

# Extending the upper–lower edge detector by means of directional masks and OWA operators

C. Lopez-Molina · M. Galar · H. Bustince ·  
B. De Baets

Received: 12 September 2012 / Accepted: 28 September 2012 / Published online: 13 October 2012  
© Springer-Verlag Berlin Heidelberg 2012

**Abstract** Despite the fact that edges are a low-level feature, the edge detection process is likely to consist of several sub-processes or phases. The one taking the most attention from the research community is the feature extraction or computation, which consists of searching for hints of the presence of an edge at each pixel. In this work we propose an extension of the upper–lower edge detector by changing the way in which it produces the estimation of the presence of an edge. More specifically, we propose to obtain estimations of the intensity change in several orientations and then to combine them using ordered weighting aggregation operators.

**Keywords** Image processing · Edge detection · Upper–lower edge detector · OWA operator

## 1 Introduction

Edge detection is the process of locating the boundaries of the most relevant objects in an image. Despite the fact that edges are an intuitive concept, they can hardly be described in mathematical terms, to the point that some authors ground their work on the idea that edges are *the set of lines that*

*human observers would consent on to be the contours in that image* [29]. Having such lack of objective definitions, most of the edge detection methods focus on finding regions with evidences similar to those used by humans to distinguish boundaries on an image. Different objects are usually distinguishable because of having different tones, and hence locating their boundaries often consists of finding regions of the image where two or more tones coexist. Of course, the problem is greatly hardened by factors such as noise, shading or bad illumination, which can produce significant variations in the tone of an object, hindering the localization of its boundaries.

The soft computing community has taken a significant interest in the problem, mostly due to the inherent ambiguity of the problem, as well as to the large number of external, unpredictable factors [4, 18, 21]. Very different techniques have been used to solve the edge detection problem, from fuzzy rule-based systems such as the FIRE method [33] or the CFED [19] to bio-inspired systems (e.g. [12, 28]). Very usually these systems benefit from their ability to deal with imprecise and/or contradictory information. In this work we focus on the upper–lower edge detector (ULED) [9], which is based on the generation of an interval-valued representation of the image using t-norms and t-conorms [15]. More specifically, we propose to use such edge detector to produce multiple interpretations of the presence of an edge at each pixel, later aggregating the information using ordered weighting averaging (OWA) operators. The reason to employ that specific class of operators is the fact that they allow the adaptation of the relevance of each of the elements to be aggregated depending on their value.

The remainder of this paper is organized as follows: In Sect. 2 we recall some of the concepts that are later used in this work. Then, Sect. 3 includes the proposal and its algorithmic details. Section 4 contains the experimental

---

This work has been partially funded by the Spanish Ministry of Science, project TIN2010-15055, and by the Research Services of the Public University of Navarra.

---

C. Lopez-Molina (✉) · M. Galar · H. Bustince  
Dpto. Automática y Computación,  
Universidad Publica de Navarra,  
31006 Pamplona, Spain  
e-mail: carlos.lopez@unavarra.es

C. Lopez-Molina · B. De Baets  
Department of Mathematical Modeling, Statistics and Bioinformatics,  
Ghent University, Coupure links 653, 9000 Gent, Belgium  
e-mail: bernard.debaets@ugent.be

tests we have carried out, while conclusions are drawn in Sect. 5.

## 2 Preliminary definitions

This section recalls some basic definitions of the concepts used hereafter.

### 2.1 Triangular norms and conorms

**Definition 1** ([3,15]) A t-norm  $T : [0, 1]^2 \rightarrow [0, 1]$  is an associative, commutative, increasing function, such that  $T(1, x) = x$  for all  $x \in [0, 1]$ . A t-norm  $T$  is called idempotent if  $T(x, x) = x$  for all  $x \in [0, 1]$ .

**Definition 2** ([3,15]) A t-conorm  $S : [0, 1]^2 \rightarrow [0, 1]$  is an associative, commutative, increasing function such that  $S(0, x) = x$  for all  $x \in [0, 1]$ . A t-conorm  $S$  is called idempotent if  $S(x, x) = x$  for all  $x \in [0, 1]$ .

In-depth studies of t-norms and t-conorms together with some other aggregation functions can be found in [1,3,15].

### 2.2 Ordered weighted averaging aggregation operators

**Definition 3** ([41]) A function  $w : [0, 1]^n \rightarrow [0, 1]$  is called an OWA operator of dimension  $n$  if there exists a weighing vector  $h = (h_1, \dots, h_n) \in [0, 1]^n$  with  $\sum_i h_i = 1$ , and such that

$$w(a_1, \dots, a_n) = \sum_{j=1}^n h_j b_j \text{ with } b_j \text{ the } j\text{th largest of the } a_i \quad (1)$$

for any  $(a_1, \dots, a_n) \in [0, 1]^n$ .

Any OWA operator is completely defined by its weighing vector. In his original definition, Yager considered functions  $w$  defined on the whole Euclidean space  $\mathbb{R}^n$  and taking values in  $\mathbb{R}$ , but for our interest it is more appropriate to reduce this to  $[0, 1]^n$ . In this work we use the four-place OWA operators in Table 1.

**Table 1** OWA operators used in this work

Name	Weighing vector
$w_1$	$\{1, 0, 0, 0\}$
$w_2$	$\{\frac{1}{2}, \frac{1}{2}, 0, 0\}$
$w_3$	$\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0\}$
$w_4$	$\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$
$w_5$	$\{\frac{2}{3}, \frac{1}{3}, 0, 0\}$
$w_6$	$\{\frac{3}{6}, \frac{2}{6}, \frac{1}{6}, 0\}$
$w_7$	$\{\frac{4}{10}, \frac{3}{10}, \frac{2}{10}, \frac{1}{10}\}$

In the remainder of this work we consider images to have dimensions of  $X$  and  $Y$  pixels. For the sake of brevity, we consider  $P = \{1, \dots, X\} \times \{1, \dots, Y\}$  to be the set of their positions. Moreover, we denote by  $\mathbb{I}_Q$  to the set of all images of such resolution whose pixels take value in  $Q$  (i.e.  $I \in \mathbb{I}_Q$  if and only if for all  $(x, y) \in P$ ,  $I(x, y) \in Q$ ).

