HAL
open science

Adaptive Higher Order Sliding Modes for Two-Dimensional Derivative Estimation

Lilia Sidhom, Mohamed Smaoui, Daniel Thomasset, Xavier Brun, Eric Bideaux

To cite this version:

Lilia Sidhom, Mohamed Smaoui, Daniel Thomasset, Xavier Brun, Eric Bideaux. Adaptive Higher Order Sliding Modes for Two-Dimensional Derivative Estimation. 18th IFAC World Congress, Aug 2011, Milan, Italy. pp.3063-3071, 10.3182/20110828-6-IT-1002.03183 . hal-00747380

HAL Id: hal-00747380

https://hal.science/hal-00747380

Submitted on 30 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Higher Order Sliding Modes for Two-dimensional Derivative Estimation

L. Sidhom M. Smaoui D. Thomasset X. Brun E. Bideaux

AMPERE LABORATORY UMR 5005-CNRS - University of Lyon, INSA of Lyon, F-69621 FRANCE (e-mail: lilia.sidhom@insa-lyon.fr).

Abstract: The higher order sliding modes is some of recent technique which is used for the derivatives noisy signals estimation. In fact, this technique is well known mostly to elaborate the control laws and is also shown a good results in the synthesis of the rth order robust differentiators. In this paper, an extension of such technique to the two-dimensional case is investigated. In effect, the higher order sliding modes differentiators are used as a novel approach of edge finding in a gray scale image. The proposed algorithm use an adaptive mechanism for tuning up its parameters in real time, in order to increase the efficiency of basic scheme. Finally, to validate the efficiency of this new algorithm, some comparative study with a conventional edge detectors is performed.

Keywords: Adaptive higher order sliding modes, differentiator, two-dimensional signal, edge detection, Gradient and Laplacian operator, Canny filter.

1. INTRODUCTION

Digital images play an important role in daily life applications such as satellite television as well as in areas of research and technology. For each particular application, some problems can be treated such as the denoising, the compression and the edge detection problem. This last one, is a fundamental tool mostly used in image processing applications. Information obtained from the edges is considered as a precursor step to feature an extraction and an object segmentation. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity, [Chanda, B. and Majumdar, D.D., 2004].

There are many ways to perform edge detection. In traditional case, edge detection is implemented by convolving the signal with some form of linear filter, [Argyle, E. , 1971]. There are some filter which are based on the differential methods. These methods may be grouped into two categories, gradient and Laplacian method, [Argyle, E. , 1971], [Gonzalez, R.C., and Woods, R.E., 2002]. The gradient method detects the edges by looking for local extrema of the gradient of the intensity function. Examples of gradient-based edge detectors are Roberts, Prewitt, and Sobel operators, [Koplowitz, J. and Greco, V., 1994], [Ando, S., 2000]. The Laplacian method searches for zero crossings in the second derivative of the image in order to find edges, [Huertas, A. and Medioni, G. , 1986], such as the Marr-Hildreth operator, [Marr, D. and Hildreth, E., 1980].

A large number of edge detectors are available in the image processing which each one is designed to be sensitive to certain types of edges. Despite the popularity of the differential methods of edge detection, these detectors suffer from some number of problems. The main problem is a sensitive of such algorithm to the noise. According to the characteristics noise of the image, the result of detection edge can widely varies. In effect, the edge detection is a hard task with the noisy images which due to the high-frequency content of the both noise and edges. Indeed, for Gradient-based scheme, the size of the kernel filter and coefficients are fixed and can not be adapted to a given image. The noise derivative can mask the real maxima that indicate edges. This sensitivity to noise is of course more accentuated for the methods based on second derivative. To remedy of this drawback, other methods are developed in the literature. Some examples can be cited: Canny filter [Canny, J. , 1986], Deriche filter [Bourennane, E., Gouton, P., Truchetet, F., and Paindavoine, M., 1993].

Recently, robust method based on algebraic approach is proposed [Fliess, M., Join, C., Mboup, M., Sedoglavic, A. , 2005], [Riachy, S., Bachalany, Y., Mboup, M., and Richard, J.P., 2008]. Based on a truncated Taylor expansion, n^{th} order derivatives are obtained through iterative integrations on the input signal. In [Fliess, M., Join, C., Mboup, M., Sedoglavic, A. , 2005], estimation of the derivatives of a multidimensional signal was considered. In the last one, an application to the edge detection is also performed.

Higher Order Sliding Modes (HOSM) is applied in many application for the one-dimensional analysis, then this theory can be carried over to the two-dimensional domain (2D). The main objective of the current article consists to extend a new version of the robust differentiator defined in Levant, A. [2003] to the 2D context. This proposed algorithm adds the adaptive aspect to the classical scheme defined in order to improve the performance of this last one.

This paper is organized as follows. Section 2 presents the new version of the rth order robust algorithm already defined in Levant, A. [2003]. Section 3 is dedicated to implement the first and the second order algorithm of the proposed scheme in order to estimate respectively the gradient and the Laplacian of a considered image. Then, a comparative study is shown with a classic methods. In this

study, some criteria are chosen to evaluate the performance of each edge detector.

