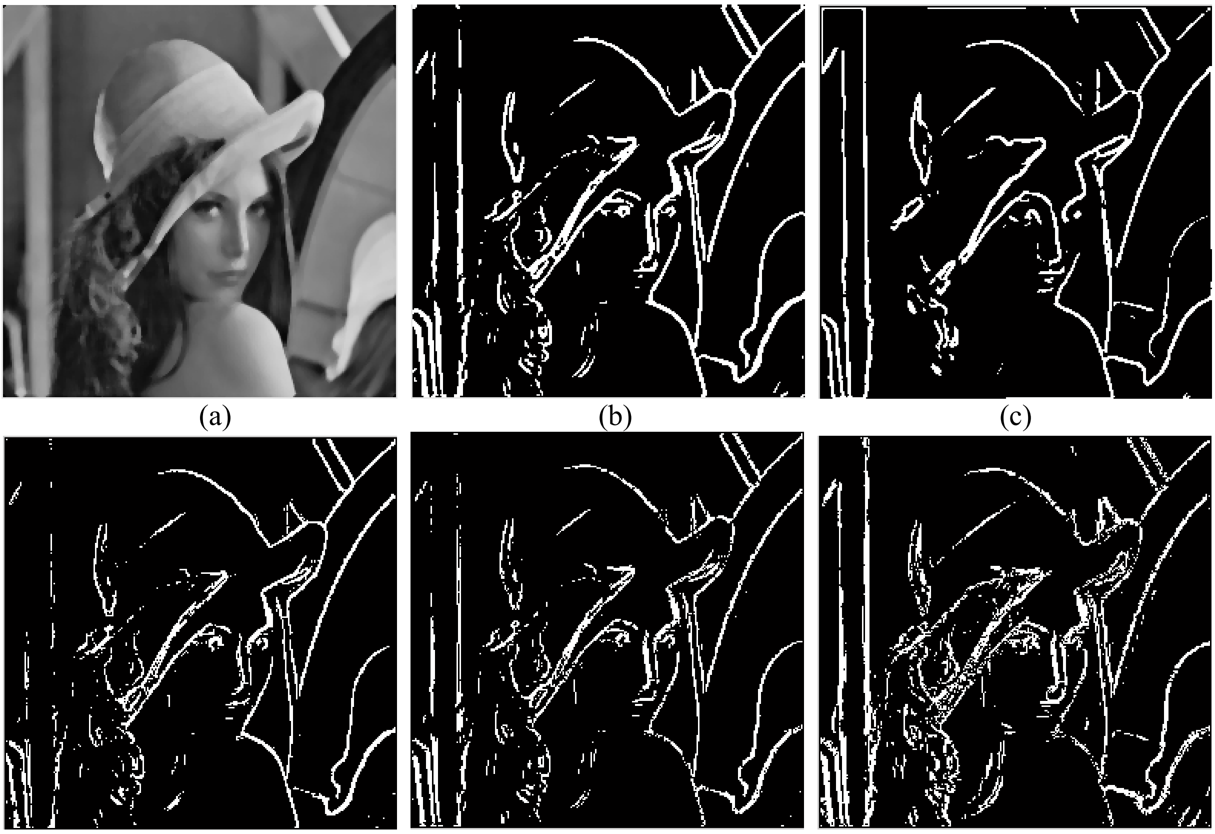
Fig. 3 Comparisons with different methods for image "Couple"
(a) the original image; (b) Sobel; (c) method in paper[12]; (d) method in paper[13]; (e) HFED; (f) our method



(a) (b) (c)

(d) (e) (f)

Fig. 4 Comparisons with different methods for image low contrast "Lena"
(a) the original image; (b) Sobel; (c) method in paper[12]; (d) method in paper[13]; (e) HFED; (f) our method



(a) (b) (c)

(d) (e) (f)

Fig.5 Comparisons with different methods for blurry image “Lena” (blurred by a 5×5 median filter)
(a) the original image; (b) Sobel; (c) method in paper[12]; (d) method in paper[13]; (e) HFED; (f) our method

Example 2: Consider the blurry image “Lena” shown in Fig. 5(a) which is of the size 256×256 and with 8-bit per pixel. Fig. 5(a) is a blurry image “Lena” blurred by a 5×5 median filter. The edge results are illustrated in Fig. 5(b)-(f), respectively. Their parameters are set as: (b) Sobel: $T = 15$; (c) method in paper [12]:Threshold is generated by Otsu; (d) method in paper [13]: $\delta = 0.2, t_1 = 0.23$, $t_2 = 0.7$ and threshold is generated by Otsu; (e) HFED: $\alpha = 0.9$ ; (f) our method: $m$=10; $n = 2 * m$. From Fig. 5 we can obviously obtain that our method exacts more meaningful edges than others.

