Efficient Approach for OS-CFAR 2D Technique Using Distributive Histograms and Breakdown Point Optimal Concept applied to Acoustic Images

Sebastián A. Villar[1], Bruno V. Menna[1], Sebastián Torcida[2] and Gerardo G. Acosta[1]

[1] INTELYMEC Group (UNCPBA) and CIFICEN (UNCPBA-CICPBA-CONICET), Av. del Valle 5737, B7400JWI Olavarría, Argentina.
[2] Departamento de Matemática, Facultad de Ciencias Exactas (Campus), UNCPBA, Tandil, Argentina.
*svillar@fio.unicen.edu.ar

Abstract: In this work, a new approach to improve the algorithmic efficiency of the Order Statistic-Constant False Alarm Rate (OS-CFAR) applied in two dimensions (2D) is presented. OS-CFAR is widely used in radar technology for detecting moving objects as well as in sonar technology for the relevant areas of segmentation and multi-target detection on the seafloor. OS-CFAR rank orders the samples obtained from a sliding window around a test cell to select a representative sample that is used to calculate an adaptive detection threshold maintaining a false alarm probability. Then, the test cell is evaluated to determine the presence or absence of a target based on the calculated threshold. The rank orders allows that OS-CFAR technique to be more robust in multi-target situations and less sensitive than other methods to the presence of the speckle noise, but requires higher computational effort. This is the bottleneck of the technique. Consequently, the contribution of this work is to improve the OS-CFAR 2D with the distributive histograms and the optimal breakdown point optimal concept, mainly from the standpoint of efficient computation. In this way, the OS-CFAR 2D on-line computation was improved, by means of speeding up the samples sorting problem through the improvement in the calculus of the statistics order. The theoretical algorithm analysis is presented to demonstrate the improvement of this approach. Also, this novel efficient OS-CFAR 2D was contrasted experimentally on acoustic images.

1. Introduction

The underwater target detection from acoustic images represents a typical process required in different automatic applications such as archaeology, resources search, inspection and maintenance of pipelines, mine or waste detection, and other types of monitoring [1, 2]. Many approaches are currently available in acoustic domain such as Multi-fractal Analysis [3], Markov Random Field (MRF) [4], Local Fourier Histograms [5], Active Contours (AC) [6], Gauss–Markov Random Field Model [7], Undecimated Discrete Wavelet Transform (UDWT) [8], among others. These approaches require computationally expensive mathematic models to underwater target detection. On the other hand, the Constant False Alarm Rate (CFAR) represents an adaptive technique able to perform accurate and robust target detection. This technique is commonly used in radar technology for detecting moving objects [9, 10] as well also in sonar technology applied in acoustic images from different sonar's device for multi-target detection [11–13], underwater pipeline detection on the seafloor [14, 15], acoustic segmentation of several types of regions [16], among others. CFAR calculate an adaptive detection threshold from interference power values to maintain an expected false alarm probability [9]. In the literature there are numerous CFAR techniques such as Cell Averaging CFAR (CA-CFAR), Order Statistic CFAR (OS-CFAR), Greatest Of CFAR (GO-CFAR), Smallest Of CFAR (SO-CFAR), Censored Mean-Level Detector CFAR (CMLD-CFAR), Trimmed Mean CFAR (TM-CFAR), and others variations [9, 17–19].

Focusing on the OS-CFAR, for each test cell this technique evaluates the presence or absence of a target sorting the samples from a sliding window to select a representative sample that is used to calculate an adaptive detection threshold maintaining a false alarm probability [9]. The representative sample could be selected by setting a fixed-order statistics threshold, as suggested in [20], or automatically estimated, based on the application of the Information Theoretic Criteria (ITC) principle which does not require any prior information about the number of interfering targets [21].

In radar OS-CFAR can maintain robust performance of clutter suppression and does not suffer large detection loss in non-stationary and non-uniform distribution clutter environment. Besides, OS-CFAR is suitable for being used in multi-target situations because of its high resolution [17–19, 22]. In sonar OS-CFAR demonstrated to be more robust in multi-target situations and less sensitive to the presence of the speckle noise [13–16]. The main drawback of OS-CFAR is its computational effort because sorting is a time-consuming task. This computational effort prevents its use in real-time applications [11], and therefore the utility of OS-CFAR technique decreases. In addition, a two-dimensional (2D) sliding window is necessary to consider more contextual information and hence to improve detections. In this case, the computational effort increases considerably.

The bottleneck of OS-CFAR is the sorting problem. Many authors in the literature have tried to approach this sorting problem. A rather thorough comparison is offered in [23, 24], where the efficiency of several methods is reviewed in the worst case for a one-dimensional array of N elements: *insertion*, with a $O(N^2)$; *selection*, which is also $O(N^2)$; *bubble sort*, $O(N^2)$; *bucket sort*, decreasing complexity to $O(N^2)$ when the distribution of elements is assumed constant, *quick sort* $O(N^2)$, *merge sort* $O(N \log N)$, *heap sort* $O(N \log N)$, just to mention the most common approaches [25]. On the other hand, in the digital image processing context numerous advances on the Median Filter (MF) computation have been made. The process of applying a MF is a nonlinear smoothing one, best known for reducing impulsive or salt-and-pepper noise from a digital image while respecting its edges [26]. Briefly, the MF sliding-window visits each image element and places its center on it. The intensity values within the window of radius size r are sorted, and the median intensity value is then used to replace the window's center in the filtered image. A rather thorough comparison of MF is offered in [25] using the classic sorting methods for a one-dimensional array. The main reference of MF for two-dimensional array is the *Huang et al.* method [27], which was the first exhibiting in the worst case a $O(r)$ per pixel algorithmic complexity (where r denotes the radius for a two-dimensional array) using a single histogram. Different approaches have since tried to break this linearity: the *Weiss* method [28] uses hierarchical histograms to reach a $O(log\,r)$ per pixel algorithmic complexity but losing simplicity, and the *Gil and Werman* method [29] has a $O(log^2\,r)$ per pixel algorithmic complexity and it is based on trees. The *Perreault and Herbert* method [30] represents a variation of *Huang et al.* algorithm using distributive histograms reaching in the worst case the lowest algorithmic complexity of $O(b)$ per pixel (where $b = 2^{image\ bit-depth}$, the number of scale levels). Finally, the *Villar et al.* method [31] demonstrated new improvements and capabilities of MF using the optimal breakdown point concept maintaining the same algorithmic complexity $O(r)$ and $O(b)$ for *Huang* and *Perrault and Herbert* versions, respectively, but reducing objective metrics computational (dynamic memory accesses, arithmetic operations, logic comparison and transition effort) that clearly outperform the corresponding standard versions.

In this work, a new approach to improve the algorithmic efficiency of the OS-CFAR 2D using the distributed histograms and the breakdown point optimal concept is presented. This efficiency improvement is demonstrated performing a theoretical algorithm analysis, as well as supported with experimental evidence on real images. This analysis allows to claim the highest algorithmic efficiency for the OS-CFAR 2D technique to date.

This article is organized as follows: Section 2 discusses the basic concepts about target detection using OS-CFAR 2D applied on acoustic image. Section 3 describes the efficient OS-CFAR 2D approach proposed using the distributive histograms and breakdown point concept. Section 4 depicts an evaluation of the theoretical efficiency analysis of the proposed approach. Section 5 shows the experimental results on real images. The article ends with some final comments in Section 6.

2. Target detection using OS-CFAR 2D

The target detection problem in the acoustic image consists of analyzing each echo signal with the purpose of to detect the presence or absence of a target. Detection is usually done through the contextual information analysis of each echo signal. In [12] two hypotheses were defined for this analysis: (i) *the echo signal is the background* (H_0), and (ii) *the echo signal is a combination of background and echoes of a target* (H_1). If the detection system decides that H_0 is validated (target is not present), then hypothesis H_0 is stated. Otherwise, if the detection system decides that H_1 is validated, then hypothesis H_1 is stated, meaning that a target is present.Then k-*th* order statistic value x_k is selected as representative of the echo signal and a detection threshold $\hat{T}$ is estimated applying a scale factor α_{OS}:

$$\hat{T} = \alpha_{OS}\, x_k \ (1)$$

This scale factor α_{OS} is a constant value determined from false alarm probability P_{fa}. As OS-CFAR keeps on a constant false alarm probability, this detection threshold only varies depending on x_k value. Therefore, this technique considers the contextual information of each cell under test $x_{i,j}$ to determine the adaptive detection threshold. Then, for each test cell, the detector system makes a decision according to the following decision strategy: For further details about this, please refer to [12, 15, 16].

$$x_{i,j} \underset{H_0}{\overset{H_1}{\gtrless}} \hat{T} \ (2)$$

3. Proposal of an efficient OS-CFAR 2D approach

The bottleneck of the OS-CFAR 2D technique is found in the sorting and selection stage of the k-*th* order statistic value. To solve the sorting problem, in the literature there are numerous approaches for one or two dimensional array that are classified based on simplicity, algorithmic complexity and objective computational metrics [23, 24, 27–30]. Among all these methods, [27] and [30] are of special interest due to they claim the lowest algorithmic complexity to date. On the other hand, the selection of k-*th* order statistic value based on [27] and [30] methods can be improved significantly using the optimal breakdown point [31]. These methods are really useful to enhance the algorithmic efficiency of the OS-CFAR 2D technique.

