

Image feature extraction using OD-monotone functions

Cedric Marco-Detchart^{1,2}, Carlos Lopez-Molina¹, Javier Fernandez^{1,2}, Miguel Pagola^{1,2}, and Humberto Bustince^{1,2}

¹ Dpto. Automatica y Computacion, Universidad Publica de Navarra,
Campus Arrosadia, 31006 Pamplona, Spain

² Institute of Smart Cities, Universidad Publica de Navarra,
Campus Arrosadia, 31006 Pamplona, Spain
`cedric.marco@unavarra.com`

Abstract. Edge detection is a basic technique used as a preliminary step for, *e.g.*, object extraction and recognition in image processing. Many of the methods for edge detection can be fit in the breakdown structure by Bezdek, in which one of the key parts is feature extraction. This work presents a method to extract edge features from a grayscale image using the so-called ordered directionally monotone functions. For this purpose we introduce some concepts about directional monotonicity and present two construction methods for feature extraction operators. The proposed technique is competitive with the existing methods in the literature. Furthermore, if we combine the features obtained by different methods using penalty functions, the results are equal or better results than state-of-the-art methods.

Keywords: Edge detection, feature extraction, ordered directionally monotone functions, penalty functions

1 Introduction

Most applications of image processing require, at some stage, automatic identification of objects on an image. In order to recognize objects, some sort of hints has to be extracted from the original image. One of the most useful hints is the boundaries of the own objects, which lead to a vast amount of studies on edge detection. The main difficulty in edge detection arises from the inherent process of capturing real word information; even the definition of edge itself is a fuzzy concept. An edge can be regarded as a location in which a big enough jump between neighbour pixel intensities happen. However, even this basic, loose definition can be criticized, since it would not consider situations as textures (in which no edges should be appointed) or *hallucinated boundaries* [21] (in which boundaries appear with little or no intensity contrast).

A great variety of edge detection methods are based on gradients, often computed by the convolution of a filter and an image. Examples are Sobel [25], Prewitt [23] and Canny [9]. Usually considered as a unique operation, edge detection is in fact reached through a sequence of operations.

These sequence of operations has been studied by several authors, mainly attempting to develop a standard, multi-phase structure for edge detection [26]. For example, Law *et al.* [15] proposed a process consisting of three steps: *filtering*, *detection* and *tracing*. A few years later, Bezdek *et al.*[4] presented a framework to encompass a variety of methods in the literature (mainly those based on gradient extraction), proposing a process made up of four phases: *conditioning*, *feature extraction*, *blending* and *scaling*.

Although we stick to the Bezdek Breakdown Structure (BSS) for our experimentation [16], our work focuses on the *feature extraction* phase, *i.e.*, the way in which visual information at each pixel is converted into problem-specific information. This problem-specific information usually reduces to gradient magnitudes, but we also consider gradient direction. Moreover, we build feature maps considering information of pixel neighbourhood and fuse all the information using Ordered Directionally Monotone (ODM) functions. These functions are monotonic along different directions over the decreasingly ordered input vector. Furthermore, we test our method in combination with other techniques using penalty functions.

The structure of the document is as follows. Section 2 is devoted to explain some mathematical concepts applied to image processing, as well as some notions about aggregation theory to introduce ODM functions. Section 3 describes our proposal. In Section 4 an application to edge detection is presented, along with some preliminary results. Finally, in Section 5 we expose conclusions and future work.

2 Preliminaries

Given an image \mathbb{I}_L , we consider it as a grid of elements, the set of positions being $D = R \times C = \{1, \dots, r\} \times \{1, \dots, c\}$, where r represents the number of rows of the image, *i.e.*, the height, and c the columns of the image, *i.e.*, the width. Each of the elements (pixels) in the grid takes values in a scale L . The use of L permits us to represent different types of images. In the case of binary images pixels will take values in $L = \{0, 1\}$, whereas grey-scale images take $L = \{0, \dots, 255\}$, and colour images in the RGB colour space are in the range $L = \{0, \dots, 255\}^3$. This work has been carried out using grey-scale images.

As this work is related to the theory of aggregation and monotonicity we expose some of the necessary theory.