## 3 Directional upper–lower edge detector

### 3.1 On the use of directional information for edge detection

Edges are one-dimensional artifacts with straight interpretation for humans. However, their local or semi-local manifestation in the image can vary greatly. Typically, edges are related to tonal variations, which usually take place in a spatially organized way. Hence, edge features and indicators should be chosen so that they can handle this fact. Very different options arise in the literature for edge characterization. The gradient (vectorial representation of the intensity change) is a very usual indicator used for discriminating the edges. It stems from understanding the image as a continuous surface whose partial derivatives can be computed. In this context, edges are supposed to manifest as local maxima of the first derivative of the signal [39]. The local variation can be expressed in terms of horizontal and vertical changes, which combine for generating the gradient. Although there exist great diversity in the techniques used for the computation of the partial derivatives of an image [7,26,34,38], most of the authors agree on using the magnitude of such vector as an indicator of the presence of an edge at each pixel of the image. In addition to the magnitude of the gradient, some authors make use of its orientation to analyse the way in which the silhouette of the objects evolve.

Although the gradient-based representation is the most usual in the literature, some representative methods use alternative expressions of the local intensity change at each pixel of the image. One relevant example is the Laplacian-based approach, which consists of looking for zero-crossings on the second derivative of the image. This family of proposals is grounded in the fact that zero-crossings in the second derivative correspond spatially to the maxima in the first derivative, i.e. to the positions at which the gradients have locally maximal magnitudes. Although the most recognized method of such family is the Marr-Hildreth method, [24]<sup>1</sup> pioneering efforts using simpler approximations to the calculation of the Laplacian are already reported in, for example, the work by Prewitt [30]. Apart from Laplacian-based methods, popular

<sup>1</sup> The Marr-Hildreth method is sometimes referred to as Laplacian of a Gaussian (LoG) or difference of Gaussians (DoG) method because of the techniques used for filtering the image. Historical perspectives on using the second derivative of a signal for edge detection can be found in [2,22,23].

detectors such as the SUSAN method [36] or the FIRE one [33] also avoid the vectorial representation of the image variation. In this work, we focus on the upper–lower Edge Detector (ULED), which was recently presented by Barrenechea et al. [9] and is based on the generation of an interval-valued representation of the image using two operators, namely upper and lower constructors.

### 3.2 The upper–lower edge detector

In [5], Bustince et al. introduce the idea of constructing fuzzy edge images from the lengths of intervals of an interval-valued image (IV image). Then, in [9] the authors pose three different objectives:

- To define two new concepts, denoted as lower construction and upperconstruction, to construct a new interval-valued image ( $I \in \mathbb{I}_{L([0,1])}$ , where  $L([0, 1])$  represents the set of all possible intervals in  $[0, 1]$ ) from a given greyscale image ( $I \in \mathbb{I}_{[0,1]}$ ).
- To generate an edge image from an interval-valued image and
- To apply these theoretical developments to real images.

The upper and lower constructors are operators created to generate bright and dark versions of an image, respectively. Initially, they were designed to produce interval-valued representations of an image, since each of them provides one of the bounds of the interval to be assigned with each position in the image. The IV image generated with these constructors has several properties, among which we find the fact that positions around large intensity variations are assigned intervals with greater length than those in homogeneous regions. In this way, the length of the interval associated with each position of the image can be taken as an indicator of the edginess of the position, more precisely of its membership degree to the edges. Hence, a fuzzy edge image can be constructed with the lengths of the intervals at each position.

**Definition 4** Let  $I \in \mathbb{I}_{[0,1]}$  be a grayscale image of dimensions  $X$  and  $Y$ . Consider two t-norms  $T_1$  and  $T_2$  and two values  $n, m \in \mathbb{N}$  so that  $n \leq \frac{X-1}{2}$  and  $m \leq \frac{Y-1}{2}$ . A lower-construction associated with  $T_1, T_2, n$  and  $m$  is given by

$$L_{T_1, T_2}^{n, m} : \mathbb{I}_{[0,1]} \rightarrow \mathbb{I}_{[0,1]} \quad \text{given by}$$

$$L_{T_1, T_2}^{n, m}[I](i, j) = T_1 \left( T_2(I(i - u, j - v), I(i, j)) \right)_{\substack{u=-n \\ v=-m}}$$

for all  $(i, j) \in P$ , where the indices  $i, j$  take values such that  $0 \leq i - u \leq X - 1$  and  $0 \leq v - j \leq Y - 1$ . The values of  $n$  and  $m$  indicate that the considered window is a matrix of dimension  $(2n + 1) \times (2m + 1)$  centred at  $(i, j)$ . For the sake of simplicity, if  $n = m$ , then we denote  $L_{T_1, T_2}^{n, m}$  as  $L_{T_1, T_2}^n$ .

**Definition 5** Let  $I \in \mathbb{I}_{[0,1]}$ . Consider two t-conorms  $S_1$  and  $S_2$  and two values  $n, m \in \mathbb{N}$  such that  $n \leq \frac{X-1}{2}$  and  $m \leq \frac{Y-1}{2}$ . The upper constructor associated with  $S_1, S_2, n$  and  $m$  is defined as

$$U_{S_1, S_2}^{n, m} : \mathbb{I}_{[0,1]} \rightarrow \mathbb{I}_{[0,1]}, \quad \text{given by}$$

$$U_{S_1, S_2}^{n, m}[I](i, j) = S_1 \left( S_2(I(i - u, j - v), I(i, j)) \right)_{\substack{u=-n \\ v=-m}}$$

for all  $(i, j) \in P$  and where the indices  $i, j$  take values such that  $0 \leq i - u \leq X - 1$  and  $0 \leq j - v \leq Y - 1$ . The values of  $n$  and  $m$  indicate that the considered window is a matrix of dimension  $(2n + 1) \times (2m + 1)$  centred at  $(i, j)$ . For the sake of clarity, if  $n = m$ , then we denote  $U_{S_1, S_2}^{n, m}$  as  $U_{S_1, S_2}^n$ .