Fig. 1 shows the proposed architecture of the efficient OS-CFAR 2D approach using distributive histograms and optimal breakdown point concept. The input image x represents a two-dimensional array of size AxB (rows by columns), where $x_{i,j}$ denotes the pixel value at the intersection of the image i-*th* row and the j-*th* column. The pixel value is an integer positive quantity b (with $b = 2^{image\ bit-depth}$). The sliding window that shifts through the whole image X have a square radius of size r centered at $x_{i,j}$. The $(2r + 1)^2$ pixel values (with $i - r \le i \le i + r$ and $j - r \le j \le j + r$) are sorted using the distributive columns histograms h and the kernel histogram H described in Section 3.1. Each column histogram h^j and the kernel histogram H represent a one-dimensional array of size b. These histograms store the frequencies of the pixel values f of the current sliding window. The k-*th* order statistic value x_k is selected using the optimal breakdown point described in Section 3.2, to estimate the threshold $\hat{T}$ applying a scale factor α_{OS}. Then, the value of the test cell $x_{i,j}$ is compared with estimated threshold $\hat{T}$ to determine if the target is present. Finally, in Section 3.3 the complete procedure for a particular pixel of an image based on a flowchart is detailed.

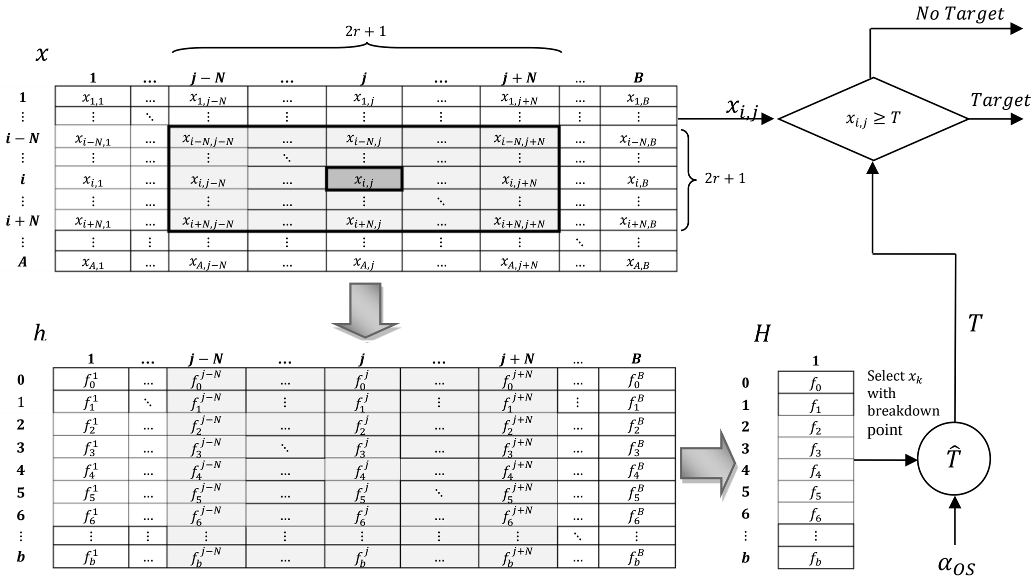


Fig. 1 *OS-CFAR 2D approach using distributive histograms and optimal breakdown point concept.*

3.1. Distributive histograms

To understand the sorting method proposed for OS-CFAR 2D, first consider the alternative of using a single kernel histogram H to store and update all the values from the current sliding window [27]. The kernel histogram H express a one-dimensional array of size b that store the frequencies f of the pixel values of the current sliding window (where $f_0, \ldots, f_b$ represents the amount of $0, .., b$ pixel values). The computing of k-*th* order statistic value x_k is done by accumulating frequencies f of kernel histogram H from one extreme of the scale and stopping when the cumulative sum reaches the boundary $k\,N_c$. The kernel histogram H is updated with new values as the sliding window scrolls through the image. Fig. 2 shows an example of a two-dimensional array x representing an input image of size AxB (rows by columns) with a squared window of radius $= 2$. When the window's center shifts one pixel to the right, e.g., from $x_{3,3}$ to $x_{3,4}$, the update of H requires those values from $x_{1,1}, \ldots, x_{5,1}$ to be removed and those values from $x_{1,6}, \ldots, x_{5,6}$ to be added in H. Therefore, $2r + 1$ removals and $2r + 1$ additions need to be carried out. Notice that this method maintains the frequency of cell values $f_0, \ldots, f_b$ between columns as sliding window shifts to its right, but is not retained between rows, i.e., when the sliding window is moved downwards. An alternative to improve this method is to maintain one histogram for each column in the image [30] and to use the additive property of histograms [28]. This property establishes that the union of two sets histograms R_1 and R_2 is simply the addition of their respective histograms:

$$H\,(R_1 \cup R_2\,) = \,H\,(R_1) + \,H\,(R_2) \quad (3)$$

In this way, to maintain the frequencies of cell values $f_0, \ldots, f_b$ between rows and columns is required one histogram h^j $(1 \leq j \leq B)$ for each image column (see Fig. 1). The column histograms h^j and the kernel histogram H are preserved and updated during all sliding window scrolling. Fig. 3 shows an example of a two-dimensional array x representing an input image of size AxB (rows by columns) with a squared window of radius $= 2$.

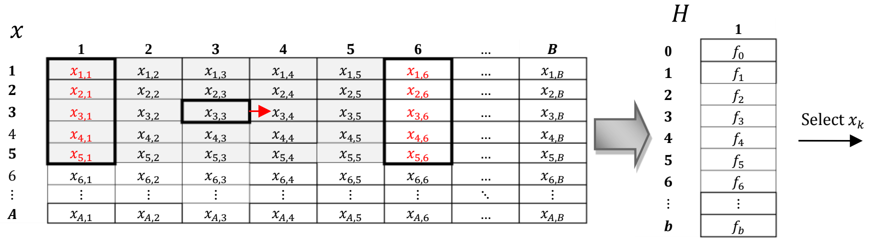
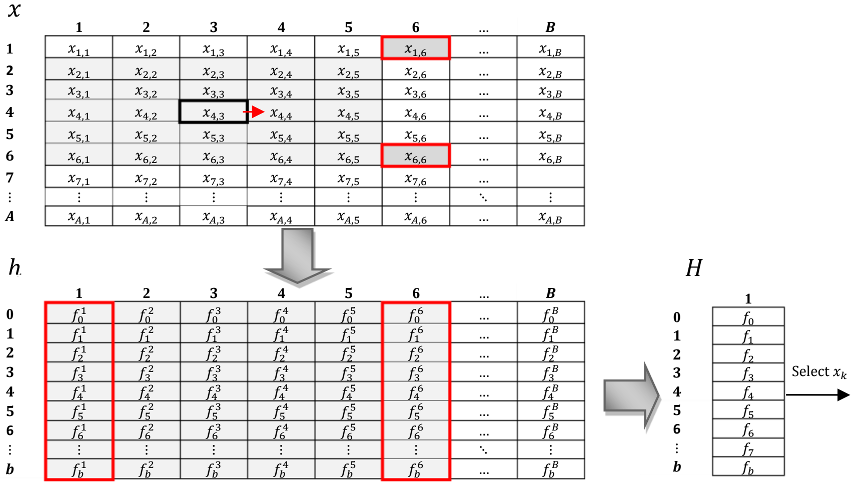


Fig 2 *Each time the window's center shifts to its right, the kernel histogram H to update requires* $2r + 1$ *additions to and* $2r + 1$ *removals* [31].