2. ADAPTIVE SLIDING MODE DIFFERENTIATOR

To design a differentiator scheme, some features of the signal and the noise must be considered, [Krener, A., 1980]. However, in some cases the structure of the signal may be unknown except some differential inequalities. In such case, the sliding mode technique can be used [Utkin, V.I., Guldner, J., and Shi, J. , 1999]. The main features of this last approach is its robustness to variation in system parameters, external disturbances and modeling errors with respect to the matching condition, [Utkin, V.I., Guldner, J., and Shi, J. , 1999]. As well as for the controllers synthesis [Smaoui, M., Brun, X., and Thomasset, D., 2005a], [Laghrouche, S., Smaoui, M., Plestan, F., and Brun, X., 2006] the sliding mode technique shows good results in the synthesis of robust differentiators [Smaoui, M., Brun, X., and Thomasset, D., 2005b], such as the Super Twisting (ST) algorithm [Levant, A., 1998]. In practice, the major problem associated this algorithm is the tuning up of its gains convergence. The accuracy of this differentiator depends on the choice of these parameters, [Smaoui, M., Brun, X., and Thomasset, D., 2005b].
In fact, the number of gains to be tuned is equal to $(r+1)$, for the rth order differentiator. So the choice of these parameters become a difficult task when the order of differentiator is increased. Furthermore their choices are not to be too large in order not to be differentiating the noise. Moreover it is not always easy to determine these gains for a given bandwidth of the input signal. A simple modification of the spectral content of the input signal or its amplitude can cause a significant error in the estimation of the derivative. In fact, the parameters of the algorithm strongly depend on the Lipschitz constant of the nth derivative of the input signal and generally this constant is not accurately known beforehand.
This problem can be resolved with adding an adaptive law to the gains of the basic algorithm. The adaptive tuning law ensures a convergence of the output signal toward the derivative input signal of the differentiator, in Lyapunov's sense. In [Suzuki, S., Furuta, K., Shiratori, S., 2003], the authors propose the adaptive form of the ST algorithm. Inspired from this work, a new version of an arbitrary order adaptive sliding mode differentiator is proposed. Indeed, the defined differentiator presents a simple form and easy design to use it in real-time.
Firstly, all the theory of the proposed algorithm is given on the one-dimensional domain (1D).
Let the input signal of differentiator $f(t)$ be a function defined on $[0,\infty[$ measurable in Lebesgues sense. This signal is considered as the sum of two terms:

$$f(t) = f_0(t) + \xi(t) \tag{1}$$

$f_0(t)$ is an unknown base signal with the nth derivative having a known Lipschitz constant $C > 0$. $\xi(t)$ is a bounded Lebesgue-measurable noise with unknown features, defined by: $\xi(t) < \epsilon$, with ϵ is sufficiently small.
In [Levant, A., 2003], an arbitrary higher order sliding modes differentiator is defined. In this paper, we proposed a new version of this r^{th} order differentiator in order to adjust on line the gains algorithm. The form of the proposed differentiator is given for any $i \in \{1, \cdots, n-1\}$ by:

$$\begin{cases} \dot{\hat{x}}_i = \hat{x}_{i+1} - \hat{\lambda}_i \left|s_i\right|^{\left(\frac{n+1-i}{n+2-i}\right)} sign(s_i) - K_i s_i \\ \dot{\hat{x}}_n = \hat{\lambda}_{n+1}\hat{x}_{n+1} - \hat{\lambda}_n \left|s_n\right|^{\frac{1}{2}} sign(s_n) - K_n s_n \\ \dot{\hat{x}}_{n+1} = -sign(s_n) \end{cases} \tag{2}$$

Where $s_1 = \hat{x}_1 - f$ and $s_i = \hat{x}_i - \dot{\hat{x}}_{i-1}$, $(i = 2, \cdots, n)$ are the sliding functions defining the estimation errors for each derivative order. K_i are positive gains ensure the convergence of the proposed algorithm.
$\hat{\lambda}_i$ $(i = 1, \cdots, n-1)$ are the dynamic gains of (2) computed in real time with using on the following adaptive laws:

$$\begin{cases} \frac{d}{dt}\hat{\lambda}_i = \left|s_i\right|^{\left(\frac{n+1-i}{n+2-i}\right)} sign(s_i) s_i, \quad (i = 1, \cdots, n-1), \\ \frac{d}{dt}\begin{bmatrix}\hat{\lambda}_n & \hat{\lambda}_{n+1}\end{bmatrix}^T = \begin{bmatrix}\left|s_n\right|^{\frac{1}{2}} sign(s_n) s_n & \int_{\mathbb{R}} sign(s_n) dt s_n\end{bmatrix}^T \end{cases} \tag{3}$$

With the equations (3), the system (2) is an rth order adaptive robust differentiator. The used function sign(.) is defined as following:

$$sign(.) = \begin{cases} 1, & if \quad (.) \geq 0 \\ -1, & if \quad (.) < 0 \end{cases} \tag{4}$$

At initial time $(t = 0)$, one has $\hat{x}_1(0) = f(0)$ and $\hat{x}_i(0) = \dot{\hat{x}}_{i-1}(0)$ with $i = \{2, \cdots, n+1\}$.
After the convergence time, the estimations of $\dot{f}_0(t)$, $\ddot{f}_0(t), \cdots, f_0^{(n)}(t)$ are an exact ones in the absence of measurement noises.

- **Convergence proof of the proposed algorithm**

The first step of the convergence proof consists to show that the estimation error relative to the first iteration $(i = 1)$ tends to zero.
For $i = 1$, the first equation of (2) becomes:

$$\dot{\hat{x}}_1 = \hat{x}_2 - \hat{\lambda}_1 \left|s_1\right|^{\left(\frac{n}{n+1}\right)} sign(s_1) - K_1 s_1 \tag{5}$$

The first derivative of the estimation error s_1 is given by:

$$\dot{s}_1 = \dot{\hat{x}}_1 - \dot{f} \tag{6}$$

Define a true value λ_1^* which is fixed one and verifies the following expression:

$$\dot{f} = \hat{x}_2 - \lambda_1^* \left|s_1\right|^{\left(\frac{n}{n+1}\right)} sign(s_1) \tag{7}$$

Indeed, this expression is derived from the non-adaptive differentiator form given in [Levant, A., 2003]. So this gain value ensures to obtain an exact estimate of the first derivative of the input signal.
Substituting (5) and (7) in (6):

$$\dot{s}_1 = -\tilde{\lambda}_1 \left|s_1\right|^{\left(\frac{n}{n+1}\right)} sign(s_1) - K_1 s_1 \tag{8}$$