Definition 1. [2, 8] *A mapping $M : [0, 1]^n \rightarrow [0, 1]$ is an aggregation function if it is monotone non-decreasing in each of its components and satisfies $M(\mathbf{0}) = 0$ and $M(\mathbf{1}) = 1$, with $\mathbf{0} = (0, \dots, 0)$ and $\mathbf{1} = (1, \dots, 1)$*

An aggregation function M is an averaging or mean if

$$\min(x_1, \dots, x_n) \leq M(x_1, \dots, x_n) \leq \max(x_1, \dots, x_n).$$

If an aggregation function is averaging then it is idempotent, and the converse is also true.

There are many types and classes of aggregation functions in the literature [13, 3], where some of them have appeared to address specific applications. In this work we make use of Ordered Weighted Averaging (OWA) operators, defined by Yager [29].

Definition 2. An OWA operator of dimension n is a mapping $\Phi : [0, 1]^n \rightarrow [0, 1]$ such that it exists a weighting vector $\mathbf{w} = (w_1, \dots, w_n) \in [0, 1]^n$ with $\sum_{i=1}^n w_i = 1$, and such that

$$\Phi(x_1, \dots, x_n) = \sum_{i=1}^n w_i \cdot x_{\sigma(i)},$$

where $\mathbf{x}_\sigma = (x_{\sigma(1)}, \dots, x_{\sigma(n)})$ is a decreasing permutation on the input \mathbf{x} .

Another relevant family of aggregation functions used in this work are Choquet integrals. Particularly, we consider the discrete Choquet integral, related to fuzzy measures which are defined on finite dimensional spaces.

Definition 3. Let $N = \{1, 2, \dots, n\}$. A function $\mathbf{m} : 2^N \rightarrow [0, 1]$ is a fuzzy measure if, for all $X, Y \subseteq N$, it satisfies the following properties:

- (m1) *Increasingness:* if $X \subseteq Y$, then $\mathbf{m}(X) \leq \mathbf{m}(Y)$;
- (m2) *Boundary conditions:* $\mathbf{m}(\emptyset) = 0$ and $\mathbf{m}(N) = 1$.

Definition 4. [3, 13] Let $\mathbf{m} : 2^N \rightarrow [0, 1]$ be a fuzzy measure. The discrete Choquet integral is the function $C_{\mathbf{m}} : [0, 1]^n \rightarrow [0, 1]$, defined, for each $\mathbf{x} \in [0, 1]^n$, by

$$C_{\mathbf{m}}(\mathbf{x}) = \sum_{i=1}^n (x_{(i)} - x_{(i-1)}) \cdot \mathbf{m}(A_{(i)}),$$

where $(x_{(1)}, \dots, x_{(n)})$ is an increasing permutation on the input \mathbf{x} , that is, $x_{(1)} \leq \dots \leq x_{(n)}$, with the convention that $x_{(0)} = 0$, and $A_{(i)} = \{(i), \dots, (n)\}$ is the subset of indices of the $n - i + 1$ largest components of \mathbf{x} .

Monotonicity can sometimes be too restrictive for some applications, excluding a large family of non-monotonic averaging functions, *e.g.*, the mode (most frequent element) is not a monotonic function though it is widely used in certain applications like image filtering [27]. From this observation Wilkin and Beliakov [28] introduced the notion of weak monotonicity.

This definition was later extended into the notion of directional monotonicity [6].

Definition 5. [6] Let $\vec{r} = (r_1, \dots, r_n)$ be a real n -dimensional vector, $\vec{r} \neq \mathbf{0}$. A function $F : [0, 1]^n \rightarrow [0, 1]$ is \vec{r} -increasing if for all points $(x_1, \dots, x_n) \in [0, 1]^n$ and for all $c > 0$ such that $(x_1 + cr_1, \dots, x_n + cr_n) \in [0, 1]^n$ it holds

$$F(x_1 + cr_1, \dots, x_n + cr_n) \geq F(x_1, \dots, x_n).$$

That is, a \vec{r} -increasing function is a function which is increasing along the ray (direction) determined by the vector \vec{r} .

From this concept of directional monotonicity, we come to ordered directionally monotone functions, where the direction along which monotonicity is required varies depending on the relative size of the coordinates of the considered input.