	1	2	3	4	5	6	…	B
1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	…	$x_{1,B}$
2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	…	$x_{2,B}$
3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$	…	$x_{3,B}$
4	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$x_{4,6}$	…	$x_{4,B}$
5	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$	$x_{5,5}$	$x_{5,6}$	…	$x_{5,B}$
6	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	$x_{6,5}$	$x_{6,6}$	…	$x_{6,B}$
7	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$	$x_{7,5}$	$x_{7,6}$	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
A	$x_{A,1}$	$x_{A,2}$	$x_{A,3}$	$x_{A,4}$	$x_{A,5}$	$x_{A,6}$	…	$x_{A,B}$

	1	2	3	4	5	6	…	B
0	f_0^1	f_0^2	f_0^3	f_0^4	f_0^5	f_0^6	…	f_0^B
1	f_1^1	f_1^2	f_1^3	f_1^4	f_1^5	f_1^6	…	f_1^B
2	f_2^1	f_2^2	f_2^3	f_2^4	f_2^5	f_2^6	…	f_2^B
3	f_3^1	f_3^2	f_3^3	f_3^4	f_3^5	f_3^6	…	f_3^B
4	f_4^1	f_4^2	f_4^3	f_4^4	f_4^5	f_4^6	…	f_4^B
5	f_5^1	f_5^2	f_5^3	f_5^4	f_5^5	f_5^6	…	f_5^B
6	f_6^1	f_6^2	f_6^3	f_6^4	f_6^5	f_6^6	…	f_6^B
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
b	f_b^1	f_b^2	f_b^3	f_b^4	f_b^5	f_b^6	…	f_b^B

	1
0	f_0
1	f_1
2	f_2
3	f_3
4	f_4
5	f_5
6	f_6
⋮	f_7
b	f_b

Fig 3 *Two steps of the proposed algorithm when the sliding windows shifts one pixel to the right: (a) The column histogram h^6 is updated by adding one pixel value $x_{6,6}$ and subtracting another pixel value $x_{1,6}$. (b) The kernel histogram H is updated by adding the modified column histogram h^6 and subtracting the leftmost one h^1.*

When the window's center shifts one pixel to the right, e.g., from $x_{4,3}$ to $x_{4,4}$, two updates are performed: (a) the rightmost column histogram h^6 is updated by removing and adding an top $x_{1,6}$ and an bottom $x_{6,6}$ pixel value, respectively; (b) the kernel histogram H is updated removing the leftmost column histogram h^1 and adding the rightmost column histogram h^6. Note that, in the update step (a), the pixel values of another rows are retained into the column histograms h^j. In the same way, the computing of k-*th* order statistic value x_k is done by accumulating frequencies f of kernel histogram H from one extreme of the scale and stopping when the cumulative sum reaches the boundary $k\,N_c$. The addition and subtraction of columns histograms only depends on the number of histogram bins b, itself a function of the image bit-depth.

3.2. Breakdown point concept

Since OS-CFAR 2D uses a sorting method for selection of k-*th* order statistic value x_k, this alternative is hence highly more reliable than the classical mean or average as utilized by other CFAR methods [9, 17–19]. In fact, the mean of a set of numbers changes when at least one of those numbers is changed or replaced. However, when using a sorting method, the result of selecting an order statistical value x_k does not vary substantially when those numbers are changed or replaced, always depending on the statistical order k established. E.g., for $k = 0.5$ assume that n (odd) ordered numbers are at hand; if the $\frac{(n+1)}{2} - 1$ lowest numbers are replaced by other arbitrary numbers but keeping them below the $\frac{(n+1)}{2}$ ranked number (the median), the new median will remain the same [31]. In general terms, the breakdown point concept denotes the percentage of data in a set that could be arbitrarily replaced without grossly modifying the value of an estimation or a computation [32]. Clearly, the mean has a 0% breakdown point, the median has almost a 50% breakdown point, and the other k-th order statistic has almost a $\min(k; 1-k)$ % breakdown point.

Each time the sliding window shifts some of its values are removed and simultaneously replaced by new values, and a new k-*th* order statistic is computed. The k-*th* order statistic value computation typically uses a bottom-up or top-down accumulating strategy (e.g., for an 8-bit grayscale where values range from 0 to 255 or 255 to 0 the frequencies are accumulated) until the cumulative sum reaches the boundary $k\,N_c$). Thereby, if the sliding window eventually processes an image region with most values near the top of the gray scale, the algorithm will get slower (an analogous problem would take place if frequencies were accumulated top-down and image regions with most of low pixel values were eventually found). Besides, it seems rather inefficient not to take into account that successive windows share most of their values and thus resulting in similar k-*th* order statistic. More precisely: the proportion of shared values between consecutive sliding windows of radius r is essentially $\frac{(2r+1)^2-2\,(2r+1)}{(2r+1)^2}$. The percentage of shared information between successive windows thus increases really fast with the radius: a 33% of shared values for a radius $r = 1$; a 60% of shared values for a radius $r = 2$, a 90% of shared values for a radius $r = 10$ and so on [31].

These inefficiencies can be overcome by making the most of the optimal breakdown point. In this way, the k-*th* order statistic value from a new sliding window can be computed significantly faster by retaining the k-*th* order statistic value from the previously processed window and updating it; in turn, this strategy enables a more efficient processing of those image regions with values in any extreme of the scale handling equally both cases. For this it is necessary to consider the previous k-*th* order statistic value Px_k, lower values Lx_k than Px_k and greater or equal values Gx_k than Px_k. When the sliding window scrolls through the image, each new value that is added to the kernel histogram H is compared with the Px_k: if the value is greater or equal than Px_k, Gx_k is incremented in one and otherwise Lx_k is incremented in one. Also, each old value that is removed of the kernel histogram H is compared with the Px_k: if the value is greater or equal than Px_k, Gx_k is decremented in one and otherwise Lx_k is decremented in one. To calculate the new k-*th* order statistic value, first it is needed to compute the threshold T_k based on k-*th* order statistic using $T = \lceil k[(2r + 1)^2 + 1]\rceil$; next, the threshold T_k is compared with Lx_k: if $T_k \leq Lx_k$, the new k-*th* order statistic value will be lower than the previous one and it will be found moving downward from the current Px_k bin in the kernel histogram H; otherwise, the new k-*th* order statistic value will be greater than the previous one and will be found moving upward from the current Px_k bin in H. The auxiliary variables Lx_k and Gx_k are accordingly updated in the process.