With $\tilde{\lambda}_1 = \hat{\lambda}_1 - \lambda_1^*$. Define a first Lyapunov function as:

$$V_1 = \frac{1}{2}\left(s_1^2 + \tilde{\lambda}_1^2\right) \tag{9}$$

The derivative along the trajectory of the function V_1 is given by:

$$\begin{aligned} \dot{V}_1 &= s_1 \dot{s}_1 + \tilde{\lambda}_1 \dot{\tilde{\lambda}}_1 \\ &= s_1 \left[-\tilde{\lambda}_1 \left|s_1\right|^{\left(\frac{n}{n+1}\right)} sign(s_1)\right] - K_1 s_1^2 + \tilde{\lambda}_1 \dot{\tilde{\lambda}}_1 \\ &= -K_1 s_1^2 + \tilde{\lambda}_1 \left[-\left|s_1\right|^{\left(\frac{n}{n+1}\right)} sign(s_1) s_1 + \dot{\tilde{\lambda}}_1\right] \end{aligned} \tag{10}$$

With the below adaptive law:

$$\dot{\hat{\lambda}}_1 = |s_1|^{(\frac{n}{n+1})} sign(s_1) s_1 \tag{11}$$

Substituting (11) in (10):

$$\dot{V}_1 = -K_1 s_1^2 \tag{12}$$

Consequently $s_1 \to 0 \;\Rightarrow\; \dot{\hat{x}}_1 = \hat{x}_2 = \dot{f}$.
To prove the algorithm convergence for any iterations $i = \{2, \cdots, n\}$, it is sufficient to perform relatively in the same way and whenever it is necessary to consider the result obtained in the previous iteration.
In the general case and for any $j \in \{2, \cdots, n-1\}$, let's start by defining the true values λ_j^* by:

$$f^{(j)} = \hat{x}_{j+1} - \lambda_j^* |s_j|^{(\frac{n+1-j}{n+2-j})} sign(s_j), \;\; \forall j \in \{2, \cdots, n-1\}. \tag{13}$$

The role of these trues values gains is identical to the λ_1^* which is defined previously.
Let's define a Lyapunov function for the iteration j:

$$V_j = \frac{1}{2}\left(s_j^2 + \tilde{\lambda}_j^2\right) \tag{14}$$

The derivative of the above equation can be written by:

$$\begin{aligned} \dot{V}_j &= s_j \dot{s}_j + \tilde{\lambda}_j \dot{\hat{\lambda}}_j \\ &= s_j \left[\dot{\hat{x}}_j - \ddot{\hat{x}}_{j-1}\right] + \tilde{\lambda}_j \dot{\hat{\lambda}}_j \end{aligned} \tag{15}$$

For the system (2), the term of $\dot{\hat{x}}_j$ is replaced by its expression, then we have:

$$\dot{V}_j = s_j \left[\hat{x}_{j+1} - \lambda_j^* |s_j|^{(\frac{n+1-j}{n+2-j})} sign(s_j) - \ddot{\hat{x}}_{j-1}\right] - K_j s_j^2 \tag{16}$$

Or in the iteration $(j-1)$, the following result is already obtained:

$$\dot{\hat{x}}_{j-1} = f^{(j-1)} \tag{17}$$

Then this last equation leads to:

$$\ddot{\hat{x}}_{j-1} = f^{(j)} \tag{18}$$

With the equations (18) and (16), the following one is resulted:

$$\dot{V}_j = s_j \left[\hat{x}_{j+1} - \lambda_j^* |s_j|^{(\frac{n+1-j}{n+2-j})} sign(s_j) - f^{(j)}\right] - K_j s_j^2 \tag{19}$$

But reference to (13), this expression is deduced:

$$\hat{x}_{j+1} - \lambda_j^* |s_j|^{(\frac{n+1-j}{n+2-j})} sign(s_j) - f^{(j)} = 0 \tag{20}$$

Consequently $s_j \to 0 \;\Rightarrow\; \dot{\hat{x}}_j = \hat{x}_{j+1} = f^{(j)}, \; \forall j \in \{2, \cdots, n-1\}$.
For the last iteration $(i = n)$, the convergence is proved exactly as the previous method except that the term of $\dot{\hat{x}}_n$ is differently expressed. In effect, from the two last equations of (2), the expression of $\dot{\hat{x}}_n$ becomes:

$$\dot{\hat{x}}_n = -\hat{\lambda}_{n+1} \int_{\mathbb{R}} sign(s_n) dt - \hat{\lambda}_n |s_n|^{\frac{1}{2}} sign(s_n) - K_n s_n \tag{21}$$

The true values λ_n^* and λ_{n+1}^* of the last iteration are defined in order to verify the following equation:

$$f^{(n)} = -\lambda_{n+1}^* \int_{\mathbb{R}} sign(s_n) dt - \lambda_n^* |s_n|^{\frac{1}{2}} sign(s_n) \tag{22}$$

We have $s_n = \hat{x}_n - \dot{\hat{x}}_{n-1}$, so it is evident to write:

$$\dot{s}_n = \dot{\hat{x}}_n - \ddot{\hat{x}}_{n-1} \tag{23}$$

Or in the previous iteration the following equality holds $\dot{\hat{x}}_{n-1} = f^{(n-1)}$. Indeed (23) becomes:

$$\dot{s}_n = \dot{\hat{x}}_n - f^{(n)} \tag{24}$$

Substituting (21) and (22) in (24):

$$\dot{s}_n = -\tilde{\lambda}_{n+1} \int_{\mathbb{R}} sign(s_n) dt - \tilde{\lambda}_n |s_n|^{\frac{1}{2}} sign(s_n) - k_n s_n \tag{25}$$

Where $\tilde{\lambda}_n = \hat{\lambda}_n - \lambda_n^*$ and $\tilde{\lambda}_{n+1} = \hat{\lambda}_{n+1} - \lambda_{n+1}^*$. In this case the Lyapunov function is chosen as:

$$V_n = \frac{1}{2}\left(s_n^2 + \tilde{\lambda}_n^2 + \tilde{\lambda}_{n+1}^2\right) \tag{26}$$

The derivative of (26):

$$\dot{V}_n = s_n \dot{s}_n + \tilde{\lambda}_n \dot{\tilde{\lambda}}_n + \tilde{\lambda}_{n+1} \dot{\tilde{\lambda}}_{n+1} \tag{27}$$

