Adaptive 2-D LMS Filter Embedded Edge Detection Application

Ufuk Paralı¹*

¹ Ardahan University, Engineering Faculty, Electrical and Electronics Engineering Department, Ardahan, Türkiye (ORCID: 0000-0003-0088-2317)

(Bu yayın 26-27 Haziran 2020 tarihinde HORÁ-2020 kongresinde sözlü olarak sunulmuştur.)

(DOI: 10.31590/ejosat.780103)


Abstract

In this numerical study the effect of embedding two-dimensional least mean square (TDLMS) adaptive filter into various edge detection systems is discussed. TDLMS and edge detection modules are arranged in the system scheme in a manner such that they work sequentially. TDLMS algorithm is commonly used in many various image processing applications. Due to its ability of updating filter coefficients without needing any a priori assumptions, TDLMS provides superior advantage in 2-D signal processing applications. We investigated the performance increment of TDLMS especially on the commonly used edge detection algorithms in the literature such as Canny, Sobel, Prewitt, Roberts and LoG (Laplacian of Gaussian). It is observed that embedding TDLMS is particularly useful in edge detection for low SNR images comparing to high SNR images. The simulation results clearly show TDLMS filter provides significant improvement for the edge detection implementation on a relatively lower SNR case comparing to a higher SNR case. Especially, TDLMS embedded Sobel, Prewitt and Roberts implementations have relatively better results than TDLMS embedded Canny and LoG implementations for a low SNR image. On the other hand, for relatively higher SNR case, embedding TDLMS filter into the edge detection system does not provide as much significant improvement as in relatively lower SNR case. But still, for a high SNR case, TDLMS embedded Canny implementation have relatively better results than TDLMS embedded Sobel, Prewitt, Roberts and LoG implementations.

Keywords: 2-D Adaptive Filter, TDLMS, Edge Detection Algorithm, Canny, Sobel, Prewitt, Roberts, LoG.

Adaptif 2-D LMS Filtre Gömülü Kenar Algılama Uygulaması

Öz


Anahtar Kelimeler: 2-D Adaptif Filtre, TDLMS, Kenar Algılama Algoritması, Canny, Sobel, Prewitt, Roberts, LoG.

* Corresponding Author: Ardahan University, Engineering Faculty, Electrical and Electronics Engineering Department, Ardahan, Türkiye, ORCID: 0000-0003-0088-2317, ufukparali@ardahan.edu.tr
1. Introduction

1-D least mean square (LMS) adaptive filtering techniques have been successfully used in many signal processing applications (Hadhoud & Thomas, 1988, 1989). LMS is a quadratic approximation algorithm (Hadhoud & Thomas, 1987; Lin, Nie, & Unbehauen, 1993). Here, the weights of the adaptive filter are updated towards a minimum of a performance surface for achieving the convergence (Smith & Campbell, 1995). In adaptive LMS filtering, without making any a priori assumptions, the algorithm of the adaptive filters can automatically match or track an unknown signal to be processed. This is the main characteristic ability of the adaptive filters where the coherency of the signal and noise components are being utilized in the mechanism for updating the filtering coefficients of the adaptive algorithm (Soni, Rao, Zeidler, & Ku, 1991; Bae, Zhang, & Kweon, 2012). Since it is not necessary to make any a priori assumptions provides the robustness of the adaptive algorithms (Kaur, Malhotra, & Kaur, 2015; Verma, Singh, & Thoke, 2015). Such an ability in updating the coefficients of a filter is also very advantageous in 2-D signal processing applications (Liu, Liu, & Pan, 2014). Thus, for image processing applications, the popular LMS adaptive algorithm has been extended into 2-D LMS (TDLMS) algorithm (Smith & Campbell, 1995). TD LMS and some of its various modified versions are mainly used in filtering, image registration, image enhancement, noise reduction and noise cancelling in image processing (Lin, Nie, & Unbehauen, 1993). One of the most important issue in image processing during filtering the noise is to maintain the quality of image signal. Statistically, sharp variations due to edges are common features for image signals (Bae, Zhang, & Kweon, 2012). Thus, the image quality dramatically decreases in noise filtering process for constant coefficient filters (Soni, Rao, Zeidler, & Ku, 1991). However, in some specific applications, edge detection could be very critical and these edges must be preserved in the image signal (Verma, Singh, & Thoke, 2015). For such cases where the edge detection and edge information are critical, instead of a constant coefficient filter, utilization of TD LMS adaptive filter must be preferred (Verma, Singh, & Thoke, 2015). By iteratively adjusting its filter parameters, adaptive TD LMS algorithm can follow abrupt alterations in the input image statistics. This provides decrement of distortion in the image signal (Prameeth, Rao, & Srinivas, 2011).