Let  $I \in \mathbb{I}_{[0,1]}$  and consider a lower constructor  $L_{T_1, T_2}^{n, m}$  and an upper constructor  $U_{S_1, S_2}^{n, m}$ . Then

$$L_{T_1, T_2}^{n, m}[I](i, j) \leq I(i, j) \leq U_{S_1, S_2}^{n, m}[I](i, j).$$

for all  $(i, j) \in P$ . Note that this implies that the images produced with upper and lower constructors can be used as boundaries for the creation of IV images.

*Remark 1* The definitions of lower constructor and upper constructor should not be confused with the fuzzy morphological operations of dilation and erosion [10], nor with erosion and dilation defined in classical mathematical morphology [13].

Let  $I \in \mathbb{I}_{[0,1]}$  and consider a lower constructor  $L_{T_1, T_2}^{n, m}$  and an upper constructor  $U_{S_1, S_2}^{n, m}$ . Then,  $I^{n, m} \in \mathbb{I}_{L([0,1])}$ , defined as

$$I^{n, m}(i, j) = [L_{T_1, T_2}^{n, m}(i, j), U_{S_1, S_2}^{n, m}(i, j)] \tag{2}$$

generates an interval-valued version of the image, that is, an image for which the value of each pixel is in  $L([0, 1])$  [14].

After building the interval valued image  $I^{n, m}$  from  $I$  the authors construct a fuzzy edge image  $F[I^{n, m}] \in \mathbb{I}_{[0,1]}$  so that

$$F[I^{n, m}](i, j) = \overline{I^{n, m}}(i, j) - \underline{I^{n, m}}(i, j) \tag{3}$$

$$F[I^{n, m}](i, j) = U_{S_1, S_2}^{n, m}[I](i, j) - L_{T_1, T_2}^{n, m}[I](i, j) \tag{4}$$

for all  $(i, j) \in P$ .

When using lower and upper constructors, the length of the interval associated with a position represents the intensity variation in its neighborhood. Then, in the construction of the fuzzy edge image, the length of the interval represents the membership degree of each element to the edges. Besides, from the definitions of upper and lower constructor we have that if the length associated with a pixel is maximal (i.e., in the window considered we have at least one white pixel and at least one black pixel), then the pixel is always considered an edge.

**Data:** An image  $I$ , two t-norms  $T_1, T_2$ , two t-conorms  $S_1, S_2$  and a neighbourhood window  $N$ .

**Result:** A fuzzy edge image  $F$

```

begin
  for each position  $(i, j)$  do
     $\alpha \leftarrow T_1_{(u,v) \in N} \left( T_2(I(i-u, j-v), I(i, j)) \right)$ ;
     $\beta \leftarrow S_1_{(u,v) \in N} \left( S_2(I(i-u, j-v), I(i, j)) \right)$ ;
     $F(i, j) \leftarrow \beta - \alpha$ ;
  end
end

```

**Algorithm 1:** Procedure for the Upper-Lower Edge Detector (ULED).

The main advantage of the ULED is that, depending on the lower and upper constructor we use, each pixel is associated with a different membership degree to the fuzzy edge images (corresponding to its interval length). This fact enables us to better adjust to the application in which we want to use the edge detection method. However, to narrow down the scope of the experiment, in the remainder of this work we only consider the upper and lower constructors based on the t-norm minimum ( $T_M$ ) and the t-conorm maximum ( $S_M$ ). The reason is that this couple of operators is the only one that guarantees that if the window centred at each  $(i, j)$  has a constant intensity, then the length of the associated interval is zero. Therefore, a pixel in a flat (plain tone) regions is never considered as part of an edge. The procedure of ULED is included in Algorithm 1.

### 3.3 Aggregating directional information

The ULED generates a scalar estimation of the amount of intensity variation at each pixel, which takes the shape of a membership degree to the edges. However, as explained in [9], non-squared masks can be used for the generation of directional information. In Fig. 2 we include the fuzzy edge image ( $W$ ) obtained using four different masks on the image in Fig. 1, each of them covering a different orientation.

Generally, edge detection methods consider the intensity changes in any possible direction, as long as there is no a priori knowledge stating otherwise. That is why, for example, the gradient is constructed with the intensity changes in orthogonal directions. However, we might only be interested in the direction in which the intensity change is stronger, simply because that is the one (presumably) perpendicular to the edge we are looking for. We can safely assume that the intensity change across the edge is stronger than that along the edge. In these conditions, it is the former that provides us the most trustworthy information about the intensity difference between the objects. In Fig. 3 we display an example of