Definition 6. [5] Let $F : [0, 1]^n \rightarrow [0, 1]$ be a function and let $\vec{r} \neq \mathbf{0}$. F is said to be ordered directionally (OD) \vec{r} -increasing if for any $\mathbf{x} \in [0, 1]^n$, for any $c > 0$ and for any permutation $\sigma \in S_n$ with $x_{\sigma(1)} \geq \dots \geq x_{\sigma(n)}$ and such that

$$1 \geq x_{\sigma(1)} + cr_1 \geq \dots \geq x_{\sigma(n)} + cr_n \geq 0,$$

it holds that

$$F(\mathbf{x} + c\vec{r}_{\sigma^{-1}}) \geq F(\mathbf{x}),$$

where $\vec{r}_{\sigma^{-1}} = (r_{\sigma^{-1}(1)}, \dots, r_{\sigma^{-1}(n)})$.

As we are also building consensus feature images we need to introduce the concept of penalty functions.

Given a set of n numerical values x_1, \dots, x_n and q averaging aggregation functions M_1, \dots, M_q , these functions allow us to select, between the q functions, the one that provides the least dissimilar output to all the inputs.

Taking into account the previous considerations, in this work we consider the following definition of a penalty function in a Cartesian product of lattices [7]:

Definition 7. A function $P_{\nabla} : ([0, 1]^n)^m \times [0, 1]^m \rightarrow [0, \infty[$ is a penalty function if, for every $\mathbf{X} = (\mathbf{x}^1, \dots, \mathbf{x}^m) \in ([0, 1]^n)^m$ (with $\mathbf{x}^i = (x_1^i, \dots, x_n^i)$ for every $i \in \{1, \dots, m\}$) and for every $\mathbf{y} = (y_1, \dots, y_m) \in [0, 1]^m$, it satisfies that:

1. $P_{\nabla}(\mathbf{X}, \mathbf{y}) \geq 0$;
2. $P_{\nabla}(\mathbf{X}, \mathbf{y}) = 0$ if and only if $x_1^i = \dots = x_n^i = y^i$ for every $i \in \{1, \dots, m\}$;
3. P_{∇} is convex in y_i or every $i \in \{1, \dots, m\}$.

3 Ordered Directionally Monotone Functions

In this section we present two alternative methods for obtaining ODM functions. Firstly we propose an affine construction and secondly we use the Choquet integral.

Theorem 1. Let $G : [0, 1]^n \rightarrow [0, 1]$ be defined, for $\mathbf{x} \in [0, 1]^n$ and $\sigma \in S_n$ such that $x_{\sigma(1)} \geq \dots \geq x_{\sigma(n)}$, by

$$G(\mathbf{x}) = a + \sum_{i=1}^n b_i x_{\sigma(i)},$$

for some $a \in [0, 1]$ and $\vec{b} = (b_1, \dots, b_n) \in \mathbb{R}^n$ such that $0 \leq a + b_1 + \dots + b_j \leq 1$ for all $j \in \{1, \dots, n\}$. Then G is OD \vec{r} -increasing for every non-null vector \vec{r} such that $\vec{b} \cdot \vec{r} \geq 0$. In particular, for every non-null vector \vec{r} which is orthogonal to \vec{b} .

Theorem 1 can be generalized taking into account the following lemma.

Lemma 1. [5] *Let $\varphi : [0, 1] \rightarrow [0, 1]$ be an automorphism (i.e., an increasing bijection). Then, if $G : [0, 1]^n \rightarrow [0, 1]$ is an ordered directionally increasing function, the function $\varphi \circ G$ is also an ordered directionally increasing function.*

Corollary 1. *Let $p > 0$. Let $G : [0, 1]^n \rightarrow [0, 1]$ be defined, for $\mathbf{x} \in [0, 1]^n$ and $\sigma \in S_n$ such that $x_{\sigma(1)} \geq \dots \geq x_{\sigma(n)}$, by*

$$G(\mathbf{x}) = \left(a + \sum_{i=1}^n b_i x_{\sigma(i)} \right)^{\frac{1}{p}}, \quad (1)$$

for some $a \in [0, 1]$ and $\vec{b} = (b_1, \dots, b_n) \in \mathbb{R}^n$ such that $0 \leq a + b_1 + \dots + b_j \leq 1$ for all $j \in \{1, \dots, n\}$. Then G is OD \vec{r} -increasing for every non-null vector \vec{r} such that $\vec{b} \cdot \vec{r} \geq 0$.