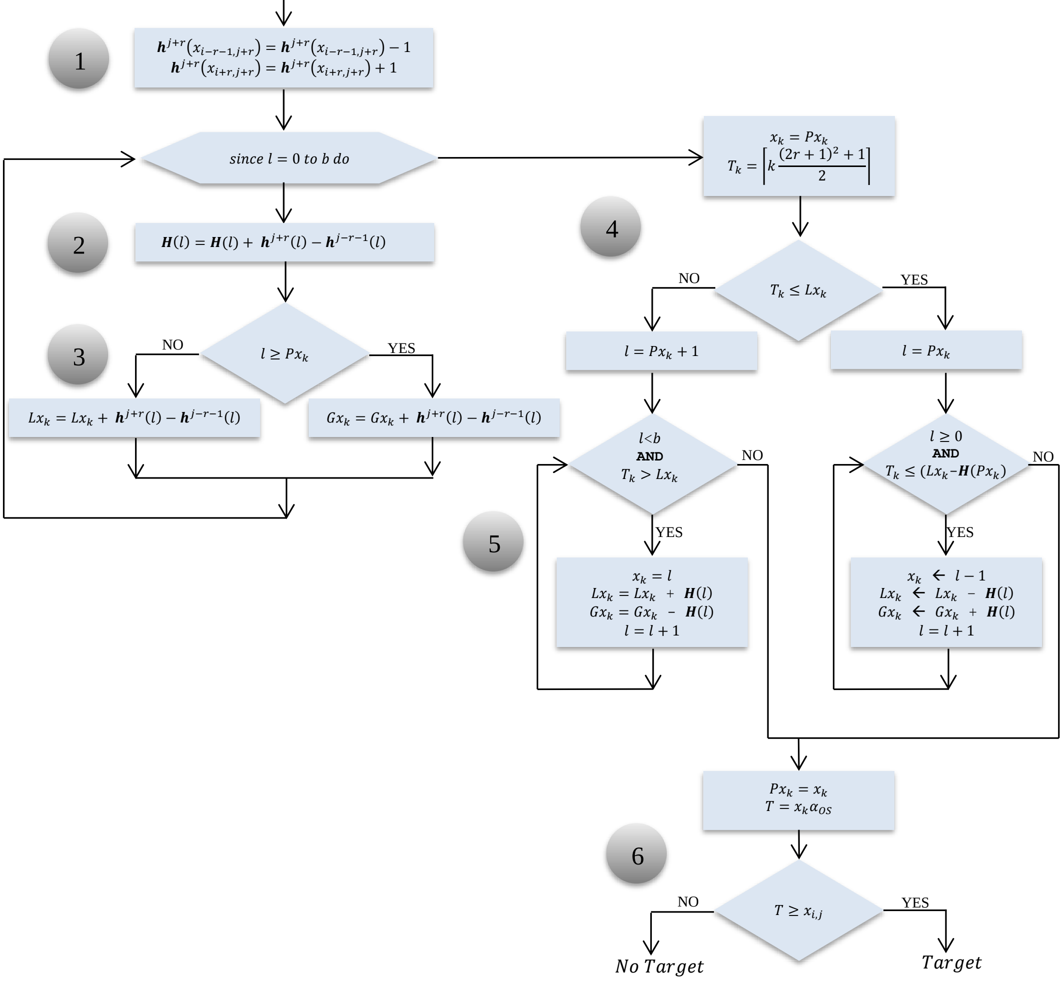

1
$\boldsymbol{h}^{j+r}(x_{i-r-1,j+r}) = \boldsymbol{h}^{j+r}(x_{i-r-1,j+r}) - 1$
$\boldsymbol{h}^{j+r}(x_{i+r,j+r}) = \boldsymbol{h}^{j+r}(x_{i+r,j+r}) + 1$
$since\ l = 0\ to\ b\ do$
2
$\boldsymbol{H}(l) = \boldsymbol{H}(l) + \boldsymbol{h}^{j+r}(l) - \boldsymbol{h}^{j-r-1}(l)$
3
$l \geq Px_k$
NO
YES
$Lx_k = Lx_k + \boldsymbol{h}^{j+r}(l) - \boldsymbol{h}^{j-r-1}(l)$
$Gx_k = Gx_k + \boldsymbol{h}^{j+r}(l) - \boldsymbol{h}^{j-r-1}(l)$
$x_k = Px_k$
$T_k = \left\lceil k\frac{(2r+1)^2+1}{2} \right\rceil$
4
$T_k \leq Lx_k$
NO
YES
$l = Px_k + 1$
$l = Px_k$
5
$l<b$ AND $T_k > Lx_k$
NO
YES
$x_k = l$
$Lx_k = Lx_k + \boldsymbol{H}(l)$
$Gx_k = Gx_k - \boldsymbol{H}(l)$
$l = l + 1$
$l \geq 0$ AND $T_k \leq (Lx_k - \boldsymbol{H}(Px_k))$
NO
YES
$x_k \leftarrow l - 1$
$Lx_k \leftarrow Lx_k - \boldsymbol{H}(l)$
$Gx_k \leftarrow Gx_k + \boldsymbol{H}(l)$
$l = l + 1$
$Px_k = x_k$
$T = x_k\alpha_{OS}$
6
$T \geq x_{i,j}$
NO
YES
$No\ Target$
$Target$

Fig. 4 *Flowchart of the OS-CFAR 2D approach using distributive histograms and breakdown point optimal concept.*

To further understand this method consider an example: a sliding window W of radius $r = 1$ with $(2r + 1)^2 = 9$ pixel values is given; then $T_k = \lceil k[(2r + 1)^2 + 1]\rceil = 7$ with $k = 0.75$. Assume that $W = [1, 3, 3, 4, 5, 5, 7, 7, 9]$ are current window's ordered values, so $Px_k = 7$, $Lx_k = 6$ and $Gx_k = 3$. Assume next that new values 1, 1 and 1 are added to W while old values 5, 5 and 1 are removed from it. This update of W results in $W = [1, 1, 1, 3, 3, 4, 7, 7, 9]$ which in turn updates $Lx_k = 6$ and $Gx_k = 3$ following the comparison with $Px_k = 7$. Since $T_k = Lx_k$ the k-*th* order statistic value and no bin-displacement through H is required. On the other hand, consider that new values 10, 9 and 8 are added to W while old values 1, 1 and 1 are removed from it. This update of W results in $W = [3, 3, 4, 7, 7, 8, 9, 9, 10]$ which in turn updates $Lx_k = 3$ and $Gx_k = 6$ following the comparison with $Px_k = 7$. Since $T_k = 7 > 3 = Lx_k$, the new k-*th* order statistic value will be found moving forward from the $Px_k = 7$ bin in the kernel histogram H. Here, only three bin-displacements through H are needed to obtain the new k-*th* order statistic value; the auxiliary variables are accordingly updated to $Px_k = 9$, $Lx_k = 6$ and $Gx_k = 3$, respectively. Since T_k is now smaller than Lx_k, the displacement will next start from Px_k and backward.

3.3. Complete procedures of efficient OS-CFAR 2D approach

Fig. 4 shows the flowchart for the complete procedure applied to each pixel value $x_{i,j}$ (i-*th* row and the j-*th* column) of an image X of size AxB using the OS-CFAR 2D presented. Note that previously it must be created and initialized the kernel histogram H, column histograms h^j $(1 \le j \le B)$ and the previous k-*th* order statistic value Px_k, lower values Lx_k than Px_k and greater or equal values Gx_k than Px_k. Essentially, the flowchart of the Fig. 4 consists of six steps for each pixel value $x_{i,j}$:

3.3.1. Updating the column histogram h^j. The rightmost column histogram h^{j+r} (where r is the radius size of the current sliding window) when the sliding window shifts to the right. A top $x_{i-r-1,j+r}$ and a bottom $x_{i+r,j+r}$ pixel values are respectively removed from and added to the rightmost column histogram h^{j+r}.

3.3.2. Updating the kernel histogram H. For all the image bit-depth (since $l = 0$ to b), those values from the sliding window leftmost column histogram h^{j-r-1} are removed from while those from the new rightmost column histogram h^{j+r} are added to the kernel histogram H.

3.3.3. Updating the variables Lx_k and Gx_k. For all the image bit-depth (since $l = 0$ to b) is compared with the k-*th* order statistic value Px_k: if the new value is greater or equal than Px_k, Gx_k is updated removing the values of the leftmost column histogram h^{j-r-1} and adding the values of rightmost column histogram h^{j+r} and otherwise Lx_k it is updated in a similar way.

3.3.4. Computing the k-th order statistic value x_k. This step computes the new k-*th* order statistic value by using the kernel histogram H and the variables Lx_k, Gx_k and Px_k. First, the previous k-*th* order statistic value Px_k is stored in the variable x_k; next, the threshold T_k is computed ($T_k = \left\lceil k\frac{(2r+1)^2+1}{2}\right\rceil$) and compared with Lx_k: if $T_k \le Lx_k$, the new k-*th* order statistic value will be lower than the previous one and it will be found moving downwards from the current Px_k bin in the kernel histogram H; otherwise, the new k-*th* order statistic value will be greater than the previous one and will be found moving upwards from the current Px_k bin in H. The auxiliary variables Lx_k and Gx_k are accordingly updated in the process.

3.3.5. Target detection. The new k-*th* order statistic value x_k is stored in the variable Px_k for the next calculation of the k-*th* order statistic value and the threshold T is estimated applying a scale factor α_{OS}. Then, the value of the test cell $x_{i,j}$ is compared with estimated threshold $\hat{T}$ to determine if the target is present.

4. Theoretical efficiency analysis

Table 1 exhibits computational metrics for the efficiency evaluation of the present OS-CFAR 2D approach. For this, a set of metrics based on computational time and space has been estimated: algorithmic complexity, dynamic memory, static memory, dynamic memory accesses, arithmetic operation and logic comparisons [23]. Metrics algorithmic complexity, dynamic memory and static memory are globally quantified while dynamic memory accesses, arithmetic operation and logic comparisons are quantified per-pixel. Each operation as update column histogram, update kernel histogram, update Lx_k and Gx_k, computing the k-*th* order statistic value and target detection are evaluated in the worst, average and best case (see Fig. 4).

Table 1 Computational metrics of the OS-CFAR 2D approach.