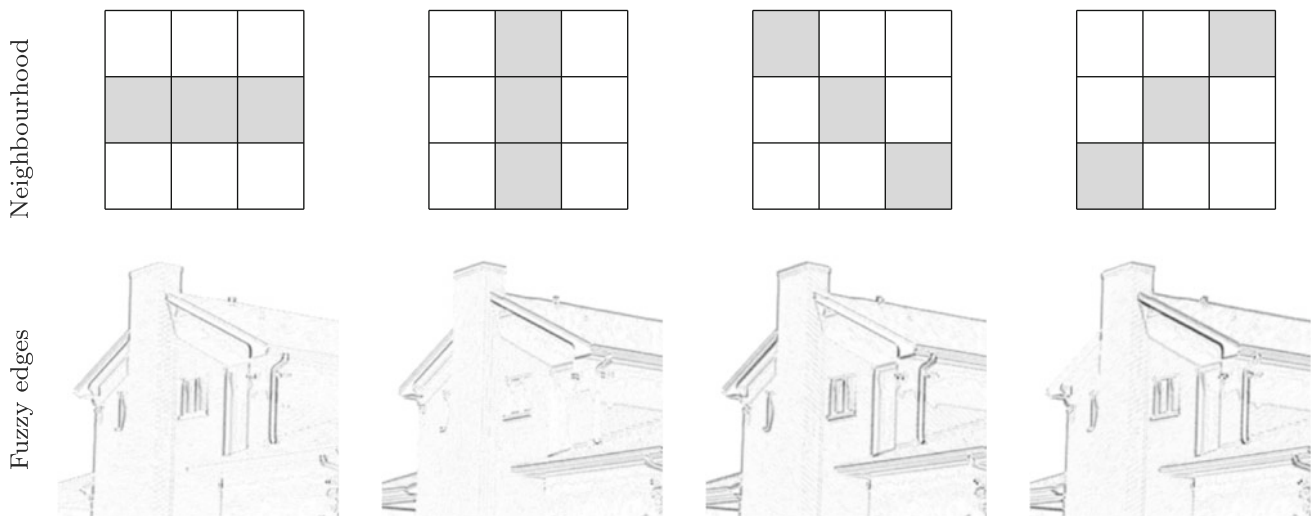
**Fig. 1** House image with resolution  $256 \times 256$

such situation, in which we are only interested in the intensity change taking place in the dominant orientation (in this case, horizontal), regardless of the one in the perpendicular one.

The idea of restricting edge detection to particular orientations (instead of considering them all) is not innovative, but has not been studied deeply in the literature. For example, Canny points out the need for adapting the size and orientation of the edge extraction filters to the specific conditions of each of the regions of the image [6, 7]. Although Canny did not develop that idea himself, other authors have attempted to do so. An interesting example is the detector by Shui et al. [35], in which Gaussian-based filters (identical to the ones proposed by Canny) are used with several different orientations, to later combine the results with those of the conventional, orthogonal operators. Note that the proper characterization of the underlying structure of an image has attracted the attention of the image processing community. Many authors propose to use descriptors other than the gradient, e.g. the structure tensor [16], although no application has been published so far for its application to edge characterization.

In this work, we use the upper and lower constructors to retrieve four intensity change estimations, one for each of the orientations (exactly as in Fig. 2). Then, we aggregate the information to produce a fuzzy representation of the edges. Initially, we considered the possibility of simply using the maximum operator, but the fact is that up to three different directions might partially cover up one of the axes (e.g., if considering a vertical edge, directions one, three and four in Fig. 2 are relevant). Hence, we use Ordered Weighted Averaging (OWA) operators, which allow us to provide each of the values different influence depending upon their values. Note that we only use decreasing-weight OWA operators, since we always consider that large values are more influential than small ones in our aggregation problem.

The algorithmic description of our proposal, which we denote directional ULED based on an OWA operator  $w$  (DULED- $w$ ), is in Algorithm 2. Note that, compared with



**Fig. 2** Example of fuzzy edge images obtained on the house image (Fig. 1) by the ULED using non-squared neighbourhoods with  $T_1 = T_2 = T_M$  and  $S_1 = S_2 = S_M$

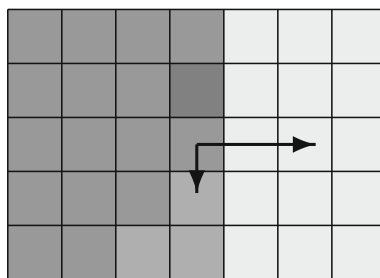
**Data:** An image  $I$ , two t-norms  $T_1, T_2$ , two t-conorms  $S_1, S_2$ , one neighbourhood size  $n$ , and an OWA operator  $w$ .

**Result:** A fuzzy edge image  $F$

```

begin
   $N_1, \dots, N_4 \leftarrow$  Create 4 directional neighbourhood masks of size  $n$ ;
  for  $i \in \{1, \dots, 4\}$  do
     $E_i \leftarrow$  ULED( $I, T_1, T_2, S_1, S_2, N_i$ );
  end
  for each position  $(i, j)$  do
     $F(i, j) \leftarrow w(E_1(i, j), E_2(i, j), E_3(i, j), E_4(i, j))$ ;
  end
end
    
```

**Algorithm 2:** Procedure for the Directional Upper-Lower Edge Detector using an OWA operator  $w$  (DULED- $w$ ).



**Fig. 3** Example of situation in which we are only interested in the dominant direction of the intensity change (in this case, horizontal). The magnitude of the black arrows represent the magnitude of the intensity change in the vertical and horizontal orientations

that of ULED, the computational complexity increases only up to a constant being now  $\mathcal{O}(4 \cdot |P| \cdot n^2) \approx \mathcal{O}(4 \cdot |P|)$  instead of  $\mathcal{O}(|P| \cdot n^2) \approx \mathcal{O}(|P|)$ . Since  $4 \ll |P|$ , the increase is not very significant.

As an example of the validity of our approach, we compare in Table 2 the membership degrees generated by ULED, DULED- $w$ , the Sobel operators [37] and the Roberts

**Table 2** Scalar edge estimation (membership degree to the edges) produced by different methods on the central pixel of the image in Fig. 3

Method	Edge estimation	Method	Edge estimation
ULED	0.50	DULED- $w_5$	0.40
DULED- $w_1$	0.40	DULED- $w_6$	0.39
DULED- $w_2$	0.40	DULED- $w_7$	0.37
DULED- $w_3$	0.38	Sobel	0.39
DULED- $w_4$	0.34	Roberts	0.46

operators [32] in the centre pixel of the image in Fig. 3. That is, the scalar estimation of the presence of an edge at that position, as performed by different methods.<sup>2</sup> In this image, the expected edge estimation at the pixels in the middle columns is 0.4, since the left and right regions have intensity 0.5 and 0.9, respectively. However, due to the influence of noisy pixels around the edge, the membership degree (or edginess) generated by each method greatly differs. Hence, in this case it is very positive to consider exclusively those orientations for which the intensity changes are maximal.