Example 3: The most common problem of many edge detection methods, as discussed above, is their high sensitivity to noise. Many algorithms work weak in noisy images. In order to show this fact, Fig. 6 and Fig. 7 illustrate the performance of several methods in presence of salt & pepper noise and speckle noise, respectively. Fig. 6(a) and Fig. 7(a) are of the size 256×256 and with 8-bit per pixel. Their parameters are set as follows.

Fig. 6: (b) Sobel: $T = 25$; (c) method in paper [12]:Threshold is generated by Otsu; (d) method in paper [13]: $\delta = 0.5, t_1 = 0.23, t_2 = 0.7$ and threshold is generated by Otsu; (e) HFED: $\alpha = 1.1$ ;(f) our method: $m$=20; $n = 2 * m$.

Fig. 7: (b) Sobel: $T = 15$; (c) method in paper [12]:Threshold is generated by Otsu; (d) method in paper [13]: $\delta = 0.5, t_1 = 0.23, t_2 = 0.7$ and threshold is generated by Otsu; (e) HFED: $\alpha = 1$; (f) our method: $m$=20; $n = 2 * m$.

From Fig. 6 we can easily observe that the performance of our method is better than others and much lower sensitive to salt & pepper noise. Similarly, it’s clear from Fig. 7 that Sobel and our method obtain better results. However, the edges of our method are thinner compared with Sobel, and there is still some noise in Fig. 7(b).

### 3.2 Speed Results

Table 3 lists the processing time for the method in paper [12], method in paper [13], HFED and the proposed method using a Core 2 Duo 1.59 GHz PC for various image sizes. From Table 3, we can observe that our method is slightly slower than method in paper [12], but it is significantly faster than HFED.

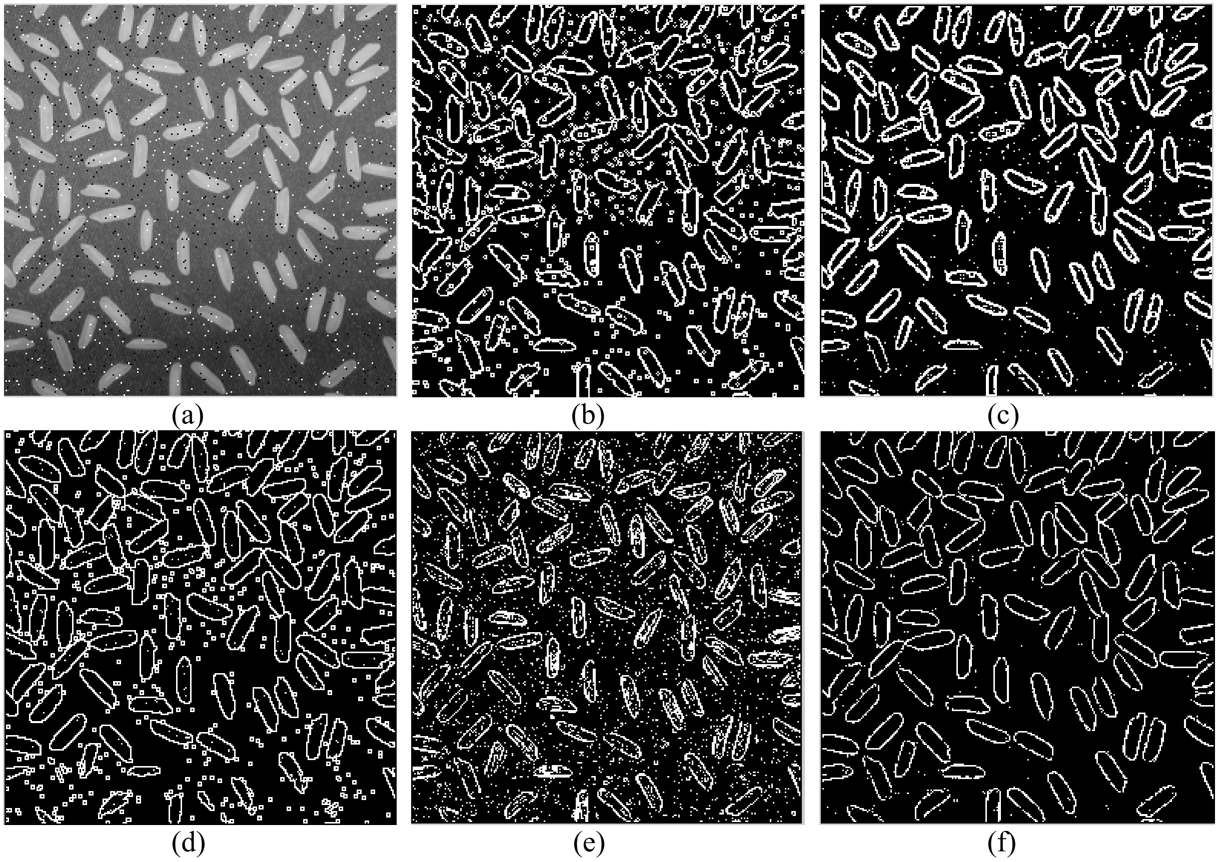



Fig. 6 Comparisons with different methods for image "Rice" with salt & pepper noise (noise density=0.02)
(a) the original image; (b) Sobel; (c) method in paper[12]; (d) method in paper[13]; (e) HFED; (f) our method

