Applying First Order Filter into Digital Images

Raya Basil Al-Othman
University of Mosul, Duhok
College of Pure Sciences of Education
Computer Science Department

ABSTRACT

With the development of computer sciences, technology and the frequent use of computers in general and the use of image processing in particular, this paper presented a new idea about the use of a filter which is famous First Order Filter in digital image processing of various kinds, where scarce but almost non-existent use of this type of filters with digital images commonly used where this filter with text electronic circuits and electronics in general. This research is the foundation stone at the beginning of the application of this type of filters with all types of digital images and the development of such use with other types of data such as (Audio, Video….etc.) through the language of MATLAB.

General Terms

Digital image processing, First order Filter, Discrete-Time Filter.

Keywords

First Order Filter, Image processing, Digital images, MATLAB language, FIR, IIR.

1. INTRODUCTION

The using of one type of filters is First Order Filter is define number of active element of filter and capacity of First Order Filter and utilized to reduce the signal amplitude in half, the processing of digital images distorted and improve, prove the viability of using this type of filters on digital images, and the foundation stone work do not other different types of data, the programming language used is MATLAB , MATLAB can be used to design standard frequency selective filters that meet user-specified requirements [3].

2. FIRST ORDER FILTER

To explain about First Order Filter, the following example is showing [3]:

For i from 1 to n

Y [i]:=y [i-1] + a *(x[i] –y [i-1])

There are two types of first order filters: First order low pass filters and First order high pass filters.

2.1 First order low pass filters

First order low pass filters can be defined as the following [3]:

H(s) $= \frac{wc}{S+wc} = \frac{1}{1+s/wc}$ **……………… (1)**

2.2 First order high pass filters

First order high pass filters can be defined as the following [3]:

H(s) $= \frac{s}{S+wc} = \frac{s/w}{1+s/wc}$ **……………… (2)**

3. DISCRETE-TIME FILTER OVERVIEW

Discrete time filter can be defined as the following [3]:

$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} bk\, x[n-k]$ **……………. (3)**

$Y(e^{jw}) = \frac{\sum_{k=0}^{M} b_k e^{-j\omega k}}{\sum_{k=0}^{N} a_k e^{-j\omega k}}\ \mathrm{X}\ (e^{jw})$**…………….. (4)**

-Discrete-time filters is divided into two types:

1. Finite impulse response (FIR): h[n] =0 for some a and b such that $-\infty < a < n < b < +\infty$

2. Infinite impulse response (IIR): not FIR

- Filters that can be described with difference – equations

FIR: N=0

IIR: N>0

-A simple **(FIR)** filter is the moving average filter.

-A simple **(IIR)** filter is the first-order low pass filter that using.

The specified filter equation in the following used in calculating one store data as in figure (1) and figure (2),and see in the section 7” part of the program that plot the relationship between cutoff frequency and a” .

Y[n] - ay [n-1] = (1-a) x[n] ………………… (5)

4. THE USING FILTER IN IMAGE

This research excellence in the application First Order Filter on digital images where instead of data of store, it was taking a photo (jpg) distorted dimension)s of (100*100) , (see figure 3, figure 4 and figure 5 respectively), size 4.92kb and taking each pixel in image and processing its data with first order filters, the size of image after using the filter is 6.62kb, then comparing the results between the data of store that use first order filter and the image data that use first order filter, (see table 1).

Table 1. Comparing between tow uses of first order filter

Types of data	Values of a	Cut off frequencies	effective
Store data	Between o and 1	Greater than zero	good
Image data	Between 0 and 20	Similar	Very well

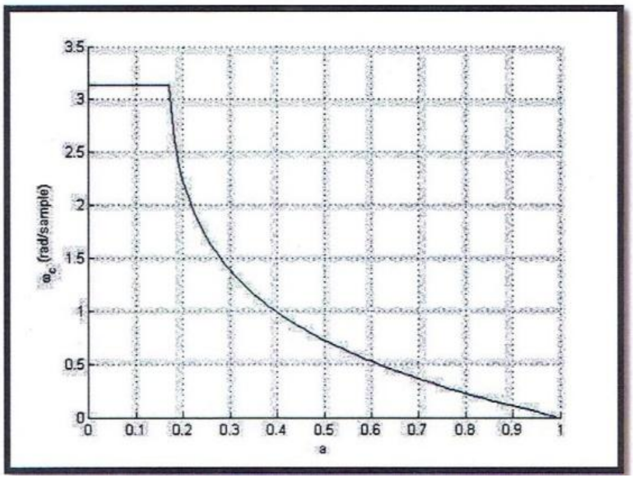


Figure 1: (Ω_c versus a in store data)