## 4 Experimental results

### 4.1 Aim of the experiment

In this section we aim to shed light on whether it is worth applying OWA operators to aggregate directional information instead of using square windows. In order to do

<sup>2</sup> Note that in the case of Roberts and Sobel operator such estimation corresponds to the Euclidean magnitude of the associated gradient



so, we compare the quality of the results of the DULED- $w$ , with several OWA operators  $w$ , with their equivalent version of the ULED, i.e. to the ULED using the same t-norms, t-conorms and window size  $n$ .

#### 4.2 Generating binary edge images

Both the ULED and the DULED- $w$  produce fuzzy edge images. In order to obtain binary (crisp) edges, some pre- and postprocessing of the image must be performed. First, prior to extracting the edge features, the images are smoothed using Gaussian filters. These filters are the most common in the literature, mainly because they offer a satisfactory trade-off between noise regularization and edge preservation [2, 40]. Then, to binary the fuzzy edge image generated by the ULED or DULED- $w$ , we have applied hysteresis [7] for binarization and morphological thinning [13, 17]. Note that the hysteresis makes use of two parameters, which are set using the automatic method by Medina-Carnicer et al. [27].

#### 4.3 Experimental dataset

In this experiment, we have used the Berkeley Segmentation Dataset (BSDS). This dataset offers a wide variety of natural images, together with several hand-made segmentations of each of them. Those segmentations can be considered as ideal solutions of the edge detection problem. It is our intention to test different edge detectors and see how close their results are to the ideal ones. The images in the BSDS have resolution  $321 \times 481$  or  $481 \times 321$  and are provided in greyscale. Each of them is associated with a set of five to nine binary human-made segmentations.

#### 4.4 Edge quality evaluation

Evaluating the performance of an edge detector is not a trivial task. In this work we compare the result produced by each of the edge detection methods with those produced by human experts. The comparison is carried out using two different measures: Baddeley's delta metric and precision/recall measures.

##### 4.4.1 Baddeley's delta metric (BDM)

This measure is designed for the comparison of binary images. It intends to measure the dissimilarity of the subset of featured points of two images, which are usually represented by a one value. Given the fact that images are usually displayed in a binary mode (following the constraints by Canny [7]), this measure can be used for our purposes. Let  $B_1$  and  $B_2$  be two binary images with the same dimensions  $X$  and

		Reality	
		Edge	Non-Edge
Classification	Edge	TP	FP
	Non-Edge	FN	TN

**Fig. 4** Confusion matrix for the edge detection problem

$Y$ . Given a value  $0 < k < \infty$  the  $k$ -BDM between the sets  $B_1$  and  $B_2$ , namely  $\Delta^k(B_1, B_2)$ , is defined as

$$\Delta^k(B_1, B_2) = \left[ \frac{1}{|P|} \sum_{p \in P} |\varphi(d(p, B_1)) - \varphi(d(p, B_2))|^k \right]^{\frac{1}{k}} \quad (5)$$

where  $d(p, B_i)$  represents a distance from the position  $p$  to the closest featured point of the set  $B_i$  and  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$  is a concave, increasing function used for weighing. Both sets must contain at least one featured element.

In our experiments, we use  $d(p, p') = \min(\|p - p'\|, 7)$ , where  $\|\cdot\|$  is the Euclidean magnitude. Then,  $d(p, B_i)$  stands for the minimum distance from the position  $p$  to an edge point of  $B_i$ . As for the  $\varphi$  function, we use  $\varphi(x) = \min(c, x)$  with  $c = \sqrt{X^2 + Y^2}$ , as in [27]. In practical terms, this means that we have  $\varphi(x) = x$ . To conclude, we have selected  $k = 2$ , as in [20, 27].

##### 4.4.2 Precision/recall

This approach is grounded in the fact that edge detection can be seen as a binary classification problem by Martin et al. [25]. Hence, it can be evaluated in terms of success and fall-out, comparing the output of an edge detection method with that generated by a human, which we can consider as ground truth. In this way, we build a confusion matrix such as the one in Fig. 4, with the elements in the main diagonal the ones correctly classified.

However, there are some considerations to be taken into account, due to the particular conditions of the edge detection problem; more specifically, the fact that edge images include spatial information embedded in the edges. It is reasonable to think that an edge displaced from its true position should not be penalized as much as if it would be completely missing. In order to solve this problem, we use a one-to-one pixel matching algorithm to map the edge pixels in the candidate edge image (generated by an edge detection method) and the ground truth ones. This matching allows for a certain spatial tolerance (in our case, as much as 1% of the diagonal of the image), so that an edge pixel can be slightly moved from its true position, yet counting as a correct classification. Note

**Table 3** Results obtained by the ULED using  $(T_1, T_2, S_1, S_2) = (T_M, T_M, S_M, S_M)$  and  $n = 3$ , compared to those by DULED with the same settings, and those by Sobel and FIRE methods.