Proof. It follows from Lemma 1, since the function $\varphi(x) = x^{\frac{1}{p}}$ is an automorphism. ■

From Theorem 1, the following two corollaries are straight.

Corollary 2. *Let \mathbf{m} be a fuzzy measure. Then the Choquet integral*

$$C_{\mathbf{m}}(\mathbf{x}) = \sum_{i=1}^n (x_{(i)} - x_{(i-1)}) \cdot \mathbf{m}(A_{(i)}),$$

is OD \vec{r} -monotone for every non-null n -dimensional vector \vec{r} such that, for all maximal chains $A_1 \supset A_2 \supset \dots \supset A_n \supset A_{n+1} = \emptyset$, it holds that

$$\sum_{i=1}^n r_{n-i+1} (\mathbf{m}(A_{(i)}) - \mathbf{m}(A_{(i+1)})) \geq 0,$$

where $\mathbf{m}(A_{(n+1)}) = 0$.

Proof.

The result follows from Theorem 1 after noting that the Choquet integral can be rewritten as

$$C_{\mathbf{m}}(\mathbf{x}) = \sum_{i=1}^n (\mathbf{m}(A_{(i)}) - \mathbf{m}(A_{(i-1)})) \cdot x_{(i)}.$$

■

Corollary 3. *Let $A : [0, 1]^n \rightarrow [0, 1]$ be an OWA operator associated to the weighting vector $\mathbf{w} = (w_1, \dots, w_n)$. Then A is OD \vec{r} -increasing for every non-null vector \vec{r} such that $\mathbf{w} \cdot \vec{r} \geq 0$.*

4 Experimental study

The aim of this experiment is to test the effectiveness of ODM functions to detect edges in images. To do so, we propose two possible constructions for ODM functions to extract feature images and then compare them to well-known methods in the literature. We also use the feature images obtained to build consensus solutions by means of penalty functions.

4.1 Proposed method and parameters

Given a grey-scale image \mathbb{I}_g the primary step is to normalize the intensity values to the range $[0, 1]$. Then we use Algorithm 1 to obtain a feature image by means of ODM functions.

Algorithm 1 Constructing an edge feature image using ODM functions

Input: A normalized grey-scale image \mathbb{I}_g and an ODM function G as in Corollary 1.

Output: A feature image \mathbb{I}_f .

- 1: **for** each pixel (x, y) of \mathbb{I}_g **do**
 - 2: Compute the corresponding values by means of the absolute value of the difference between $\mathbb{I}_g(x, y)$ and its 8-neighbourhood;
 - 3: Order the eight values of step 2 in a decreasing way;
 - 4: Apply the ODM function G , with its corresponding a, p values and \vec{r}, \vec{b} vectors (see Eq. (1)), to the values obtained in step 3;
 - 5: Assign as intensity of the pixel (x, y) of \mathbb{I}_f the value obtained in step 4.
 - 6: **end for**
-

As we consider an 8-neighbourhood around each position (x, y) (*i.e.* neighbours from $(x - 1, y - 1)$ to $(x + 1, y + 1)$). Each value needed for step 2 is computed as:

$$x_1 = |a_{(x,y)} - a_{(x-1,y-1)}|, \dots, x_8 = |a_{(x,y)} - a_{(x+1,y+1)}|$$

for each of the values.

In step 3, these differences are ordered in a decreasing way; that is,

$$x_{\sigma(1)} \geq x_{\sigma(2)} \geq \dots \geq x_{\sigma(7)} \geq x_{\sigma(8)}.$$

Finally, in step 4 an ODM function is applied to each position in the image, with different a, p, \vec{r} and \vec{b} parameters.

To test our method we propose two expressions to build ODM functions for step 4 of Algorithm 1. We construct these expressions by means of Corollary 1 using Eq. (1) and choosing specific values for the specified parameters.