The purpose of this numerical study is to increase the performance of edge detection algorithms by embedding TD LMS adaptive filter into the edge detection system. For improvement, we take into consideration the Canny, Sobel, Prewitt, Roberts and LoG (Laplacian of Gaussian) edge detection algorithms that are commonly used in the literature.

The rest of this paper is organized as follows. In Section II, first the mathematical background of edge detection algorithms and the TD LMS adaptive filters are explained. Then, the methodology for the TD LMS embedded edge detection system scheme is described. In Section III, the implementation results are given and discussed. Finally, Section IV provides the conclusions and the proposed future studies of this work.

2. Material and Methodology

2.1. Mathematical Background

2.1.1. Edge Detection Algorithms

The edge detection procedure has three steps which are noise smoothing, edge detection enhancement and edge localization. In an edge detection application, the main tradeoff is between the concepts of detection and localization. It is not possible to improve these two criteria simultaneously (Trucco & Verri, 1998). In an optimal edge detector, for a good detection, the probability of missing real edges and detecting edges due to noise must be minimized. Besides, for a good localization, the detected edges must be as close as possible to the true edges (Haralick & Shapiro, 1992). Several edge detection filter algorithms have been developed in the literature to reach an optimal comprise between the aforementioned criteria (Pratt, 1991). The mostly used edge detector algorithm in machine vision applications is the Canny edge detector (Plataniotis, Rao, & Srinivas, 2011; Canny, 1986). In this algorithm, a Gaussian smoothing is applied (convoluted) to an input intensity image \( I \) corrupted by noise to obtain (Trucco & Verri, 1998):

\[
J = G \ast I
\]

(1)

where \( G \) is a Gaussian with zero mean and standard deviation \( \sigma \). Here, the value of the \( \sigma \) is chosen with respect to the noise level and localization-detection tradeoff (Pratt, 1991). Once \( J \) is calculated, image gradient components \( J_x \) and \( J_y \) are computed for each pixel \((i, j)\) by convolving the rows and columns with the following mask (or so to say stencil) (Trucco & Verri, 1998):

\[
\begin{bmatrix}
1 & 0 & -1
\end{bmatrix}
\]

(2)

Then, the edge strength \( e_s(i,j) \) and edge normal orientation \( e_o(i,j) \) are evaluated as follows (Canny, 1986):

\[
e_s(i,j) = \sqrt{J_x^2(i,j) + J_y^2(i,j)}
\]

(3)

\[
e_o(i,j) = \arctan \frac{J_y}{J_x}
\]

(4)

Here, the values obtained from equations (3) and (4) form the strength image \( E_s \) and the orientation image \( E_o \), respectively. Depending on the application, two more steps named as nonmaximum suppression as the first step and hysteresis thresholding as the second step are applied in Canny edge algorithm (Canny, 1986). In the first step, strength image \( E_s \) is utilized in nonmaximum suppression where
the bulges around the local maxima are diminished to get 1-pixel wide edges (Trucco & Verri, 1998). The output of the first step is then utilized by the second step where the local maxima are discarded by eliminating pixels having a value less than a specific threshold (Haralick & Shapiro, 1992). Another commonly used edge detection algorithm is the Sobel algorithm where after Gaussian noise smoothing (as in eq 1) of the input intensity image, the following stencils are used as masks for filtering (Gupta & Mazumdar, 2013):

\[
\begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix}
\]

(5)

and

\[
\begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix}
\]

(6)

Using these masks, we obtain \( J_1 \) and \( J_2 \), respectively. Then we calculate the gradient magnitude of each pixel \((i,j)\) as in equation (3).