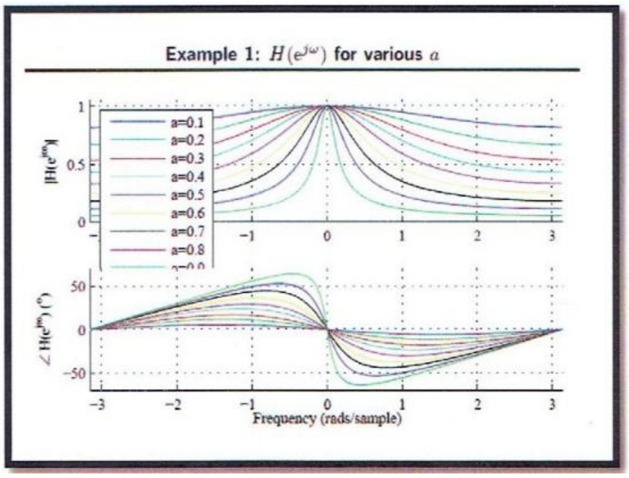


Figure 2: ($H(e^{j\omega})$ for various a in store data)

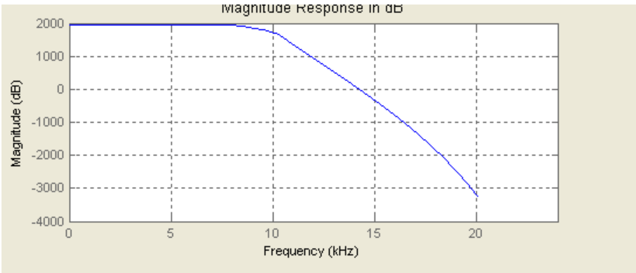


Figure 3: (Ω_c versus a in image data)