Elaborating on the expression in Eq. (1), the parameter p adjusts the brightness level of the feature image. In this way, the image is lighter when $p > 1$ or

darker if $0 < p < 1$ [11]. Then, we consider $\vec{r} = (x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(s)})$; and for the first ODM construction (*Case 1*) we use the following \vec{b} :

$$\vec{b} = \left(\frac{x_{\sigma(1)}}{2 \sum_{i=1}^8 x_{\sigma(i)}}, \frac{x_{\sigma(2)}}{2 \sum_{i=1}^8 x_{\sigma(i)}}, \dots, \frac{x_{\sigma(s)}}{2 \sum_{i=1}^8 x_{\sigma(i)}} \right)$$

Considering the second proposed construction (*Case 2*) we maintain the same previous parameters and change \vec{b} as follows:

$$\vec{b} = \left(\frac{|x_{\sigma(1)} - x_{\sigma(s)}|}{2 \sum_{i=1}^8 |x_{\sigma(i)} - x_{\sigma(s)}|}, \dots, \frac{|x_{\sigma(\tau)} - x_{\sigma(s)}|}{2 \sum_{i=1}^8 |x_{\sigma(i)} - x_{\sigma(s)}|}, 0 \right)$$

Regarding the value of $\frac{1}{p}$ we take for both cases $\frac{1}{p} = 0.35$ and $a = 0$. Choosing $a = 0$ is because we need $G(\mathbf{0}) = 0$ if and only if it takes this value. And then having $a = 0$, $G(\mathbf{1}) = 1$ if and only if $b_1 + \dots + b_n = 1$. Note that in the case of a flat region in the image, *i.e.*, when all the pixels have the same value we would obtain a zero denominator, so we mark directly the corresponding position in the feature image as not containing an edge.

Figure 1 displays the results obtained by applying our proposed algorithm with the two alternative construction methods for ODM.

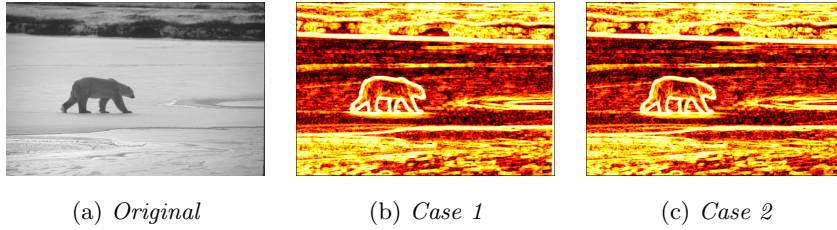


Fig. 1. Original image from BSDS [1] (100007) along with feature images obtained after applying Algorithm 1 with different ODM functions to original image.

4.2 Experimental framework

In order to analyse the behaviour of our proposal we follow the scheme given by Bezdek *et al.*[4], adding the final step to quantify the results of the method:

- (S1) Smooth the image applying a Gaussian filter (with $\sigma = 1$) to \mathbb{I}_g ;
- (S2) Obtain the feature image with Algorithm 1;

- (S3) Thin the feature image using non-maxima suppression [9];
- (S4) Binarize the thinned image using the hysteresis method [22].
- (S5) Compare the binary image with ground truth images [10].

As we want to evaluate the performance of ODM functions for edge detection, we carry out a fair comparison between different edge detection methods focusing on step (S2). Hence, the remaining steps are homogeneous for all contending methods.

For comparison purposes, we test our algorithm with the following well-known edge detection approaches: the Canny method [9] with $\sigma_C = 2.25$ for the derivative operator, as indicated in [19]; the Gravitational Edge Detection (GED) method [17], using the probabilistic sum (G_{SP}) and the maximum (G_{SM}); Fuzzy morphological edge detector [12] with two variants, using the Schweizer-Sklar [24] t-norm and t-conorm (FM_{SS}) and the minimum and maximum (FM_{MM}).

4.3 Feature image fusion using penalty functions

As a complementary experiment, we consider different feature images obtained in step (S2), using different edge detection methods, and we build a consensus feature image based on the concept of penalty function. In this way, the corresponding value of a pixel is the least dissimilar to all the proposed feature images.