This last one can be rewritten as:

$$\begin{aligned} \dot{V}_n = &-K_n s_n^2 + \tilde{\lambda}_n \left[-|s_n|^{\frac{1}{2}} sign(s_n) s_n + \dot{\tilde{\lambda}}_n\right] + \tilde{\lambda}_{n+1} \\ &\left[-\int_{\mathbb{R}} sign(s_n) dt s_n + \dot{\tilde{\lambda}}_{n+1}\right] \end{aligned} \tag{28}$$

By choosing the terms of $\dot{\tilde{\lambda}}_n$ and $\dot{\tilde{\lambda}}_{n+1}$ by their expression which are defined by (3). Then $\dot{V}_n = -K_n s_n^2$ and the convergence of s_n is ensured: $s_n \to 0 \;\Rightarrow\; \dot{\hat{x}}_n = f^{(n)}$. The including of this adaptive mechanism on this algorithm can emphasize the convergence of estimation error and it does not necessarily guarantee that the adaptive gains approach to their true values. Note that the gains K_i, $(i = 1, \cdots, n)$ are initialized only once.
To estimate the gradient and the Laplacian of the two-dimensional signal, only the first and the second order of the differentiator will be used. It suffices to replace n in the system (2) by 1 and 2 to obtain respectively the dynamic model of the used algorithms. A frequency responses of the two-order adaptive algorithm is done with different values of the convergence gains in [Sidhom, L., Pham, M.T., Thevenoux, F., and Gautier, M., 2010].

3. IMPLEMENTATION OF ALGORITHM IN TWO-DIMENSIONAL DOMAIN

In this paper, only the grayscale images (monochromatic) are considered. The image can be represented by a matrix or by a real function of two variables $g(x, y)$ defined in $(\mathbb{R} \times \mathbb{R}) \to \mathbb{R}^+$.
Where:
(x, y): are spacial coordinates of a pixel in the image.
The value of g at spatial coordinates (x, y) is a positive scalar quantity which represent a measure of the luminous intensity of a pixel.
The discretization of the used algorithm is made with basing on the explicit Euler method with sampling step equal to 10^{-3} second.
In the conventional approaches, the gradient or the Laplacian are computed by using a mask method, such as the Prewitt mask. However, the gradient and/or the Laplacian computation is done pixel by pixel for the proposed algorithm. In fact, this adaptive algorithm is initialized by the first pixel intensity value of the matrix $g(.)$. So the algorithm starts with a zero estimation error. Then a prediction of the intensity of the next pixel is made by applying the proposed differentiator to approximate the gradient and/or the Laplacian of the some input function $g(.)$.

Thereafter, the new value of the error will be considered which is defined by the difference between the predicted value and the true value recovered by the input matrix g. These steps are repeated for each row and column of the treated matrix. In other words, the discrete algorithm can be described by two loops where the luminous intensity of the next pixel is always predicted. For the second-order algorithm, the computation of the gradient and the Laplacian of $g(.)$ is performed in parallel, at the same loop. It is true that in the case of an image, the prediction seems unnecessary because all information can be retrieved directly from the input matrix. However, this application allows us to have an idea about the effectiveness of such algorithm for an image in order to extend the work to an image sequence. Indeed, it is much more interesting for the image sequence to predict the motion of each pixel in the sequence.

4. SIMULATION RESULTS

In order to show the effectiveness of the proposed algorithm, a comparative analysis with a classical methods is presented. In effect, four different edge detectors were selected for comparison: mask gradient, mask Prewitt, Canny filter and Laplacian operator. For this last one, the comparison is made with the Second Order Algorithm (SOA).

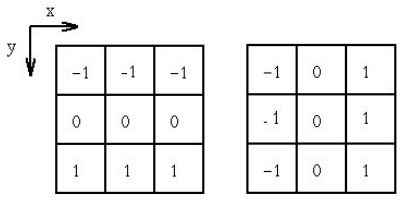
The mask gradient consists to combine at each pixel the convolution of the image with the two rectangular following masks $[-1, 1]$ and $[-1, 1]^T$, according to the considered direction. These kernels are designed to respond maximally to edges running horizontally and vertically relative to the pixel grid. These kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation. These can then be combined together to find the absolute magnitude of the gradient at each point. Typically, an approximate magnitude can computed using:

$$|G| = |\nabla g(x, y)| \approx \sqrt{(G_x^2 + G_y^2)}. \tag{29}$$

Where $G_x = \partial_x g(x, y)$ and $G_y = \partial_y g(x, y)$ are respectively the vertical and the horizontal gradient of the initial image. The Perwitt operator consists to convolve the input matrix by the 3×3 kernels as shown in Figure 1. So the gradient computation is conducted through these two masks, the first one performing the horizontal gradient, the second one the vertical gradient. One mask is simply the other rotated by 90. Compared to the gradient operator, the Prewitt mask have the advantage of producing two effects. Furthermore the gradient computation in one direction, these masks perform also smoothing in the orthogonal direction. This smoothing makes these masks somewhat less sensitive to noise than the gradient mask. Therefore, the Prewitt kernels are based on the idea of the central difference:

$$\frac{\partial g}{\partial x} \approx [g(x+1, y) - g(x-1, y)]/2. \tag{30}$$

The third edge detector used in this study is the Canny filter. This last one is the result of solving an optimization problem with some constraints. The imposed criteria are sensibility, localization and local unicity, [Canny, J. , 1986], [Muralidharan, R., and Chandrasekar, Dr.C., 2010]. This popular method for detecting edges begins by



(a) Prewitt mask for horizontal direction (b) Prewitt mask for vertical direction

Fig. 1. Prewitt masks.