Two other commonly used edge filters in the literature are Prewitt and Roberts algorithms (Shrivakshan & Chandrasekar, 2012). They are similar to Sobel filter with only difference are the utilized stencil masks (Plataniotis & Venetsanopoulos, 2013). In Prewitt algorithm the following stencils are used (Shrivakshan & Chandrasekar, 2012):

\[
\begin{bmatrix}
-1 & -1 & -1 \\
0 & 0 & 0 \\
-1 & -1 & -1
\end{bmatrix}
\]

(7)

and

\[
\begin{bmatrix}
-1 & 0 & -1 \\
-1 & 0 & -1 \\
-1 & 0 & -1
\end{bmatrix}
\]

(8)

On the other hand, the following stencils are used as filter masks in Roberts algorithm (Shrivakshan & Chandrasekar, 2012):

\[
\begin{bmatrix}
1 & -1 \\
-1 & 1
\end{bmatrix}
\]

(9)

\[
\begin{bmatrix}
-1 & 1 \\
1 & -1
\end{bmatrix}
\]

(10)

Finally, the last filter we take into consideration in this study is the LoG edge detection algorithm. In this method, similar to the other techniques, the intensity image is first convolved with a Gaussian filter as in eq 1. Then the enhancement step is applied where the second derivative (Laplacian in two dimensions) is calculated by (Trucco & Verri, 1998; Kumar & Saxena, 2013):

\[
\nabla^2 f = g = \nabla^2 [G * I].
\]

(11)

Due to the linearity of the operators, the convolution and Laplacian are interchangeable. Thus the laplacian of Gaussian is evaluated in this step as follows:

\[
g = [\nabla^2 G] * I.
\]

(12)

Here, the 2-D LoG function with standard Gaussian deviation has the following form (Shrivakshan & Chandrasekar, 2012):

\[
LoG(x, y) = -\frac{1}{\pi\sigma^2} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}.
\]

(13)

In the intensity image, the response of the LoG operator will be zero where the gradient of the intensity is zero (Kumar & Saxena, 2013). On the other hand, around the points where intensity varies, the LoG function will have a positive response for the darker part and a negative response for the lighter part (Haralick & Shapiro, 1992; Kumar & Saxena, 2013).

2.1.2. TDLMS Adaptive Filter

Figure 1 below shows the general scheme of TDLMS adaptive filter where the filter output signal \( y(m,n) \) is evaluated via the dot product of the Wiener filter weight coefficients matrix \( W \) with the reference input signal matrix as follows (Hadhoud & Thomas, 1988, 1989):
\[ y(m, n) = \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} W(l, k) x(m - l, n - k). \]  

Using equation (14), at any iteration step, the error signal \( e \) is calculated by the following formula:

\[ e = d(m, n) - y(m, n). \]  

Once error signal is obtained, the updating equation for the weight coefficients matrix of Wiener filter can be defined by:

\[ W_{j+1}(l, k) = W_j(l, k) + 2\mu e_j x(m - l, n - k) \]  

where \( j \) is the iteration number, \( e_j \) is the error signal calculated at the \( j^{th} \) iteration and \( \mu \) is the scalar multiplier ranging between \(-10^{-8}\) to \(-10^{-12}\) in the literature (Hadhoud & Thomas, 1988; Lin, Nie, & Unbehauen, 1993). In other words it is a step size value that controls the convergence rate, filter stability and the residual error in the steady state of the adaptive procedure (Hadhoud & Thomas, 1988; Smith & Campbell, 1995). The Wiener algorithm aims to reach a set of weights minimizing the mean square error (MSE) given by:

\[ MSE = E(e^2). \]  

Here, \( E(e^2) \) is the expectation value of the square of the error signal (Lin, Nie, & Unbehauen, 1993). In equation (16), for determining the value of \( \mu \), there are various application dependent techniques. Among these methods, due to providing a high performance in the convergence of \( MSE \) in the iteration limit and a better noise cancellation, the trial and error technique has been one of the most commonly used procedure in the literature (Kaur, Malhotra, & Kaur, 2015; Verma, Singh, & Thoke, 2015). In this study \( \mu \) is determined to be \(-10^{11}\) by trial and error.

2.2. Methodology

Figure 2 shows the steps of the application in this study. The first block is the mean filter where for a low computational cost, it is applied directly for the whole image instead of applying it to each chosen window. The second block is windowing which provides the delay components of the reference input signal, \( x \). The primary input is obtained by contaminating the mean filtered original image by AWGN with a variance of \( 1.6\times10^5 \) (Hadhoud & Thomas, 1988). The results have been found for a Wiener filter of 10x10 dimension where the initial filter coefficients are set randomly in Matlab. The reference input matrix and the Wiener filter coefficient matrix must have the same dimensions. The iteration has been terminated in TDLMS block when the \( MSE \) settles down a stationary value. After completing the iteration steps in TDLMS block, the filter output \( y(m, n) \) is sent to Edge Detector block. In this block, the aforementioned edge filter algorithms take the \( y(m, n) \) as input and gives the detected edges as output.
3. Results and Comparison