For the experimentation we use the following specific expression for the penalty function:

$$P_{\nabla}(\mathbf{X}, \mathbf{y}) = \sum_{q=1}^m \sum_{p=1}^n |x_p^q - y_q|^2. \quad (2)$$

To obtain the consensus feature image we follow the next steps for each pixel of the images at position (x, y) :

- 1) Take 3 pixels from each feature image in the same position;
- 2) Apply a series of q aggregation functions, M_1, \dots, M_q ;
- 3) Compute the penalty function given by Eq. (2) over the previous result;
- 4) Get the values associated to the aggregation providing the smallest value of Eq. (2)

As an example of this procedure we propose a construction method, *Case 3*, where we aggregate a series of five feature images with five aggregation functions. Concretely we use the *minimum*, the *maximum*, the *arithmetic mean* and two *OWA operators*; the first one associated with the quantifier *the largest possible amount* and the second one with the quantifier *the largest part of*. With the aggregation functions we take the following feature images: *Case 1*, *Case 2*, Canny, FM_{SS} , G_{SP} .

4.4 Dataset and quantification of the results

For our experiments we have used the images of Berkeley Segmentation Dataset (BSDS500) [1], specifically 100 natural images from the test set. Associated to each original image there exist several hand-labelled segmentations denoted as *ground truth images*.

As the ground truth are binary images, we are dealing with a classification problem where each pixel is considered in a confusion matrix as in the Martin *et al.* approach [20], where *True Positive*, *False Positive*, *etc.* are considered. The matching process between the solution obtained and all the human ground-truths is done comparing pixel-to-pixel in both images. Because of possible spatial differences we need some tolerance in the comparison, as an edge pixel can be slightly displaced w.r.t. the ground-truth. Concretely we apply a tolerance of 2.5% of the image diagonal. This comparison is done using the approach presented by Estrada and Jepson in [10] and available at [14].

In order to interpret the previous values, we use the following Precision/Recall measures:

$$Prec = \frac{TP}{TP + FP}, \quad Rec = \frac{TP}{TP + FN}, \quad F_\alpha = \frac{Prec \cdot Rec}{\alpha \cdot Prec + (1 - \alpha) \cdot Rec}.$$

In this work we stick to the usual value of $\alpha = 0.5$ for the F -measure according to [20, 18], representing the harmonic mean between $Prec$ and Rec .

4.5 Experimental results

In Table 1 the results of each edge detection method are indicated displaying the average of $Prec$, Rec and $F_{0.5}$. On the one hand, in terms of Rec we can infer that we have obtained as good results as the Canny method with *Case 1* and *Case 2*, i.e., not including a lot of false positives. On the other hand, we may observe that FM_{SS} combines a medium precision with a very high recall, therefore the majority of edges are detected at the cost of including a high number of false positives. Considering the overall measure, i.e., $F_{0.5}$, the results achieved with *Case 1* and *Case 2* are competitive with the ones obtained with the Canny method and gravitational forces. In addition, if we observe *Case 3* we have substantially improved the results and even surpassed the best performer, namely, the Canny method.

As complementary comparative measure we consider the number of images being the best and worst performer in terms of $F_{0.5}$. In Table 2 we show the results of such a measure and we observe that with our proposed approaches we obtain a score of 0 in terms of worst result. Moreover, there exist a good number of best images obtained with *Case 1* and *Case 2* methods. The result is comparable to the one obtained with gravitational forces method taking into account that in some cases this method is the worst performer. Considering *Case 3* we can see that we slightly improve the result of best images obtained, taking them mainly from the Canny method and G_{S_P} methods.

Edge Detection Methods	<i>Prec</i>	<i>Rec</i>	$F_{0.5}$
<i>Case 1</i>	0.61	0.67	0.61
<i>Case 2</i>	0.60	0.69	0.62
<i>Case 3</i>	0.61	0.73	0.64
FM_{SS}	0.44	0.87	0.57
FM_{MM}	0.70	0.40	0.45
<i>Canny</i>	0.66	0.65	0.63
G_{SP}	0.59	0.71	0.62
G_{SM}	0.60	0.68	0.60

Table 1. Comparison of ODM functions *Case 1*, *Case 2* approach, along with penalty functions *Case 3* respect to Gravitational, Fuzzy Morphology and the Canny method in terms of *Prec*, *Rec* and $F_{0.5}$.