	ULED	DULED-w							Sobel	FIRE
		w1	w2	w3	w4	w5	w6	w7		
$\sigma = 1.0$										
$\Delta^2$	3.13	3.06	3.10	3.09	3.09	3.09	3.10	3.09	3.04	3.02
Rank	5.8	3.5	4.6	3.9	4.4	4.5	5.0	4.3	–	–
B/W	10/41	45/14	11/8	21/16	2/5	3/4	2/8	6/4	–	–
P/R	0.40/0.84	0.40/0.83	0.40/0.84	0.40/0.85	0.40/0.84	0.40/0.85	0.40/0.85	0.40/0.84	0.43/0.82	0.43/0.77
F	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.53	0.52
$\sigma = 2.0$										
$\Delta^2$	2.63	2.57	2.58	2.57	2.58	2.58	2.58	2.58	2.63	2.55
Rank	5.9	4.3	4.4	3.4	4.1	4.4	4.7	4.8	–	–
B/W	11/49	23/13	4/3	37/11	11/8	4/4	3/6	7/6	–	–
P/R	0.52/0.73	0.53/0.72	0.53/0.73	0.53/0.73	0.52/0.72	0.52/0.73	0.52/0.73	0.52/0.73	0.52/0.72	0.53/0.63
F	0.58	0.58	0.59	0.59	0.58	0.58	0.58	0.58	0.58	0.55
$\sigma = 3.0$										
$\Delta^2$	2.43	2.37	2.36	2.35	2.36	2.36	2.36	2.36	2.42	2.36
Rank	6.0	4.5	4.3	3.8	4.5	4.2	4.3	4.3	–	–
B/W	10/54	18/9	8/6	33/12	6/7	11/7	7/2	7/3	–	–
P/R	0.56/0.64	0.57/0.64	0.57/0.64	0.58/0.63	0.57/0.64	0.57/0.64	0.57/0.64	0.57/0.64	0.57/0.63	0.57/0.54
F	0.58	0.58	0.59	0.58	0.58	0.58	0.58	0.58	0.57	0.53

that pixel matching has been extensively studied, but no consensus has been reached, since the result of each algorithm is strongly dependent on the conditions of the problem [8, 31]. In order to do the pixel-to-pixel matching we use the Cost Scaling Algorithm by Goldberg [11].

From the confusion matrix we extract the precision and recall evaluations, defined as

$$P_{REC} = \frac{TP}{TP + FP} \text{ and } R_{EC} = \frac{TP}{TP + FN}. \tag{6}$$

In this way, we evaluate two different facets of the problem: the accuracy (using  $P_{REC}$ ) and the fallout (using  $R_{EC}$ ).

#### 4.4.3 Experimental setup

In order to focus on the impact of the specific technique we propose, we restrict our experiments to neighbourhood masks of size three, five and seven. For each edge detector and image in the dataset, we compare the output with all of the hand-made solutions. Then, the evaluation of the method for that image is the triplet  $(\Delta^2, P_{REC}, R_{EC})$  produced by the hand-made solution having a lower  $\Delta^2$ .

#### 4.5 Results

Tables 3, 4, 5 contain the results of the evaluation of ULED and DULED-w, each of them considering a different

neighbourhood size. Three different values of  $\sigma$  are used in the preprocessing of the image. For each method and neighbourhood size we list the average  $\Delta^2$  in the 100 images of the BSDS *test* set, the average ranking (in terms of  $\Delta^2$ ), the number of images in the dataset for which it is the best and worst contender and the average  $P_{REC}$  and  $R_{EC}$ . For illustrative purposes, we also list the value of the F-measure, which is computed as the harmonic mean of  $P_{REC}$  and  $R_{EC}$ . Together with the results by the ULED and the DULED-w, we include those obtained in the same conditions by the popular Sobel [37] and FIRE [33], which are usually taken as reference works by their combination of performance and simplicity. In this case, we have combined the Sobel and FIRE operators for edge extraction with the pre- and postprocessing techniques in Sect. 4.2. Note that these methods do not take part in the computation B/W and Rank, in order not to hinder the interpretation of the results by ULED and DULED-w.

There are several conclusions to be made from the results of the experiment. First, we observe that the results gathered by the algorithms are as expected regarding the value of  $\sigma$ , since larger smoothing operators lead to higher precision and lower recall. Moreover, we observe that the differences between the results by the proposed method (DULED-w) and those by existing methods are minimal. In general, the Sobel method and the FIRE method perform slightly better than the ULED/DULED-w when they use small masks combined with low-smoothing conditions (e.g.  $n = 3$  and  $\sigma = 1.0$ ), but

**Table 4** Results obtained by the ULED using  $(T_1, T_2, S_1, S_2) = (T_M, T_M, S_M, S_M)$  and  $n = 5$ , compared to those by DULED with the same settings, and those by Sobel and FIRE methods

	ULED	DULED-w							Sobel	FIRE
		w1	w2	w3	w4	w5	w6	w7		
$\sigma = 1.0$										
$\Delta^2$	3.02	2.81	2.72	2.68	2.74	2.70	2.71	2.69	3.04	3.02
Rank	7.5	6.9	4.3	2.4	5.1	3.5	3.6	2.8	–	–
B/W	2/76	0/19	6/1	49/1	5/1	11/2	4/0	23/0	–	–
P/R	0.42/0.81	0.47/0.78	0.50/0.76	0.51/0.75	0.49/0.76	0.50/0.76	0.50/0.76	0.50/0.76	0.43/0.82	0.43/0.77
F	0.53	0.56	0.58	0.58	0.57	0.58	0.58	0.58	0.53	0.52
$\sigma = 2.0$										
$\Delta^2$	2.63	2.52	2.52	2.48	2.52	2.51	2.52	2.51	2.63	2.55
Rank	6.8	4.8	5.0	2.6	4.8	3.6	4.3	4.2	–	–
B/W	5/68	15/11	5/8	47/4	8/3	9/1	4/3	7/2	–	–
P/R	0.52/0.72	0.54/0.70	0.54/0.70	0.55/0.70	0.54/0.71	0.54/0.70	0.54/0.71	0.54/0.70	0.52/0.72	0.53/0.63
F	0.58	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.58	0.55
$\sigma = 3.0$										
$\Delta^2$	2.44	2.35	2.34	2.32	2.34	2.33	2.34	2.34	2.42	2.36
Rank	6.8	4.4	4.5	3.5	4.6	4.0	4.3	4.0	–	–
B/W	5/67	22/6	9/3	31/5	10/3	7/7	4/3	12/6	–	–
P/R	0.57/0.65	0.57/0.64	0.58/0.64	0.58/0.62	0.57/0.64	0.58/0.63	0.57/0.63	0.57/0.63	0.57/0.63	0.57/0.54
F	0.58	0.58	0.59	0.58	0.58	0.58	0.58	0.58	0.57	0.53