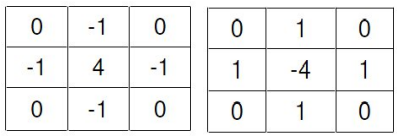
smoothing an image by convolving it with a Gaussian of a given δ value. With this smoothed image, derivatives in both x and y direction are computed. Thereafter, these derivatives are used to compute the gradient magnitude of the intensity image. In effect, the gradient is calculated using the derivative of the Gaussian filter. For a white noise, the Canny algorithm is known as the optimal edge detector, [Canny, J. , 1986]. This filter requires one adjustment parameter δ. With choosing a value of δ, x and y derivative masks are generated as outputs of the filter. The resulted masks are applied to the images using convolution.

As known, another alternative edge finding is to locate the zeros in the second derivative. The second derivative at any point in an image is obtained by using the Laplacian operator. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. Denote the Laplacian operator by $\nabla^2 g$ which is given by:

$$\nabla^2 g = \approx \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}. \tag{31}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. The commonly used small kernels are shown in Fig.2.

In order to extract the edge strength computed by one of



(a) Laplacian mask for horizontal direction (b) Laplacian mask for vertical direction

Fig. 2. Laplacian operator convolution mask

the cited methods, the next stage is to apply a threshold, to decide whether edges are present or not at an image point. The easiest way is to threshold the gradient image, assuming that all pixels having a local gradient above the threshold must represent an edge. When the threshold is low then more lines will be detected, and the results become increasingly susceptible to noise which conducts to picking out irrelevant features from the image. Conversely a high threshold may miss subtle lines, or segmented lines. The selection of a threshold value is an important design decision that depends on a number of factors, such as image brightness, contrast, level of noise, and even edge direction. Typically, the threshold is selected following an

analysis of the gradient image histogram. So, selection of threshold is an important parameter to get better performance for considered noisy images. The output of the thresholding stage is extremely sensitive and there are no automatic procedure that determines a valid threshold for all type of image.
Thus, in our simulation, the global thresholding histogram is used, so the thresholding is just applied to the gradient magnitude image. In effect, the threshold value S is determined by taking into account the distribution of the different values of the pixels in the image of the gradient module. The idea is to select a percentage of the most significant edges. In reality, with using this kind of thresholding method, the obtained edges will in general be thick. In our case no post-processing will be performed on the simulation results. The aim is to compare all algorithms with the same thresholding method.

4.1 Simulation results without noise

To validate the efficiency of the different edge detectors, simulation has been carried out using MATLAB image processing Toolbox. In the first time, a standard gray scale image "Lena" of size (256×256) (see Fig. 3(a)) is selected for simulation study.
Firstly, the image is processed without any noise. With the First Order Algorithm (FOA), the vertical and the horizontal gradient (see Fig. 3(b),3(c)) of the input matrix are estimated with imposed the convergence gain to $K_1 = 500$. Finally the gradient magnitude is normalized (see Fig. 4(a)))and threshold to find the edge in the image (see Fig. 4(b)).
Afterward, the Second Order Algorithm (SOA) is used to estimate the Laplacian relative to the initial image. The results are given by Figure 6. It seemed unnecessary to provide the first output of SOA because it is relatively identical to the output of FOA.
Therefore, we just present the results of FOA for the gradient computation.
In reality, it is very hard to determine the difference between each edge detectors, the obtained results are relatively the same (see Fig. 4(b) and 5). In fact, the absence of the truth edge map reveals some difficulties to compare the quality of the resulted edge maps.

4.2 Simulation results with noise

Then the evidence for the best detector is judged by studying the edge maps relative to some synthetic image. The edge of this last one is exactly known. This edge is so used as a reference for the comparison study. Corrupted an image with noise allows us to test the resistance of an edge detector operator to noise and assess the performance of various noise filters. Consequently, this test image was got corrupted with three different kinds of noise, generated using in MATLAB environment. Then for each of the three noisy images, the performance of the different edge detector was examined.
The test image is prone to different types of noises. Noise arises as a result of unmodeled processes going on in the production and capture of real signal. It may be caused by a wide range of sources: variation in the detector sensitivity, environmental variations, transmission or quantization

(a) Initial gray image: image 1

(b) G_x of image 1

(c) G_y of image 1

Fig. 3. horizontal G_x and Vertical G_y gradient of image 1: with FOA.

(a) Normalized magnitude of image 1

(b) Edge of image 1

Fig. 4. Edge of image 1: with FOA.

errors...
The noise can be described by an additive noise model, where the recorded image, $i(m,n)$ is the sum of the true image $t(m,n)$ and the noise $n(m,n)$:

$$i(m,n) = t(m,n) + n(m,n). \tag{32}$$

In this study, different types of noise are considered:
- Gaussian noise which has a zero-mean Gaussian distribution and is described by its standard deviation σ_n or variance, [Chang, S.G., Bin, Y., and Vetterli, M., 2006],
- Salt & Pepper noise is an idealized form of impulse noise model. The pixels values in grayscale image corrupted by various impulse noise models are generally replaced by values equal to or near the maximum or minimum of the allowable range. The strength of impulse noise is very high as compared to the strength of image signal,[Wang, Z., and Hang, D., 1999]. In fact, the noise impulses can be of negative or positive type. The negative impulses appear as black (pepper) points and positive impulses appear as white (salt) points in the image. An image contaminated by this kind of noise degrades by sharp and sudden dis-

(a) Result of Gradient operator

(b) Result of Prewitt operator

(c) Result of Canny filter: with $\delta = 0.8$

Fig. 5. Edge detection by Gradient mask, Prewitt mask and Canny filter.

(a) Normalized Laplacian of image 1

(b) Normalized Laplacian after thresholding

Fig. 6. Results with SOA: Laplacian estimation.

turbances in the image signal and it appears as randomly scattered white and black pixels over the image with some density d,

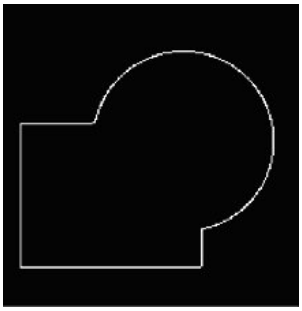
- Speckle noise is a multiplicative noise which adds to the image i using the equation $J = i + n_s i$, where n_s is uniformly distributed random noise with zero-mean and a variance σ_{n_s}, [Gagnon, L. and Smaili, F.D., 1996].