Algorithmic complexity	Dynamic memory	Static memory	Operation	Type case	Dynamic memory accesses	Arithmetic operations	Logic comparisons
$O(b)$	X of size AxB H of size $bx1$ h^j $(1 \le j \le B)$ of size $bx1$	$i, j, r, l, b, k,$ x_k, T_k, T, α_{OS} $Px_k, Lx_k,$ Gx_k	Update Column Histogram	Worst Case Average Case Best Case	8	16	0
			Update kernel Histogram	Worst Case Average Case Best Case	$4b$	$6b$	b
			Update Lx_k and Gx_k	Worst Case Average Case Best Case	$2b$	$5b$	b
			Computing the k-th order statistic value	Worst Case Average Case Best Case	$3b$ $\frac{3b}{2}$ 1	$5b+5$ $\frac{5b+5}{2}$ 6	$2b+1$ $b+1$ 3
			Target detection	Worst Case Average Case Best Case	0	1	1

As shown in Table 1, the algorithmic complexity of the OS-CFAR 2D approach is $O(b)$ per pixel where $b = 2^{image\ bit-depth}$. To compute the dynamic memory it is needed the amount of memory to allocate the input image of size AxB, the kernel histogram H of size $bx1$ and the B column histograms h^j ($1 \leq j \leq B$) of size $bx1$. Besides, it is needed to allocate static memory for integer ($i, j, r, l, b, x_k, Px_k, Lx_k$ and Gx_k) and real (k, T_k, T and α_{OS}) auxiliary variables. Focusing in the five operations (see Fig. 4):

1. Updating the column histogram h^j needs 8 dynamic memory accesses (4 accesses for the input image x and column histogram h^j, respectively) and 16 arithmetic operations between addition and subtraction (worst, average and best case).
2. Updating the kernel histogram H needs $4b$ dynamic memory accesses ($2b$ accesses for the kernel histogram h^j and column histogram H, respectively), $6b$ arithmetic operations for addition and subtraction and b logic comparisons for loop since 0 to b (worst, average and best case).
3. Updating the variables Lx_k and Gx_k needs $2b$ dynamic memory accesses to column histogram h^j, $5b$ arithmetic operations between addition and subtraction and b logic comparisons for the conditional $l \geq Px_k$ (worst, average and best case).
4. Computing the k-*th* order statistic value x_k. First, the initialization of the threshold T_k requires 6 arithmetic operations and 1 logic comparison for the conditional $T_k \leq Lx_k$. Second, there are needed $3b$ dynamic memory accesses for histogram H, $5b$ arithmetic operations between addition and subtraction and $2b$ logic comparisons for the loop. Therefore, the worst case needs $3b$ dynamic memory accesses, $5b + 5$ arithmetic operations $2b + 1$ logic comparisons. The average and best case require $\frac{3b}{2}$ and 1 dynamic memory accesses, $\frac{5b+5}{2}$ and 6 arithmetic operations, $b + 1$ and 3 logic comparisons, respectively. Take into account that in the experimental results we will show that using the breakdown point optimal concept leads to the likelihood that the best case will be produced for the calculation of the k-*th* order statistic value.
5. Target detection needs 1 arithmetic operation to multiply x_k by α_{OS} and 1 logic comparison for the conditional $T \geq x_{i,j}$ (worst, average and best case).

5. Experimental results

The proposed approach for OS-CFAR 2D technique using the distributive histograms and breakdown point optimal concept was developed in C++ code taking advantage of the data structure within OpenCV 2.3 [33]. The programming environment (IDE) was Nokia QtCreator for GNU/Linux implementation C++ code. The framework was executed on a PC with a 2.4 GHz Intel Core i7-3630 CPU and 8-GB RAM memory with Ubuntu 14.04 LTS (32 bits) operating system.

The upper part of Fig. 5 shows an acoustic image under test. This image was obtained with an Edgetech® company side scan sonar device, on-board an autonomous underwater vehicle. It shows an underwater pipeline deployed on seafloor. The autonomous underwater vehicle was sailing at an altitude of $5m$ above the seafloor with a speed of $2m/s$. This acoustic image was cropped for better presentation and acquired from the right channel (starboard) of the side scan sonar device (note the shadow in the right part of underwater pipeline). The size of test images is 2000×900 ($A \times B$) with 256 levels of the grayscale. In addition, useful inspection features can be observed: free span, rock dump, and reflective objects on the seafloor. The free span feature can be clearly observed where the shadow close to the pipeline is not completely defined. In other words, a seafloor acoustic reverberation area is between pipeline and shadow. From this feature, it can be inferred that the pipeline is not deployed on the seafloor and could collapse causing economic and environmental disaster. The most common method to solve this problem is known as rock dumping where the pipeline is wider. Usually, rock dumping has to be examined to ensure stability.

This the test image was processed for different radius size r (1, 3 and 5) and order statistics k (0.5, 0.6, 0.75 and 0.8) and the resulting image of k-th order statistic value x_k are shown in Fig. 5. As it can be seen, when the radius r of the sliding window grows, more pixel values are used to compute the k-*th* order statistic value x_k. For example, for $r = 1$, 3 and 5 there are required 9, 49 and 121 pixel values, respectively. This increment in the size of the sliding window produces a fuzzy result because it uses a greater amount of pixel values to calculate the same k-*th* order statistic value x_k. Note that, when the smaller the radius size is selected, the detection results will include more geometrical details but also will be potentially more affected by noise. Conversely, the greater the radius size is, the higher the noise reduction will be, but the lesser the geometrical details retained.

On the other hand, when the statistical order k-*th* increases ($k = 0.5, 0.6, 0.75$ and 0.8) the resulting images become brighter due to the selected statistical order value x_k. In addition, a greater separation between shadow and highlight zones of the acoustic image is produced.

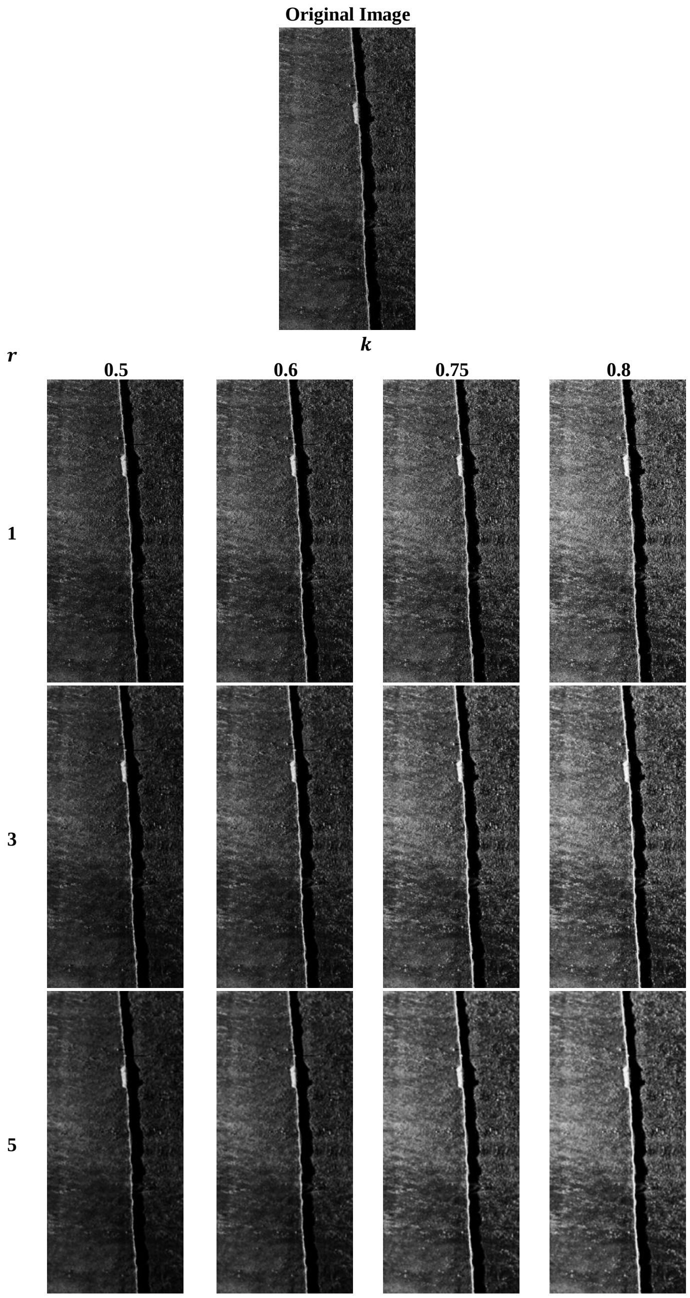


Fig. 5 *Visualization of the k-th order statistic value x_k for different radius size r (1, 3 and 5) and order statistics k (0.5, 0.6, 0.75 and 0.8) for sonar image with an pipeline on seafloor.*

In Fig. 6 it is shown a comparison of the standard and the breakdown point approaches to compute, for each order statistic k (0.5, 0.6, 0.75 and 0.8), the order statistic value x_k using the column-average bin-displacements measurement (vertical axis) versus the column number (horizontal axis). For each order statistic k, the blue, green and red curves describe the effort when radius sizes $r = 1$, 3 and 5 are used, respectively.