In figure 3, the images chosen for the edge detection implementation are shown. The reason for choosing these figures is that, while figure 3(a) has densely distributed and slim edges, figure 3(b) has relatively less densely distributed and coarse edges. On the other hand, figure 3(c) has relatively higher amount of noise contamination than the other two figures. In order to have a detailed comparison, we first apply the edge detection algorithms (Canny, Sobel, Prewitt, Roberts, LoG) separately on each image without utilizing TDLMS filter. Then, we used the procedure depicted in figure 2.

![Image 3](image-url)

**Figure 3.** Images used for TDLMS embedded edge detection application (a) Regular eye image. (b) Jet turbine image. (c) Transmission electron microscope image of SARS-CoV-2 (Covid-19) (Yan, Shin, Pang, Meng, Lai, & Pang, 2020)

Below, figures 4(a), 4(b), 4(c), 4(d) and 4(e) give the edge detection results for figure 3(a) using Canny, Sobel, Prewitt, Roberts and LoG algorithms without TDLMS filter, respectively. On the other hand, figures 4(f), 4(g), 4(h), 4(i) and 4(j) give the edge detection results for the same algorithms, this time with TDLMS filter, respectively. Figures 5 and 6 depict the results for figures 3(b) and 3(c), respectively, in the same manner.

The results clearly show that TDLMS filter provides significant improvement for the edge detection implementation (see figures 6(f), 6(g), 6(h) and 6(i)) on figure 3(c) which has relatively lower SNR comparing to figures 3(a) and 3(b). Especially, TDLMS embedded Sobel, Prewitt and Roberts implementations have relatively better results than TDLMS embedded Canny and LoG implementations for a low SNR case. On the other hand, for figures 3(a) and 3(b) which have both relatively higher SNR than figure 3(c), embedding TDLMS filter into the edge detection system does not provide as much significant improvement as in figure 3(c). Besides, for a high SNR case, TDLMS embedded Canny implementations have relatively better results (see figures 4(f) and 5(f)) than TDLMS embedded Sobel, Prewitt, Roberts and LoG implementations.
Figure 4. Edge detection results for figure 3(a). (a) Canny filter. (b) Sobel filter. (c) Prewitt filter. (d) Roberts filter. (e) LoG filter. (f) TDLMS embedded Canny filter. (g) TDLMS embedded Sobel filter. (h) TDLMS embedded Prewitt filter. (i) TDLMS embedded Roberts filter. (j) TDLMS embedded LoG filter.
Figure 5. Edge detection results for figure 3(b). (a) Canny filter. (b) Sobel filter. (c) Prewitt filter. (d) Roberts filter. (e) LoG filter. (f) TDLMS embedded Canny filter. (g) TDLMS embedded Sobel filter. (h) TDLMS embedded Prewitt filter. (i) TDLMS embedded Roberts filter. (j) TDLMS embedded LoG filter.
Figure 6. Edge detection results for figure 3(c). (a) Canny filter. (b) Sobel filter. (c) Prewitt filter. (d) Roberts filter. (e) LoG filter. (f) TDLMS embedded Canny filter. (g) TDLMS embedded Sobel filter. (h) TDLMS embedded Prewitt filter. (i) TDLMS embedded Roberts filter. (j) TDLMS embedded LoG filter.
4. Conclusion

The ability of updating filter coefficients without needing any a priori assumptions is very advantageous in 2-D signal processing applications. Thus, 2-D LMS (TDLMS) algorithm is commonly used in many various image processing applications. In this work, we analyzed the effect of TDLMS algorithm on commonly used edge detection algorithms. For this purpose, we designed a procedure in which we embedded TDLMS and edge detector algorithm such that they work sequentially.

The analysis has shown that embedding TDLMS filter is particularly useful in edge detection implementation of relatively low SNR images comparing to high SNR images. Significantly, the performance of Sobel, Prewitt and Roberts algorithms increase more than the TDLMS embedded Canny and LoG implementations.

As a future work, we aim to improve the TDLMS filter having relatively higher coherency for the aforementioned edge detector algorithms especially to be utilized for detecting the edges of infrared (IR) and ultraviolet (UV) micron-sized targets.

References