		Edge Detection Method											
*		FM_{SS}		FM_{MM}		<i>Canny</i>		G_{SP}		G_{SM}			
		✓	✗	✓	✗	✓	✗	✓	✗	✓	✗		
<i>Case 1</i>		18	0	15	18	2	76	39	1	18	2	8	3
<i>Case 2</i>		18	0	15	18	2	76	39	1	18	2	8	3
<i>Case 3</i>		29	0	14	18	2	76	35	1	12	2	8	3

Table 2. Comparison of best and worst approaches in terms of $F_{0.5}$ considering ODM functions *Case 1*, *Case 2* and penalty approaches with five feature images *Case 3*.

As a visual demonstration, we show in Fig. 2 the results obtained with all the approaches considered in our experiments. As we can easily observe, the binary edges given by each approach are different, some of them depict more edges than others, giving more information but taking less care of false positives; therefore one approach can provide a better solution for a specific problem than other edge detection method.

5 Conclusion

We have take a novel approach to edge detection using ODM functions. Given a neighbourhood we have measured the changes of intensity around each pixel taking into account the direction defined by the intensity variation vector, ordered in a decreasing way. Such functions are proven effective to determine the existence of an edge. Furthermore, using different ODM functions to obtain feature images and considering $F_{0.5}$ measure, we can conclude that the proposed methods are very competitive in comparison with the Canny method and gravitational forces, outperforming the Fuzzy Morphology approach. Moreover, if we use penalty functions to build a consensus feature image, we get the best value of the $F_{0.5}$ measure, overtaking the Canny method. However, regarding the number of images for which our methods get the best or the worst result, we see that, using penalty functions, the Canny method is the best method for more images, but our method provides the best result regarding worst count. Also, we would

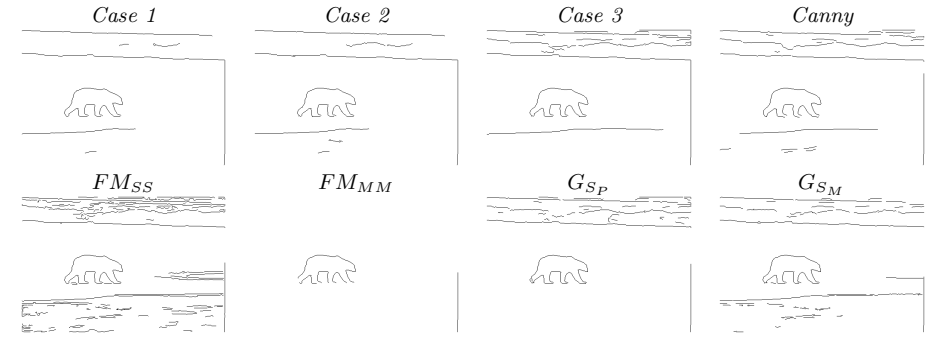


Fig. 2. Binary image obtained with ODM functions and penalty functions (*Case 1*, *Case 2*, *Case 3*), Canny, Fuzzy Morphology (FM_{SS} , FM_{MM}) and Gravitational forces (G_{SP} , G_{SM}).

like to stress the simplicity of our method, in particular when the feature image is built using the expression in Eq. (1).

Regarding future research lines, we consider studying more possible values for the proposed parameters and optimize them, in order to confirm this preliminary results, and to analyse the spatio-temporal complexity of the algorithm in relation with other edge detection algorithms.

Acknowledgments.