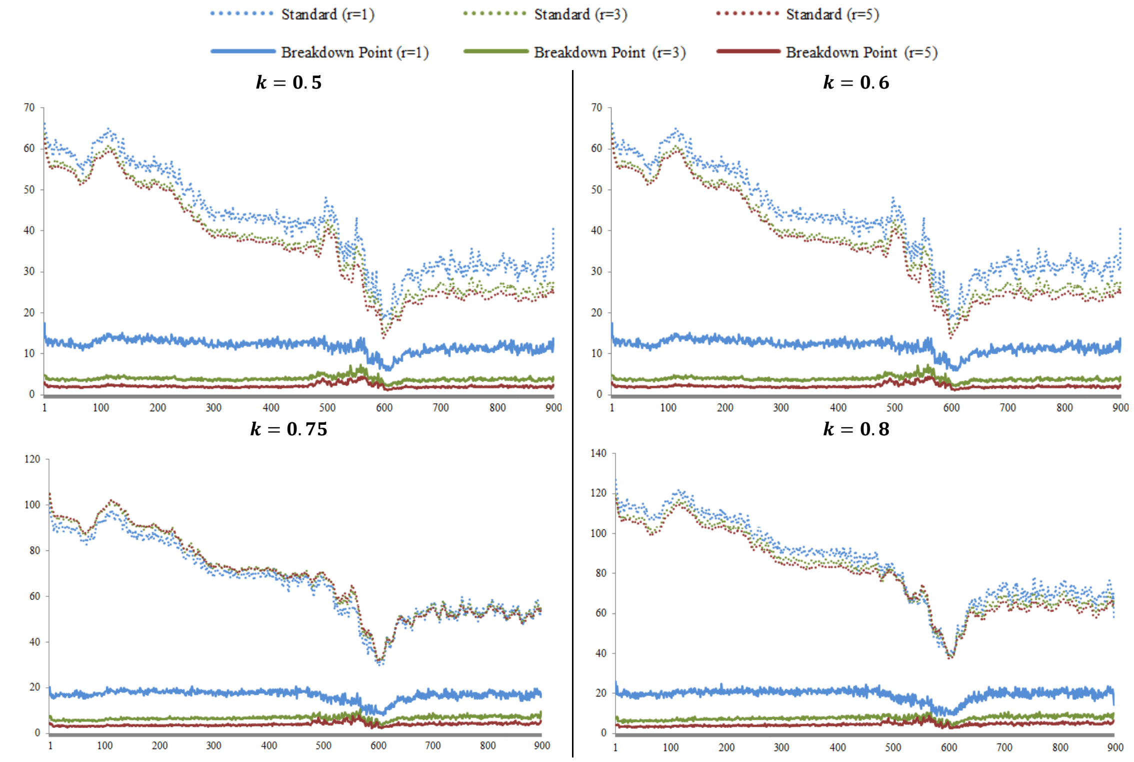


Fig. 6 Column-average bin-displacements for calculation of the k-th order statistic value x_k when different r radius size (1, 3 and 5) and order statistics k (0.5, 0.6, 0.75 and 0.8) are used.

In addition, the Table 2 shows the minimum and maximum column-average bin-displacements from Fig. 6. Note that, when the radius increases the curves corresponding to the breakdown point version stabilize in comparison to the standard version. This is caused by the greater number of processed values that allows calculating the order statistic value x_k and therefore the effort to achieve it tends to be zero.

Fig. 7 shows the images detection results of the proposed OS-CFAR 2D method for different radius sizes r (1, 3 and 5) and statistical order k (0.5, 0.6, 0.75 y 0.8) for a constant $P_{fa} = 0.013$. The P_{fa} was selected as lower as possible to avoid false alarms. Besides, the results represent a segmentation of three classes: highlight (white color), shadow (black color), and seafloor reverberation areas (gray color) employing the detection of multiclass strategy proposed in [16]. Note that, the intermediate step of the selection of the k-*th* order statistic value x_k was shown in Fig. 5. As it also can be seen in Fig. 7, when the radius r of the sliding window increases, greater number of pixel values are used to calculate the k-*th* order statistic value x_k, and therefore the number of false detections decreases.

Table 2 Minimum, maximum and mean column-average bin-displacements comparison by radius size r and order statistics k-*th* order statistic value x_k for standard and breakdown point versions of the Fig. 6.

r	x_k	**Breakdown Point**			**Standard**		
		Min	**Max**	**Mean**	**Min**	**Max**	**Mean**
1	0.5	5,92	17,42	11,96	18,28	66,16	42,10
	0.6	6,67	19,80	14,13	23,37	81,74	52,58
	0.75	7,92	20,49	16,77	29,63	100,69	66,66
	0.8	8,73	26,03	19,77	38,23	127,00	85,60
3	0.5	2,17	7,27	3,94	15,25	63,70	37,62
	0.6	2,74	8,53	4,93	20,38	77,91	48,10
	0.75	3,59	10,76	6,74	32,03	104,98	68,93
	0.8	3,93	10,70	7,68	38,79	119,03	81,65
5	0.5	1,10	4,36	2,15	13,87	62,51	36,24
	0.6	1,48	5,66	2,72	18,98	76,48	46,49
	0.75	2,34	7,91	3,95	31,16	104,90	68,96
	0.8	2,56	9,02	4,47	37,43	117,58	79,76

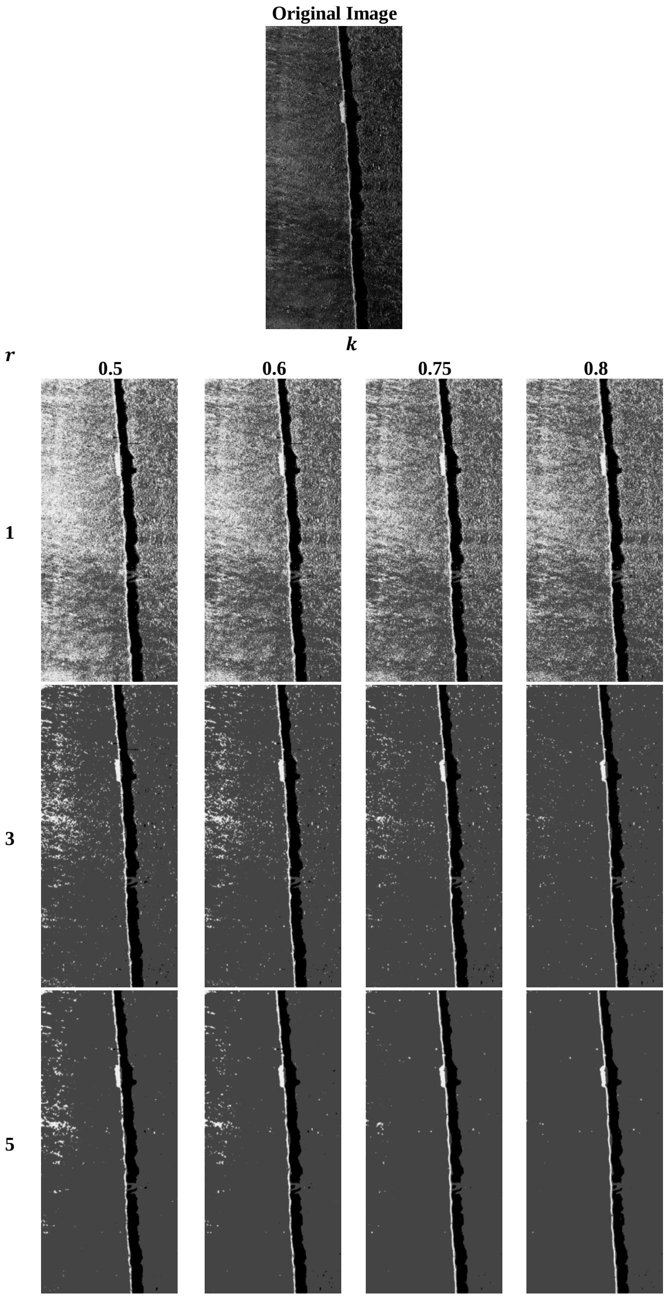


Fig. 7 *Sonar image for pipeline detections on the seafloor for different radius size r (1, 3 and 5) and order statistics k (0.5, 0.6, 0.75 and 0.8).*

On the other hand, when the k-*th* statistical order is increased, the amount of false detection is also decreased. The combination of in the settings parameters (r, k and P_{fa}) depends on the image resolution. Note that the selection of the parameters: radius size $r = 5$ (121 pixel values), statistical order $k = 0.8$ and $P_{fa} = 0.013$ allow to demonstrate a good trade-off with the detection results obtained. In addition, these parameters configuration allows to differentiate clearly the inspection features: free span, rock dump, and reflective objects on the seafloor.