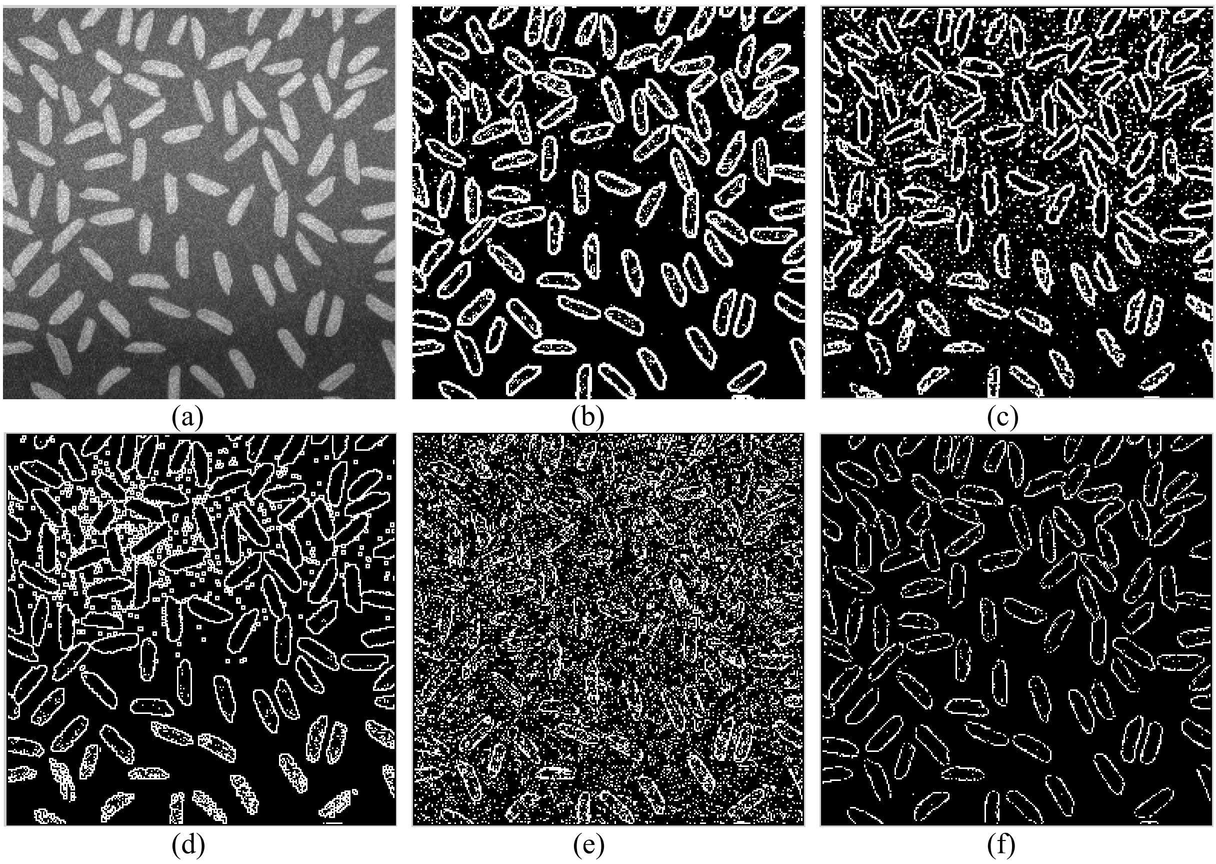



Fig. 7 Comparisons with different methods for image "Rice" with speckle noise (mean=0, variance=0.01)
(a) the original image; (b) Sobel; (c) method in paper[12]; (d) method in paper[13]; (e) HFED; (f) our method

Table 3 Processing time for various edge detectors /s

| | paper[12] | paper[13] | HFED | ours |
|---|---|---|---|---|
| Couple 512×512 | 0.73 | 0.69 | 1.95 | 0.74 |
| low contrast Lena 512×512 | 0.69 | 0.66 | 1.89 | 0.69 |
| blurry Lena 256×256 | 0.23 | 0.20 | 0.60 | 0.25 |
| Rice_ salt & pepper noise 256×256 | 0.20 | 0.18 | 0.58 | 0.21 |
| Rice_ speckle noise 256×256 | 0.25 | 0.23 | 0.63 | 0.25 |

## 4 Conclusions

To overcome the limitations of gradient-based edge detection methods and present fuzzy edge detection algorithms, we develop a novel edge detection method based on multiple features and fuzzy reasoning in this paper. More specially, they are achieved by introducing multiple features into new proposed fuzzy rules in the fuzzy reasoning process. The computer simulation results have shown the comparison of edge detection results among many different methods. Clearly, the proposed approach performs better than the previous methods not only in keeping low contrast and blurry edge details, but also in noisy images. Additionally, our fuzzy rules are significantly reduced compared with HFED.