In [Abdou, I. E., and Pratt, W. K., 1979], [Heath, M., Sarkar, S., Sanocki, T., and Bowyer, K., 1998], [Maini, R. and Sohal, J.S., 2006], [Juneja, M., and Sandhu., P. S., 2009], different methods of comparative study for edge detectors were presented. To assess the quality of the edge maps of the exploited detectors, two kind of criteria are selected in our case: qualitative criteria such as complexity of the method, settings parameter... and quantitative criteria such as the parameters Mean Magnitude Error (MME), Mean Edge Error (MEE) and the computation time.

The MME indicates the average difference of the pixels between the exact magnitude of the image test and the estimated one. The MEE represents the error between the exact edge (see Fig. 7(b))and the obtained one. A higher MME and MEE indicates a greater difference between the desired and the processed image.

Upon the simulation tests, the MME and the MEE values

(a) Initial image: test image



(b) Exact edge of the test image

Fig. 7. Test image

for each kind of noise are summarized in Tables 1,2 and 3. The results of edge detection are shown in Fig. 9,11 and 13. It has been observed that the FOA works well both with the Gaussian, the Salt & Pepper as well as Speckle noise corrupted test image. Especially for the Salt & Pepper noise, the proposed algorithm does not amplify the noise in the resulted image, which is not the case for the others methods.

The performance of the Gradient mask is much superior in Speckle noise corrupted image as compared to the other noises. In effect, the Gradient mask is considered just as a high pass filter so it has not a smoothing action, which can explain the amplification of the noise, especially for the Salt & Pepper noise, (see Fig.11(a)).

However, the Prewitt filter and the Canny filter can estimate the Gradient of the input matrix with smoothing it, which explains the decrease of the noise on the edge detector image (see Fig. 11(b),11(c)). However, Prewitt mask presents the worst result for Salt & Pepper noise corrupted image compared to other methods. In fact, Prewitt edge detector makes an averaging of neighboring pixels. Since the Salt Pepper noise pixel values are often very different from the surrounding values, they tend to distort significantly the calculated average pixel by the averaging of neighboring pixels. Therefore the average value calculated will be significantly different from the true value. Then, performance of Prewitt edge detector decreases sharply for salt & pepper noise.

The performance of the Canny algorithm depends heavily on the adjustable parameter δ, which is the standard deviation for the Gaussian filter. δ also controls the size of the Gaussian filter. The bigger the value for this parameter, the larger the size of the Gaussian filter becomes. This implies more blurring, necessary for noisy images, as well as detecting larger edges. Smaller values of δ imply a smaller Gaussian filter which limits the amount of blurring and maintains finer edges in the image. The user can tailor the algorithm by adjusting this parameter to adapt to different type of image. In our case, the δ is fixed to 0.8.

The presence of a smooth action in the Prewitt mask and the Canny filter may add a slight phase shift that may well distort the estimate of the gradient. For the adaptive higher order algorithm, the reduction of the amplification noise and the differentiation action are combined and are

done simultaneously.

For the tables 1,2 and 3, it has been noticed that for the

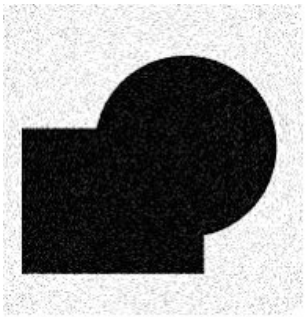
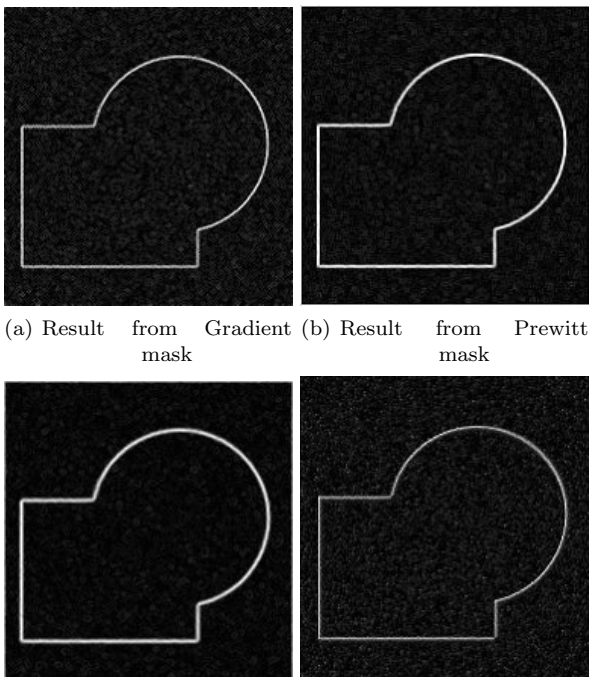


Fig. 8. Test image corrupted by Gaussian white noise with $\sigma_n = 0.01$



(a) Result from Gradient mask (b) Result from Prewitt mask

(c) Result from Canny filter (d) Result from FOA ($K_1 = 550$)

Fig. 9. Edge detection of the corrupted image by the Gaussian white noise.

Gaussian noise, the Canny filter presented the lowest value of the MEE. It is normally to have such result because the Canny filter is the optimal one for the such noise, [Canny, J. , 1986].

It is important to note that the obtained result can be

Table 1. Comparative table of errors: Gaussian white noise

Method	MME	MEE
Gradient mask	0.0574	0.0038
Prewitt mask	0.0709	0.0128
Canny filter	0.0579	0.0049
FOA($K_1 = 550$)	0.0468	0.0024

improved by changing the thresholding method already used. Then the MEE depends strongly on the choice of the threshold, then it is better to consider the second indicator

Table 2. Comparative table of errors: Salt & Pepper noise

Method	MME	MEE
Gradient mask	0.031	0.0135
Prewitt mask	0.0339	0.0144
Canny filter	0.0124	0.0177
FOA($K_1 = 400$)	0.0028	0.0033

MME. For improve the results of the MEE, other thresholding method can be used such hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. However such algorithm must be imposed two thresholds, so it must add two other paramaters to adjust. From the MME values, FOA performance is better for image corrupted with Salt & Pepper noise as compared to Gaussian noise or Speckle one.