In Fig. 8, three different regions of interest (ROI) extracted from the test image (Fig. 5) are shown; their sizes are $200\ x\ 200$ ($A\ \times B$) pixel values in every case, and the Cartesian coordinates of the corresponding upper-left vertex used as reference are, respectively: (50, 50), (460, 460); and (500, 1000). These three sampled regions were deliberately chosen to capture different inspection features descripted: (ROI-1) seabed reverberation; (ROI-2) underwater pipeline with rock dumping and (ROI-1) underwater pipeline deployed on the seafloor with free span.

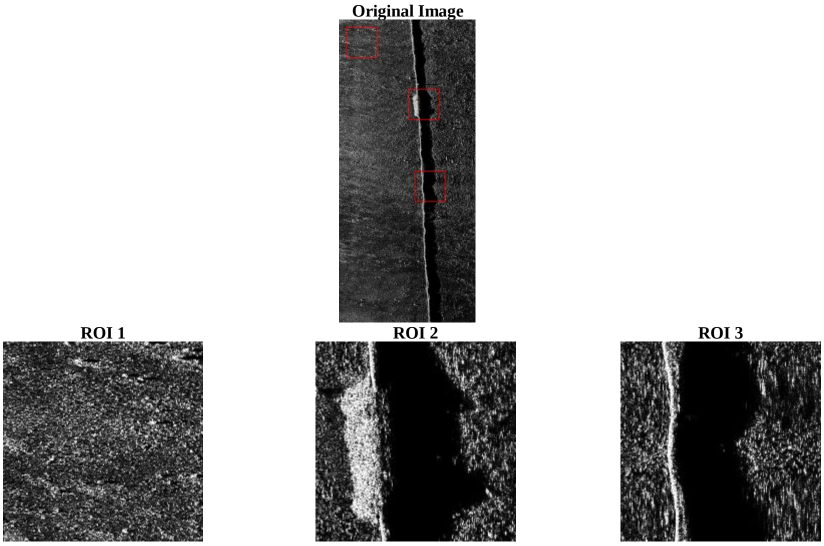


Fig. 8 *Three ROI selected from sonar image with an pipeline on seafloor.*

Regarding the ROI-1, the seabed reverberation is presented as a noisy image (speckle noise) with variations of the pixel values in any range of the gray scale. In the ROI-2, the underwater pipeline with rock dumping is presented with acoustic highlight close to the maximum value of the gray scale range. Also, the presence of rock dumping shows a pronounced acoustic shadow greater than that of the pipeline close to the minimum value of the gray scale range. In the ROI-3, the underwater pipeline deployed on the seafloor with free span is presented with acoustic reverberation between highlight and shadow. Note that the pipeline is not completely supported on the seafloor due to the presence of seabed reverberation between the pipeline and shadow.

Fig. 9 exhibits for each ROI previously introduced in Fig. 8, the following further details: the result image of k-*th* order statistic value x_k; the detection image applied the OS-CFAR 2D technique; and the comparison of standard and breakdown point versions to calculate the order statistic value x_k using the column-average bin-displacements measurement (vertical axis) versus the column number (horizontal axis). The parameters settings for OS-CFAR 2D were $k = 0.8$, $P_{fa} = 0.013$ and different radius sizes $r = 1, 3$ and 5. As it can be seen, when the radius size increases with a constant order statistic and false alarm probability the number of false detections decreases.

Considering the curves of ROI-1, it is shown for both versions (standard and breakdown point) that they are stable in the presence of seabed reverberation even with a large difference in the measurement of column-average bin-displacements.

ROI-2 and ROI-3, shows that the standard version has peaks close to the maximum value of the gray scale range due to the presence of the pipe and rock dumping, then it decays to the minimum value of the gray scale range because of the shadow. Finally it grows again due to the presence of seabed reverberation. Note that the rock dumping peak of the ROI-2 is much wider than the peak of pipeline in the ROI-3. On the other hand, the curves of breakdown point version are more stable in the presence of the pipeline and rock dumping, so they require a smaller amount of bin-displacements.

Towards the application of this approach in real time scenarios, these results show that the computational effort makes it feasible. However, notice should be taken in the fact that a fixed false alarm probability (P_{fa}) yields a fixed detection threshold ($\hat{T}$) for OS-CFAR, and consequently, false alarms may appear, as it was presented in [15]. Future work on adaptive detection threshold selection, in the sense proposed in [21], is still pending.

Fig. 10 exhibits the experimental results of processing time against radius size for the common implementation of the five OS-CFAR 2D versions utilized: (1) *Perreault and Herbert* method [30] (colour line blue), (2) *Weiss method* [28] (colour line black) (3) *bubble sort method* [25] (colour line cyan) (4) *bucket sort method* [25] (colour line green) and (5) proposed method (colour line red). Note that, only in this context the processing time makes sense as a measure of experimental efficiency.

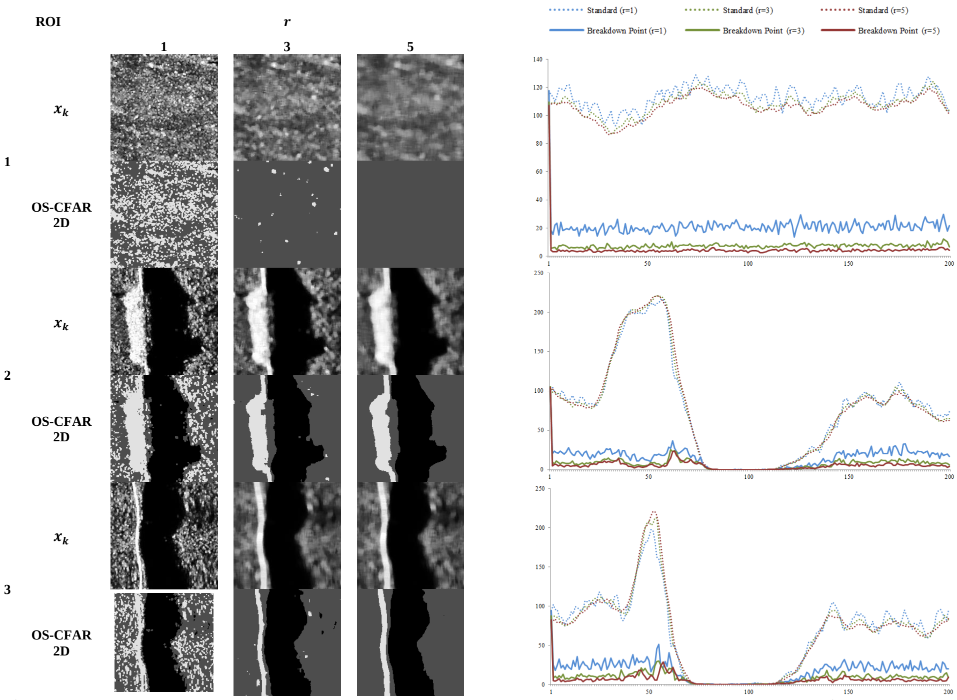


Fig. 9 *ROI of Fig. 8: the k-th order statistic value x_k image; the detection image result applying the OS-CFAR 2D technique; and the comparison of standard and breakdown point versions to calculate the order statistic value x_k using the column-average bin-displacements measurement (vertical axis) versus the column number (horizontal axis).*

These methods were tested on the Fig.5 whose size is 2000x900 (AxB) pixels from an 8 bit (b = 256) grayscale. In Fig. 10, the vertical axis indicates the processing time (in seconds) while the horizontal axis indicates the MF window's radius size. As shown, the processing time varies exponentially with the radius size for the classical bubble sort and bucket sort methods; the processing time varies linearly with the radius size for the Weiss method; conversely, the Perreault and Herbert and proposed method the processing time is constant due to depend on the number of scale levels ($b = 2^{image\ bit-depth}$) and not on the radius size. Worth noting that, the processing time of the proposed method is reduced by using distributive histograms and breakdown point optimal concept.