## Acknowledgment

This research work is financially supported by the National Natural Science Foundation of China (91120006). I would also like to give my special thanks to the anonymous reviewers of this paper for their contributions to this work.

*References:*

[1] E. Gose, R. Johnsonbaug, S. Jost, *Pattern Recognition and Image Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[2] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing, Third Edition,* Publishing House of Electronics Industry, 2010.

[3] Musoromy Zuwena, Ramalingam Soodamani, Bekooy Nico, Edge detection comparison for License Plate detection, *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2010, pp. 1133-1138.

[4] Huang Jing, You Xinge, Tang Yuan Yan, A novel iris segmentation using radial suppression edge detection, *Signal Processing*, Vol. 89, No.12, 2009, pp. 2630-2643.

[5] Shamsoddini A, Trinder JC, Edge detection based filter for SAR speckle noise reduction, *International Journal of Remote Sensing*, Vol.7, No.33, 2012, pp. 2296-2320.

[6] Chen Shuhan, Shi Weiren, Wang Kai, Automatic edge detection using vector distance and partial normalization, *WSEAS Transactions on Computers,* Vol.10, No.9, 2011, pp. 301-309.

[7] D.S. Kim, W.H. Lee, I.S. Kweon, Automatic edge detection using 3×3 ideal binary pixel patterns and fuzzy-based edge thresholding, *Pattern recognition letters,* Vol.25, No.1, 2004, pp. 101-106.

[8] C. Ducottet, T. Fournel, C. Barat, Scale adaptive detection and local characterization of edges based on wavelet transform, *Signal Processing*, Vol.84, No.11, 2004, pp. 2115-2137

[9] JA Jiang, CL Chuang, YL Lu, Mathematical morphology-based edge detectors for detection of thin edges in low-contrast regions, *IET Image Processing,* Vol.1, No.3, 2007, pp. 269-277.

[10] Basturk, A, Gunay, E, Efficient edge detection in digital image using a cellular neural network optimized by differential evolution algorithm, *Expert systems with applications*, Vol.36, No.2, 2008, pp. 2645-2650.

[11] PAL S.K., KING R.A, On edge detection of X-ray images using fuzzy sets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No.1, 1983, pp. 69 -77.

[12] Yang Yong, Huang Shuying, Modified Pal and King algorithm for fuzzy edge detection, *Chinese Journal of Scientific Instrument*, Vol. 29, No.9, 2008, pp. 1918-1922.

[13] Zhang Jinping, Lian Yongxiang, Dong Linfu, A new method of fuzzy edge detection based on Gauss function, *International Conference on Computer and Automation Engineering (ICC AE)*, Vol.4, 2010, pp. 559-562.

[14] Jinbo Wu, Zhouping Yin, The fast multilevel fuzzy edge detection of blurry images, *IEEE Signal Processing Letters,* Vol.14, No.5, 2007, pp. 344-347.

[15] F. Russo, A new class of fuzzy operators for image processing: design and implementation,

*IEEE International Conference on Fuzzy Systems,* Vol.2, 1993, PP. 815-820.
[16] F. Russo, G. Ramponi, Fuzzy operator for sharpening of noisy images, *IEEE Electronics Letters,* Vol.28, No.18, 1992, pp. 1715-1717.
[17] H.R. Tizhoosh, Fast fuzzy edge detection, *Proceedings North American Fuzzy Information Processing Society,* 2002, pp. 239-242.
[18] L.R. Liang, C.G. Looney, Competitive fuzzy edge detection, *Applied Soft Computing*, Vol.3, No.2, 2003, pp. 123-137.
[19] Eghbal G. Mansoori, Hassan J. Eghbali, Heuristic edge detection using fuzzy rule-based classifier, *Journal of Intelligent and Fuzzy Systems*, Vol.17, No.5, 2006, pp. 457-469.
[20] Li Xin, Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis,* Prentice-Hall, Englewood Cliffs, NJ, 1994.
[21] Chung-Chia Kang, Wen-June Wang, Fuzzy reasoning-based directional median filter design, *Signal Processing,* Vol.89, No.3, 2009, pp. 344-351.
[22] Otsu N, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.SMC-9, No.1, 1979, pp. 62-65.