**Table 5** Results obtained by the ULED using  $(T_1, T_2, S_1, S_2) = (T_M, T_M, S_M, S_M)$  and  $n = 7$ , compared with those by DULED with the same settings, and those by Sobel and FIRE methods

	ULED	DULED-w							Sobel	FIRE
		w1	w2	w3	w4	w5	w6	w7		
$\sigma = 1.0$										
$\Delta^2$	3.02	2.82	2.58	2.52	2.61	2.55	2.55	2.54	3.04	3.02
Rank	7.5	7.1	4.7	1.9	5.7	3.1	3.4	2.6	–	–
B/W	3/74	0/24	3/0	59/1	3/1	10/0	3/0	19/0	–	–
P/R	0.42/0.81	0.45/0.77	0.52/0.68	0.54/0.67	0.52/0.69	0.53/0.67	0.53/0.67	0.53/0.67	0.43/0.82	0.43/0.77
F	0.53	0.54	0.56	0.57	0.56	0.57	0.57	0.57	0.53	0.52
$\sigma = 2.0$										
$\Delta^2$	2.63	2.45	2.42	2.38	2.42	2.40	2.41	2.41	2.63	2.55
Rank	7.2	5.7	4.8	2.4	4.7	3.5	4.0	3.8	–	–
B/W	6/75	7/10	5/5	50/2	5/2	11/3	3/0	13/3	–	–
P/R	0.52/0.72	0.55/0.67	0.56/0.66	0.57/0.65	0.56/0.66	0.56/0.66	0.56/0.65	0.56/0.66	0.52/0.72	0.53/0.63
F	0.58	0.58	0.59	0.59	0.59	0.59	0.58	0.58	0.58	0.55
$\sigma = 3.0$										
$\Delta^2$	2.44	2.31	2.31	2.28	2.31	2.30	2.31	2.31	2.42	2.36
Rank	6.9	4.3	4.6	3.2	4.5	3.6	4.3	4.6	–	–
B/W	6/70	20/8	2/5	40/4	9/4	11/3	5/3	7/3	–	–
P/R	0.57/0.65	0.58/0.61	0.58/0.62	0.59/0.60	0.58/0.62	0.58/0.61	0.57/0.61	0.58/0.61	0.57/0.63	0.57/0.54
F	0.58	0.58	0.58	0.57	0.58	0.57	0.58	0.57	0.57	0.53



are outperformed when such masks are enlarged (see  $n = 7$ ,  $\sigma = 1.0$ ).

Regarding the utility of the aggregation of directional information, we find that the DULED- $w$  always outperforms the corresponding ULED in terms of average  $\Delta^2$ . The improvement obtained with DULED- $w$  depends on the specific  $w$  and  $\sigma$ , but in general we observe that the most influential factor is the size of the neighbourhood mask used by the upper and lower constructors. When  $n = 3$ , both algorithms retrieve similar results, while the difference between them increases together with  $n$ . This is due to the fact that, opposed to what happens with large, squared neighbourhood masks, the small ones are not prone to include non-relevant pixels with pernicious influence in the characterization of the edges. The ULED is also outperformed in terms of Rank., what indicates that DULED- $w$  is consistently better than its counterpart. In the same way, we find that most of the worst possible results among the pool of candidates are due to the ULED, while it rarely produces the best edge image. We must note, however, that ULED is competitive in terms of precision and recall, especially in the latter. It often happens that the ULED obtains better recall than its comparable DULED- $w$ , although this usually comes together with lower precision. This indicates that the ULED excels in avoiding false negatives, but at the same time generates a larger number of false positives.

Regarding the use of different OWA operators, there are no clear conclusions to be taken. Although we cannot perceive significant differences between them, there are two findings worth noting:

- The OWA operators using less values (e.g.  $w_1$ ) are prone to produce the worst possible results. This is due to the fact that the DULED- $w$  is then more sensitive to the presence of noise. Indeed, one single noisy pixel can make a big difference, especially when light (low  $\sigma$ ) Gaussian smoothing is used.
- The OWA operators using a great number of values in the aggregation (e.g.  $w_7$ ) hardly ever obtain the best possible results, and this happens mostly when  $\sigma$  is low. The reason is that, as stated before, strong (great  $\sigma$ ) Gaussian regularization reduces noise more effectively than the light one. Hence, OWA operators using a greater number of values do not take advantage from their robustness against noise.

## 5 Conclusions

In this work we have introduced a technique for generating scalar image variation interpretation from the aggregation of directional information. This technique has been tested in combination with the upper–lower edge Detector, assuming

that it is able to improve the original technique, which produced the scalar information without considering any directional feature. Moreover, it constitutes a pioneering effort for using well-studied information aggregation techniques for edge feature synthesis.