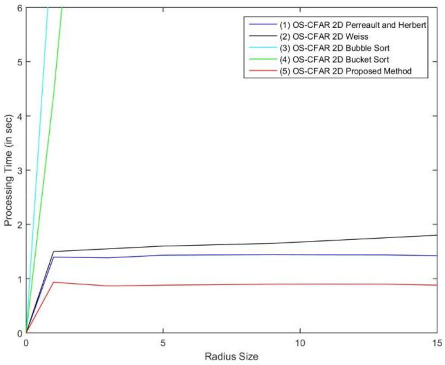


***Fig.10** Processing time vs. radius size for OS-CFR 2D using the (1) Perreault and Herbert method [30] (colour line blue), (2) Weiss method [28] (colour line black) (3) bubble sort method [25] (colour line cyan) (4) bucket sort method[25] (colour line green) and (5) proposed method (colour line red).*

6. Conclusion

The OS-CFAR 2D technique is extensively used in radar and sonar technology to cope with different applications although it requires higher computational effort. This work presents a new approach to improve the algorithmic efficiency of the OS-CFAR 2D using the distributive histograms and the optimal breakdown point concept. This approach was evaluated using the theoretical algorithm analysis as well as its experimental results on acoustic images. These studies demonstrate the great improvement obtained using the novel proposal presented for the OS-CFAR 2D technique, mainly from the computation effort standpoint.

7. References

1 Lurton, X.: 'An Introduction to Underwater Acoustics. Principles and Applications' (Springer, ch. 7 and 8, 2002, Springer Praxis Books in Geophysical Sciences)

2 Chapple, P.: 'Automated Detection and Classification in High-Resolution Sonar Imagery for Autonomous Underwater Vehicle Operations.' (Maritime Operations Division Defense Science and Technology Organization, 2008)

3 Carmichael, D.R., Linnett, L.M., Clarke, S.J., Calder, B.R.: 'Seabed Classification Through Multifractal Analysis of Sidescan Sonar Imagery'*IEE Proc. Radar Sonar Navig.*, 1996, **143**, (3), p. 140.

4 Mignotte, M., Collet, C., Perez, P., Bouthemy, P.: 'Sonar image segmentation using an unsupervised hierarchical MRF model'*IEEE Trans. Image Process.*, 2000, **9**, (7), pp. 1216–1231.

5 Cutter, G.R., Rzhanov, Y., Mayer, L.A.: 'Automated Segmentation of Seafloor Bathymetry from Multibeam Echosounder Data Using Local Fourier Histogram Texture Features'*J. Exp. Mar. Biol. Ecol.*, 2003, **285–286**, pp. 355–370.

6 Lianantonakis, M., Petillot, Y.R.: 'Sidescan Sonar Segmentation Using Texture Descriptors and Active Contours'*IEEE J. Ocean. Eng.*, 2007, **32**, (3), pp. 744–752.

7 Ye, X.F., Zhang, Z.H., Liu, P.X., Guan, H.L.: 'Sonar Image Segmentation based on GMRF and Level-Set Models'*Ocean Eng.*, 2010, **37**, (10), pp. 891–901.

8 Celik, T., Tjahjadi, T.: 'A Novel Method for Sidescan Sonar Image Segmentation'*IEEE J. Ocean. Eng.*, 2011, **36**, (2), pp. 186–194.

9 Richards, M.A.: 'Fundamental of Radar Signal Processing' (ch. 7, 2005, McGraw-Hill)

10 Levanon, N.: 'Radar Principles' (John Wiley and Sons, ch. 5 and 6, 1988)

11 Gao, J., Li, H., Chen, B., Zhou, T., Xu, C., Du, W.: 'Fast two-dimensional subset censored CFAR method for multiple objects detection from acoustic image'*IET Radar Sonar Navig.*, 2017, **11**, (3), pp. 505–512.

12 Acosta, G.G., Villar, S.A.: 'Accumulated CA–CFAR Process in 2-D for Online Object Detection From Sidescan Sonar Data'*IEEE J. Ocean. Eng.*, 2015, **40**, (3), pp. 558–569.

13 de Moustier, C.: 'OS-CFAR detection of targets in the water column and on the seafloor with a multibeam echosounder', in (2013)

14 Villar, S.A., Sousa, A.L., Rozenfeld, A., Acosta, G.G.: 'Pipeline Detection System in Acoustic Images Utilizing CA-CFAR', in OCEANS 2013 MTS/IEEE, (2013)

15 Villar, S.A., Acosta, G.G., Sousa, A.L., Rozenfeld, A.: 'Evaluation of an Efficient Approach for Target Tracking from Acoustic Imagery for the Perception System of an Autonomous Underwater Vehicle'*Int. J. Adv. Robot. Syst.*, 2013, **11**, (24), pp. 1–13.

16 Villar, S.A., Paula, M.D., Solari, F.J., Acosta, G.G.: 'A Framework for Acoustic Segmentation Using Order Statistic-Constant False Alarm Rate in Two Dimensions

From Sidescan Sonar Data'*IEEE J. Ocean. Eng.*, 2018, **43**, (3), pp. 735–748.

17 Rohling, H.: 'Radar CFAR Thresholding in Clutter and Multiple Target Situations'*IEEE Trans. Aerosp. Electron. Syst.*, 1983, **AES-19**, (4), pp. 608–620.

18 Mashade, M.B.E.: 'Detection performance of the trimmed-mean CFAR processor with noncoherent integration'*IEE Proc. - Radar Sonar Navig.*, 1995, **142**, (1), pp. 18–24.

19 Ritcey, J.A.: 'Performance Analysis of the Censored Mean-Level Detector'*IEEE Trans. Aerosp. Electron. Syst.*, 1986, **AES-22**, (4), pp. 443–454.

20 Nathanson, F.E., Reilly, J.P., Cohen, M.N.: 'Radar Design Principles' (McGraw-Hill, ch. 4, 1991, 2nd edn.)

21 Magaz, B., Belouchrani, A., Belouchrani, M.: 'Automatic Threshold Selection in OS-CFAR Radar Detection Using Information Theoretic Criteria'*Prog. Electromagn. Res. B*, 2011, **30**, pp. 157–175.

22 Blake, S.: 'OS-CFAR theory for multiple targets and nonuniform clutter'*IEEE Trans. Aerosp. Electron. Syst.*, 1988, **AES-24**, (6).

23 Cormen, T.H., Stein, C., Rivest, R.L., Leiserson, C.E.: 'Introduction to Algorithms' (McGraw-Hill Higher Education, 2001, 2nd edn.)

24 Skiena, S.S.: 'The algorithm design manual' (Springer-Verlag London, 2008, 2nd edn.)

25 Juhola, M., Katajainen, J., Raita, T.: 'Comparison of algorithms for standard median filtering'*Signal Process. IEEE Trans. On*, 1991, **39**, (1), pp. 204–208.

26 Tukey, J.: 'Exploratory Data Analysis' (Addison-Wesley, ch. 7, 1977)

27 Huang, T., Yang, G., Tang, G.: 'A Fast Two-Dimensional Median Filtering Algorithm'*IEEE Trans. Acoust. Speech Signal Process.*, 1979, **27**, (2), pp. 13–18.

28 Weiss, B.: 'Fast Median and Bilateral Filtering'*ACM Trans. Graph.*, 2006, **25**, (3), pp. 519–526.

29 Gil, J., Werman, M.: 'Computing 2-D min, median, and maxfilters'*IEEE Trans. Pattern Anal. Mach. Intell.*, 1993, **15**, (5), pp. 504–507.

30 Perreault, S., Hebert, P.: 'Median Filtering in Constant Time'*IEEE Trans. Image Process.*, 2007, **16**, (9), pp. 2389–2394.

31 Villar, S.A., Torcida, S., Acosta, G.G.: 'Median Filtering: a New Insight'*J. Math. IMAGING Vis.*, 2017, **57**, (1), pp. 130–146.

32 Huber, P.J.: 'Robust Statistics' (Wiley, 1981)

33 Bradski, G., Kaehler, A.: 'Learning OpenCV: Computer Vision with the OpenCV Library' (ch. 3, 4 and 5, 2008, 1st edn.)