This work is supported by the Spanish Ministry of Science (Project TIN2016-77356-P) and the Research Services of Universidad Publica de Navarra.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence* 33(5), 898–916 (2011)
2. Beliakov, G., Pradera, A., Calvo, T.: *Aggregation Functions: A Guide for Practitioners*, vol. 18 (2007)
3. Beliakov, G., Sola, H.B., Sánchez, T.C.: *A practical guide to averaging functions*, vol. 329. Springer (2016)
4. Bezdek, J., Chandrasekhar, R., Attikouzel, Y.: A geometric approach to edge detection. *IEEE Trans. Fuzzy Syst.* 6(1), 52–75 (1998)
5. Bustince, H., Barrenechea, E., Sesma-Sara, M., Lafuente, J., Dimuro, G.P., Mesiar, R., Kolesarova, A.: Ordered directionally monotone functions. justification and application. *IEEE Transactions on Fuzzy Systems* PP(99), 1 (2017)
6. Bustince, H., Fernandez, J., Kolesárová, A., Mesiar, R.: Directional monotonicity of fusion functions. In: *Eur. J. Oper. Res.* vol. 244, pp. 300–308 (2015)
7. Bustince, H., Beliakov, G., Pereira Dimuro, G., Bedregal, B., Mesiar, R.: On the definition of penalty functions in data aggregation. *Fuzzy Sets Syst.* 323, 1–18 (2017)

8. Calvo, T., Kolesárová, A., Komorníková, M., Mesiar, R.: Aggregation Operators: Properties, Classes and Construction Methods. *Aggreg. Oper. New Trends Appl.* 97(1), 3–104 (2002)
9. Canny, J.F.: A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8(6), 679–698 (1986)
10. Estrada, F.J., Jepson, A.D.: Benchmarking image segmentation algorithms. *Int. J. Comput. Vis.* 85(2), 167–181 (2009)
11. Forero-Vargas, M.G.: Fuzzy thresholding and histogram analysis. In: *Fuzzy Filters for Image Processing*, pp. 129–152. Springer (2003)
12. Gonzalez-Hidalgo, M., Massanet, S., Mir, A., Ruiz-Aguilera, D.: On the choice of the pair conjunction-implication into the fuzzy morphological edge detector. *IEEE Trans. Fuzzy Syst.* 23(4), 872–884 (2015)
13. Grabisch, M., Marichal, J.L., Mesiar, R., Pap, E.: *Aggregation Functions (Encyclopedia of Mathematics and Its Applications)*. Cambridge University Press, New York, NY, USA, 1st edn. (2009)
14. Kermit Research Unit (Ghent University): The kermit image toolkit (kitt), www.kermitimagetoolkit.com, online
15. 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)
16. Lopez-Molina, C.: The breakdown structure of edge detection- Analysis of individual components and revisit of the overall structure. Ph.D. thesis, Universidad Publica de Navarra (2012)
17. 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)
18. Lopez-Molina, C., De Baets, B., Bustince, H.: Quantitative error measures for edge detection. *Pattern Recognition* 46(4), 1125–1139 (2013)
19. Lopez-Molina, C., De Baets, B., Bustince, H.: A framework for edge detection based on relief functions. *Inf. Sci. (Ny)*. 278, 127–140 (2014)
20. Martin, D.R., Fowlkes, C.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)
21. Martin, D.R.: An empirical approach to grouping and segmentation (2003)
22. Medina-Carnicer, R., Muñoz-Salinas, R., Yeguas-Bolivar, E., Diaz-Mas, L.: A novel method to look for the hysteresis thresholds for the Canny edge detector. *Pattern Recognit.* 44(6), 1201–1211 (2011)
23. Prewitt, J.: Object enhancement and extraction (1970)
24. Schweiser, B., Sklar, A.: Associative functions and statistical triangle inequalities. *Publicaciones Mathematicae, Debrecen* 8, 169–186 (1961)
25. Sobel, I., Feldman, G.: A 3x3 isotropic gradient operator for image processing. Hart, P. E. Duda R. O. *Pattern Classif. Scene Anal.* pp. 271–272 (1973)
26. Torre, V., Poggio, T.: On Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8(2), 147–163 (1986)
27. van de Weijer, J., van den Boomgaard, R.: Local mode filtering. In: *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, 8-14 December 2001, Kauai, HI, USA. pp. 428–433. IEEE Computer Society (2001)
28. Wilkin, T., Beliakov, G.: Weakly monotonic averaging functions. *Int. J. Intell. Syst.* 30(2), 144–169 (2015)
29. Yager, R.R.: On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Trans. Syst. Man Cybern.* 18(1), 183–190 (1988)