## References

1. Alsina, C., Frank, M.J., Schweizer, B.: Associative functions: triangular norms and copulas. World Scientific Publishing Company, Singapore (2006)
2. Barrenechea, E., Bustince Sola, H., De Baets, B., Lopez-Molina, C.: Construction of interval-valued fuzzy relations with application to the generation of fuzzy edge images. *IEEE Trans. Fuzzy Syst.* **19**(5), 819–830 (2011)
3. Basu, M.: Gaussian-based edge-detection methods—a survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **32**(3), 252–260 (2002)
4. Beliakov, G., Pradera, A., Calvo, T.: Aggregation functions: a guide for practitioners, studies in fuzziness and soft computing, vol. 221. Springer, Berlin (2007)
5. Bezdek, J., Chandrasekhar, R., Attikouzel, Y.: A geometric approach to edge detection. *IEEE Trans. Fuzzy Syst.* **6**(1), 52–75 (1998)
6. Bustince, H., Barrenechea, E., Pagola, M., Fernandez, J.: Interval-valued fuzzy sets constructed from matrices: application to edge detection. *Fuzzy Sets Syst.* **160**(13), 1819–1840 (2009)
7. Canny, J.: Finding edges and lines in images. Technical report, Massachusetts Institute of Technology, Cambridge (1983)
8. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 679–698 (1986)
9. De Baets, B.: Fuzzy techniques in image processing, chap. Generalized idempotence in fuzzy mathematical morphology, pp. 58–75. Physica-Verlag, New York (2000)
10. Galar, M., Fernandez, J., Beliakov, G., Bustince, H.: Interval-valued fuzzy sets applied to stereo matching of color images. *IEEE Trans. Image Process.* **20**(7), 1949–1961 (2011)
11. Goldberg, A.V., Kennedy, R.: An efficient cost scaling algorithm for the assignment problem. *Math. Program.* **71**, 153–177 (1995)
12. Gudmundsson, M., El-Kwae, E., Kabuka, M.: Edge detection in medical images using a genetic algorithm. *IEEE Trans. Med. Imaging* **17**(3), 469–474 (1998)
13. Haralick, R.M., Sternberg, S.R., Zhuang, X.: Image analysis using mathematical morphology. *IEEE Trans. Pattern Anal. Mach. Intell.* **9**(4), 532–550 (1987)
14. Jurio, A., Paternain, D., Lopez-Molina, C., Bustince, H., Mesiar, R., Beliakov, G.: A construction method of interval-valued fuzzy sets for image processing. In: Proceedings of the IEEE symposium on advances in type-2 fuzzy logic systems (2011)
15. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. Kluwer Academic Publishers, Dordrecht (2000)
16. Köthe, U.: Pattern recognition, lecture notes in computer science, vol. 2781. In: Edge and junction detection with an improved structure tensor. pp. 25–32. Springer, Berlin (2003)
17. Lam, L., Lee, S.W., Suen, C.: Thinning methodologies—a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(9), 869–885 (1992)
18. Law, T., Itoh, H., Seki, H.: Image filtering, edge detection, and edge tracing using fuzzy reasoning. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(5), 481–491 (1996)
19. Liang, L.R., Looney, C.G.: Competitive fuzzy edge detection. *Appl. Soft Comput.* **3**(2), 123–137 (2003)
20. Lopez-Molina, C., Bustince, H., Fernandez, J., Couto, P., De Baets, B.: A gravitational approach to edge detection based on triangular norms. *Pattern Recognit.* **43**(11), 3730–3741 (2010)

21. Lopez-Molina, C., De Baets, B., Bustince, H.: Generating fuzzy edge images from gradient magnitudes. *Comput. Vis. Image Underst.* **115**(11), 1571–1580 (2011)
22. Mahmoodi, S.: Edge detection filter based on Mumford-Shah green function. *SIAM J. Imaging Sci.* **5**(1), 343–365 (2012)
23. Mallat, S., Zhong, S.: Characterization of signals from multiscale edges. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(7), 710–732 (1992)
24. Marr, D., Hildreth, E.: Theory of edge detection. *Proc. R. Soc. Lond.* **207**(1167), 187–217 (1980)
25. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(5), 530–549 (2004)
26. McIlhagga, W.: The Canny edge detector revisited. *Int. J. Comput. Vis.* **91**, 251–261 (2011)
27. Medina-Carnicer, R., Madrid-Cuevas, F., Carmona-Poyato, A., Muñoz-Salinas, R.: On candidates selection for hysteresis thresholds in edge detection. *Pattern Recognit.* **42**(7), 1284–1296 (2009)
28. Nezamabadi-pour, H., Saryazdi, S., Rashedi, E.: Edge detection using ant algorithms. *Soft Comput.* **10**, 623–628 (2006)
29. Papari, G., Petkov, N.: Edge and line oriented contour detection: state of the art. *Image Vis. Comput.* **29**(2–3), 79–103 (2011)
30. Prewitt, J.M.S.: Object enhancement and extraction. In: *Picture Processing and Psychopictorics*. pp. 75–149. Academic Press, New York (1970)
31. Rabin, J., Delon, J., Gousseau, Y.: A statistical approach to the matching of local features. *SIAM J. Imaging Sci.* **2**(3), 931–958 (2009)
32. Roberts, L.G.: Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology (1963)
33. Russo, F., Ramponi, G.: Edge extraction by FIRE operators. In: *IEEE Conference on Fuzzy Systems*, vol. 1, pp. 249–253, Orlando (1994)
34. Shen, J., Castan, S.: An optimal linear operator for step edge detection. *CVGIP: Gr. Models Image Process.* **54**(22), 112–133 (1992)
35. Shui, P.L., Zhang, W.C.: Noise-robust edge detector combining isotropic and anisotropic Gaussian kernels. *Pattern Recognit.* **45**(2), 806–820 (2012)
36. Smith, S.M., Brady, J.M.: SUSAN—a new approach to low level image processing. *Int. J. Comput. Vis.* **23**, 45–78 (1997)
37. Sobel, I., Feldman, G.: A  $3 \times 3$  isotropic gradient operator for image processing. In: Presented at a talk at the Stanford Artificial Intelligence Project (1968)
38. Sun, G., Liu, Q., Liu, Q., Ji, C., Li, X.: A novel approach for edge detection based on the theory of universal gravity. *Pattern Recognit.* **40**(10), 2766–2775 (2007)
39. Torre, V., Poggio, T.: On edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(2), 147–163 (1984)
40. Witkin, A.P.: Scale-space filtering. In: *Proceedings of the international joint conference on artificial intelligence*, vol. 2, pp. 1019–1022. Karlsruhe, Germany (1983)
41. Yager, R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.* **18**(1), 183–190 (1988)