In figure 14, the impact of the Salt & Pepper noise on

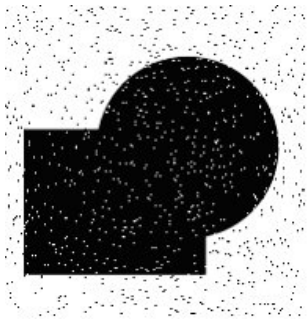
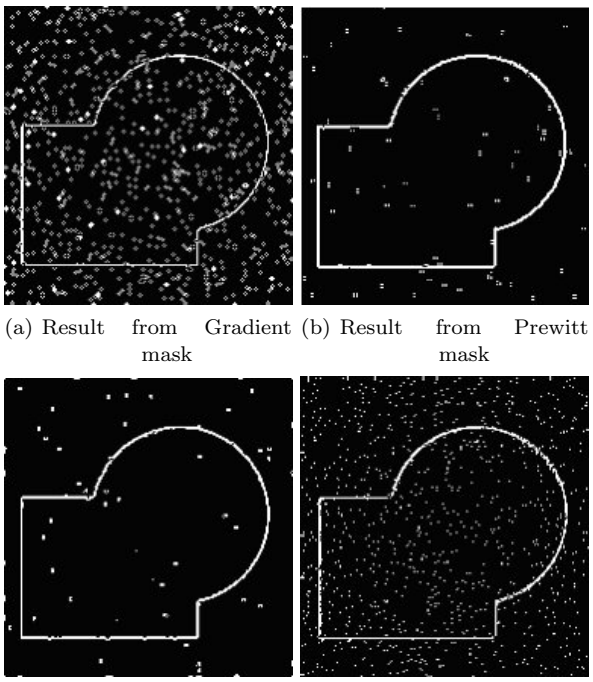


Fig. 10. Test image corrupted by Salt & Pepper noise with density $d = 0.01$



(a) Result from Gradient mask (b) Result from Prewitt mask

(c) Result from Canny filter (d) Result from FOA ($K_1 = 400$)

Fig. 11. Edge detection of the test image corrupted by Salt & Pepper noise.

the image "Lena" is often described the true image and the recorded one for each methods.

Table 3. Comparative table of errors: Speckle noise

Method	MME	MEE
Gradient mask	0.0288	0.0019
Prewitt mask	0.0465	0.0157
Canny filter	0.0112	0.000471
FOA($K_1 = 400$)	0.0235	0.000261

The figures 14(e), 14(f) show the importance of choice of the convergence gain of FOA on the rate of amplification noise in the resulted image. Indeed, increasing the value of the gain K_1, the amplifier noise reduction decreases. So this value does not be chosen so large. The same remark is valid for SOA, where two gains must be chosen in this case.

Table 4 summarizes the Elapsed time in second of

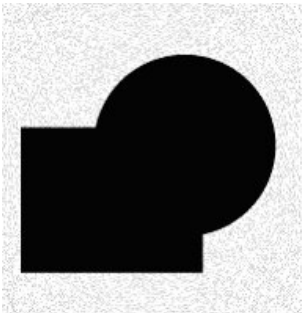
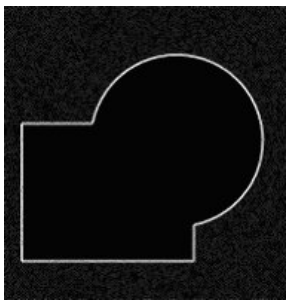
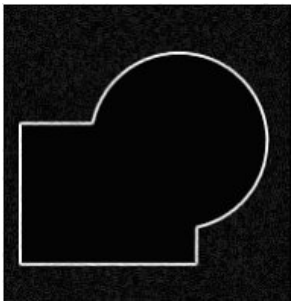


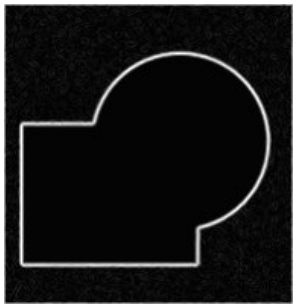
Fig. 12. Test image corrupted by Gaussian white with $\sigma_{n_s} = 0.01$



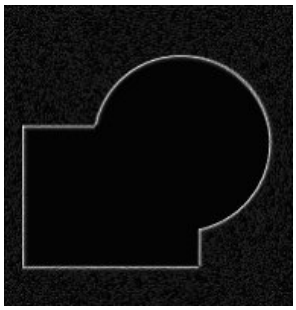
(a) Result from Gradient mask



(b) Result from Prewitt mask



(c) Result from Canny filter



(d) Result from FOA ($K_1 = 400$)

Fig. 13. Edge detection of the corrupted image by Speckle noise.

the used methods for different size of the input image. From these results, the convolution operation with one of the Gradient or Prewitt mask or with the derivative of a Gaussian for the Canny filter is less cumbersome

Table 4. Elapsed time (s) for different size of image

	Image size		
Method	$(50 * 50)$	$(125 * 255)$	$(256 * 256)$
Gradient mask	0.00057	0.0017	0.0098
Prewitt mask	0.0011	0.0023	0.0040
Canny filter	0.0043	0.0064	0.0150
FOA	0.0410	0.2556	1.0863
Laplacian Operator	0.009	0.0138	0.0166
SOA	0.0624	0.3930	1.6825

in calculation than the imbricated loops of the adaptive algorithms which the computation is done pixel by pixel. The proposed algorithm (FOA or SOA) is computationally more expensive compared to the other methods and the Elapsed time is quickly increases with the size of the image.

Although the proposed algorithm shows good results

(a) Image 1 corrupted by Salt & Pepper noise with density $d = 0.01$.