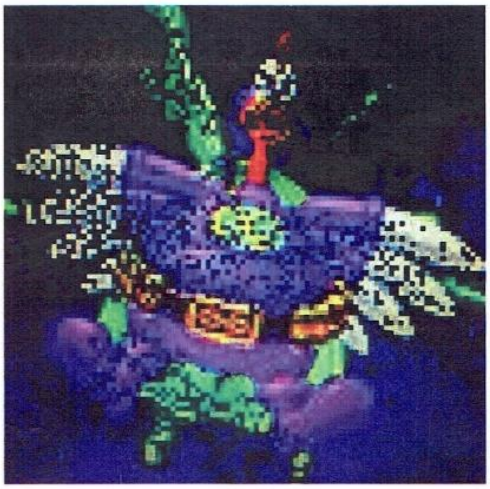


Figure 4: image before using First Order Filter

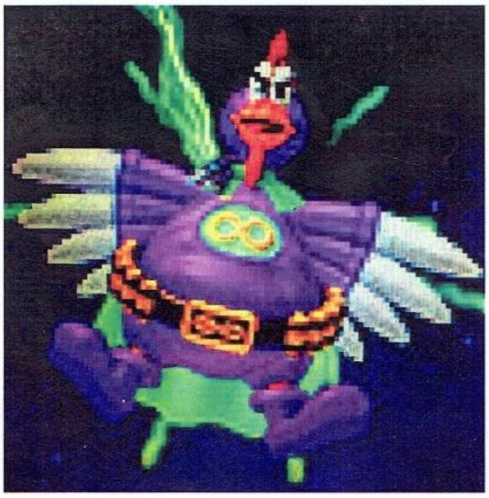


Figure 5: image after using First Order Filter

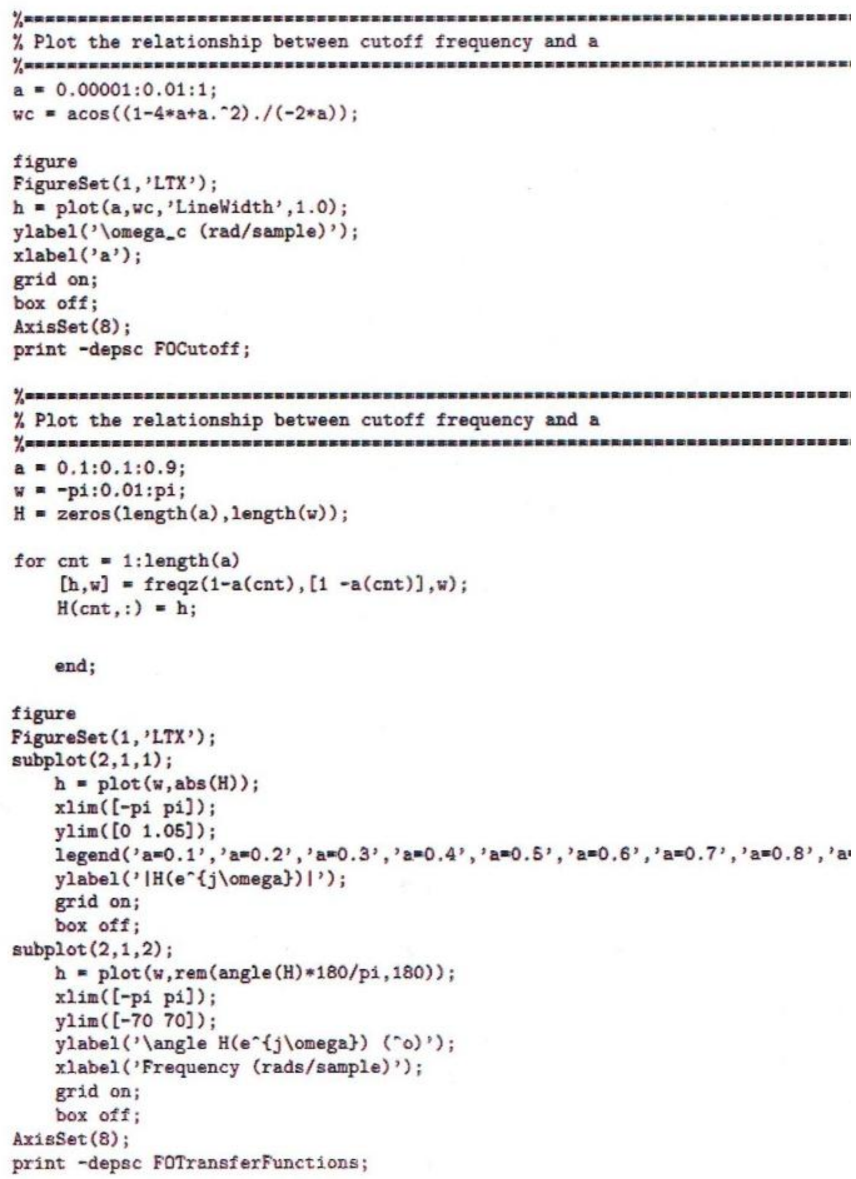
5. CONCLUSION

1. Using First Order Filter on digital images.
2. The possibility of improving digital images distorted.
3. The use of all kinds of data on the previous program (image –video-text).
4. Change the image size (increase in size) where the distorted image size is 4.92kb and the image size after undergoing the treatment process is 6.62kb.

6. FUTURE WORK

1. The possibility of applying later (Second Order Filter), (Third Order Filter).
2. The possibility of using moving average filter.
3. Application of research on all kinds of images (gray images, color images, small images, big images, and multi dimensions of images).
4. The foundation stone for the use of other different types of filters.
5. The possibility of introducing other random values to get the best results.

7. PART OF PROGRAM



```
%============================================================================
% Plot the relationship between cutoff frequency and a
%============================================================================
a = 0.00001:0.01:1;
wc = acos((1-4*a+a.^2)./(-2*a));

figure
FigureSet(1,'LTX');
h = plot(a,wc,'LineWidth',1.0);
ylabel('\omega_c (rad/sample)');
xlabel('a');
grid on;
box off;
AxisSet(8);
print -depsc FOCutoff;

%============================================================================
% Plot the relationship between cutoff frequency and a
%============================================================================
a = 0.1:0.1:0.9;
w = -pi:0.01:pi;
H = zeros(length(a),length(w));

for cnt = 1:length(a)
    [h,w] = freqz(1-a(cnt),[1 -a(cnt)],w);
    H(cnt,:) = h;

    end;

figure
FigureSet(1,'LTX');
subplot(2,1,1);
    h = plot(w,abs(H));
    xlim([-pi pi]);
    ylim([0 1.05]);
    legend('a=0.1','a=0.2','a=0.3','a=0.4','a=0.5','a=0.6','a=0.7','a=0.8','a
    ylabel('|H(e^{j\omega})|');
    grid on;
    box off;
subplot(2,1,2);
    h = plot(w,rem(angle(H)*180/pi,180));
    xlim([-pi pi]);
    ylim([-70 70]);
    ylabel('\angle H(e^{j\omega}) (^o)');
    xlabel('Frequency (rads/sample)');
    grid on;
    box off;
AxisSet(8);
print -depsc FOTransferFunctions;
```

8. REFERENCES

[1] Alrabity, A. M. 2011. Learning MATLAB step by step.

[2] http://mawdoo3.com/.

[3] University of Puerto Rico, Mayaguez Campus http://www.ece.uprm.edu.

[4] https://ar.m.wikipedia.org

[5] https://ar.wikipedia.org

[6] https://en.mwikipedia.org

[7] https://www.kutub.com

[8] www.alarabimag.com

[9] www.kutub.info