(b) Result from Gradient mask

(c) Result from Prewitt mask

(d) Result from Canny filter

(e) Result from FOA($K_1 = 650$)

(f) Result from FOA($K_1 = 350$)

Fig. 14. Edge detection of the corrupted image 1 by the Salt & Pepper noise.

compared to conventional methods, but it presents certain number of parameters to adjust and more lines to write for the implementation.

5. CONCLUSION

In this work various classic edge detection techniques were implemented and compared to the adaptive higher order sliding mode algorithms. The convergence proof of the proposed algorithm is carried out. The novelty of this paper concerns the implementation of the adaptive higher orders sliding modes to the application of processing image. The advantages of applying such algorithm as an edge detector can be summarized in two essential points: i) its use for different orientation (vertical, horizontal, diagonal), it is sufficient to change the gradient approximation that is used, ii) the proposed algorithms works quite well for digital images corrupted with Gaussian noise, Salt & Pepper noise and the Speckle noise. Due to pixel by pixel computation, the adaptive differentiator presents a more bigger computation time than the conventional methods which represents a limit of this algorithm. The results of this study are quite promising especially for video applications where the movement of a pixel can be estimated throughout the sequence. For future work, this algorithm can be used to predict the optical flow on the image sequence as the Kalman filter commonly used for this type of application. For optical flow estimation using the proposed algorithm, a compression of a sequence images can then be made.

REFERENCES

Sidhom, L., Pham, M.T., Thevenoux, F., and Gautier, M. Identification of a robot manipulator based on an adaptive higher order sliding modes differentiator. In *IEEE/ASME Inter. Conf. on Advanced Intelligent Mechatronics (AIM)*, Montreal, 6-9 July 2010.

Wang, Z., and Hang, D. Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal processing*, 46(1):78–80, 1999.

Abdou, I. E., and Pratt, W. K. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5):753763, May 1979.

Ando, S. Consistent gradient operators. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 22, pages 252–265, March 2000.

Argyle, E. . Techniques for edge detection. *Proc. IEEE*, 59:285–286, 1971.

Bourennane, E., Gouton, P., Truchetet, F., and Paindavoine, M. Generalisation du filtre de canny deriche pour la detection des contours de formes quelconques. In *14th colloque GRETSI -Juan-Les-pins*, Universit de Bourgogne Laboratoire G.E.R.E, 1993.

Canny, J. . A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679714, 1986.

Chanda, B. and Majumdar, D.D. *Digital Image Processing And Analysis.* PHI, second (2nd) edition, 2004.

Chang, S.G., Bin, Y., and Vetterli, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. On Image Proces.*, 9:1532–1546, 2006.

Fliess, M., Join, C., Mboup, M., Sedoglavic, A. . Estimation des derivees d'un signal multidomensionnel avec application aux images et aux videos. In *Actes 20e Col. GRETSI, Louvain-la-neuve :available at http ://hal.inria.fr/inria00001116*, 2005.

Gagnon, L. and Smaili, F.D. Speckle noise reduction of airborne sar images with symmetric daubechies wavelets. *Proceedings of SPIE*, 2759:14–24, 1996.

Gonzalez, R.C., and Woods, R.E. *Digital image proce.* New Jersey, 2002.

Heath, M., Sarkar, S., Sanocki, T., and Bowyer, K. Comparison of edge detectors : a methodology and initial study. *Computer Vision and Image Understanding*, 69 (1):38–54, 1998.

Huertas, A. and Medioni, G. . Detection of intensity changes with sub-pixel accuracy using laplacian-gaussian masks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(5):651–664, 1986.

Juneja, M., and Sandhu., P. S. Performance evaluation of edge detection techniques for images in spatial domain. *Inter. J. of Comp. Theory and Engineering*, 1:614–621, 2009.

Koplowitz, J. and Greco, V. On the edge location error for local maximum and zero-crossing edge detectors. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 16, pages 1207–1212, December, 1994.

Krener, A. Kalman-bucy and minimax filtering. *IEEE Trans. on Automatic Control*, AC-25(2):291–292, 1980.

Laghrouche, S., Smaoui, M., Plestan, F., and Brun, X. Higher order sliding mode control based on optimal approach of an electropneumatic actuator. *Inter. J. of Control*, 79(2):119–131, 2006.

Levant, A. Robust exact differentiation via sliding mode technique. *Automatica*, 34:379–384, 1998.

Levant, A. Higher-order sliding modes, differentiation and output-feedback control. *Inter. J. of Control*, 76:924–941, 2003.

Maini, R. and Sohal, J.S. Performance evaluation of prewitt edge detector for noisy images. *Inter.J.on Graphics, Vision and Image Processing*, 6(3):39–46, 2006.

Marr, D. and Hildreth, E. Theory of edge detection. *Proc. of the Royal Society of London.*, 207(1167):187–217, February 1980.

Muralidharan, R., and Chandrasekar, Dr.C. Edge detection using the cannys edge detector with user defined filter to obtain good feature for object recognition. In *Proceedings of the Int. Conf. on Information Science and Applications*, Chennai, India., February 2010.

Riachy, S., Bachalany, Y., Mboup, M., and Richard, J.P. An algebraic method for multi-dimensional derivative estimation. In *16th IEEE Mediterranean Conf. on Control and Automation*, Ajaccio, Corsica, France, 2008.

Smaoui, M., Brun, X., and Thomasset, D. A combined first and order sliding mode approach for position and pressure control of an electropneumatic system. In *In American Control Conf.*, pages 3007–3012, Portland, Oregon, USA, 2005a.

Smaoui, M., Brun, X., and Thomasset, D. A robust differentiatorcontroller design for an electropneumatic system. In *Proceeding of the 44th IEEE Conf. on Decision and Control, and the European Control Conf.*, Sevilia, Spain, 2005b.

Suzuki, S., Furuta, K., Shiratori, S. Adaptive impact shot control by pendulum-like juggling system. In *Inter. Conf. on Motion and Vibration Control*, volume 46, pages 973–981, Japon, 2003.

Utkin, V.I., Guldner, J., and Shi, J. . *Sliding mode control in electromechanical systems.* Number 338. 1999.