(U) A Statistical Differencing Filter for Image Enhancement

Walter Rose

US Army CCDC
Armaments Center

Picatinny Arsenal, NJ 07806

Walter.s.rose5.civ@mail.mil

(U) Abstract: The use of digital radiography in the context of industrial manufacturing has become widespread. In this paper, we describe the construction and implementation of a filter for image enhancement. We also describe how to extend the results in this paper to make the celebrated Wallis filter.

(U) Research Innovation and Objective(s): The constructed filter is demonstrated upon the famed Lena image along with some radiographs of army relevant items. Important metrics such as the entropy are calculated and discussed. Both quantitative and qualitative conclusions are drawn.

(U) Impacts on Warfighter Mission: Image processing will continue to play a vital role in modern industrial radiography. Understanding how these filters work, will allow for radiographers to make informed decisions about using these filters and designing their own custom filters for army applications.

(U) Keywords: Image processing, Wallis Filter, Statistical Difference Filter, Signal to Noise Ratio SNR, Entropy, Spatial Domain, Frequency Domain, Entropy, MATLAB®, Non-Destructive Testing, Radiography

1. (U) Introduction

(U) Radiographic testing is a non destructive testing technique which allows for the detection of subsurface flaws.

(U) Image processing is a vast field that encapsulates image enhancement, image restoration, encoding and compression. We will concern ourselves in this paper with a solution to the problem of image enhancement.

(U) Image enhancement can be solved using filtering. Filters can be used to blur or sharpen images. Filtering can occur in the spatial or frequency domain. In the spatial domain, filtering usually consists of convolving an image with a kernel. A kernel is a small matrix that is specifically designed for a specific function. For example, the simplest filtering operations are the low pass and high pass filters.

(U) For example, in a low pass filter the high frequencies are attenuated. High spatial frequencies correspond to a large variation in intensity values.

(U) Filtering in the frequency domain involves applying the fast fourier transform to the original image. In fourier space, one can simply multiply the transformed image, with an appropriate kernel. After multiplication one must apply the inverse fourier transform to return to the spatial domain to view the filtered image.

(U) There are many techniques to enhance an image via the spatial domain. The simplest and most widely used spatial domain filters belong to a class of transformations called, point-wise intensity transformations. Let F(m,n) represent the matrix of greyscale values of our original image. A point wise intensity transformation then takes the form,

$$G(m,n) = T[F(m,n)]$$

(U) Where T would be a prescribed function that depends on the greyscale intensity value at location (m,n) only. G(m,n) represents the new transformed intensity values. We refer the reader to the following reference for any number of such transformations.

(U) The statistical difference filter and its generalization, the Wallis filter is an example of a spatially adaptive filter. More on this in the following section. The remainder of the paper will be in describing how to build this filter and its effect on images.

2. (U) Method

2.1 (U) Theory

(U) First, some notation. Let F(m,n) represent the matrix of greyscale values of our original image. Let G(m,n) represent the new transformed intensity values.

(U) The mean of intensity of a submatrix is then given by the following expression,

$$M(j,k) = \frac{1}{W^2}\sum_{j-w}^{j+w}\sum_{n=k-w}^{k+w} F(m,n)$$

Similarly, the standard deviation of a submatrix is given by,

$$D(j,k) = \frac{1}{W}\sum_{j-w}^{j+w}\sum_{n=k-w}^{k+w} [F(m,n) - M(m,n)]^{\frac{1}{2}}$$

Then finally,

$$G(j,k) = \frac{F(j,k)}{D(j,k)} \qquad (1)$$

(U) The statisical differencing filter has the effect of "edge crispening". That is, pixels in the neighborhood that deviate significantly from their neighbors are amplified. Alternatively, pixels that do not deviate much from the average are not affected by this transformation.

(U) Now we can generalize the statistical differencing filter in a couple ways. Suppose we knew our image was degraded by some known distribution such as gaussian noise. Consider the following new transformation.

$$G(j,k) = \left(1 - \frac{\sigma_n^2}{\sigma_{loc}^2}\right)F(j,k) + \frac{\sigma_n^2}{\sigma_{loc}^2}M(j,k)$$

The noise variance is given by, σ_n^2. The local image variance is given by σ_{loc}^2 and is calculated via the equations above. In regions that include sharp edges we would expect that, $\sigma_{loc}^2 \gg \sigma_n^2$. This means the ratio, $\frac{\sigma_n^2}{\sigma_{loc}^2} \to 0$. This implies that $G(j,k) = F(m,n)$, the transformed intensities are equivalent to the original intensities. Now consider a neighborhood that is relatively, flat. Mathematically this means, $\sigma_{loc}^2 < \sigma_n^2$. This means the ratio, $\frac{\sigma_n^2}{\sigma_{loc}^2} \to 1$. This means that $G(j,k) = M(j,k)$, these pixels will be transformed to the mean.

(U) Now we can consider a final variation of this. The Wallis operator is defined by the following expression.

$$G(j,k) = (F(j,k) - M(j,k))A(j,k) + B(j,k)$$

(U) Here, A(j,k) is a spatially dependent gain factor and B(j,k) is a spatially dependent backround factor. [2] There are numerous closed form expressions one can use for the terms A and B above.

The spatially dependent gain factor, A(j,k), can be calculated using the following formula.

$$A(j,k) = \frac{A_{max}D_d}{A_{max}D(j,k) + D_d}$$

(U) Here D_d represnts the desired standard deviation factor, and A_{max} represents the maximum gain factor to prevent large output values when the neighborhood standard deviation, D(j,k) is small. [2]

The spatially dependent backround factor, B(j,k), can be calculated using the following formula.

$$B(j,k) = pM_d + (1-p)M(j,k)$$

(U) Here M_d represnts the desired standard deviation factor, and p, is a mean proportionality factor controlling the background flatness of the enhanced image. [2]

One can find other expressions for the spatially dependent factors in the literature.

2.2 (U) Algorithm

(U) We now detail the construction of the statistical difference filter. Implentation was accomplished using MATLAB® The complete code is attached to the appendix at the end of the report.

(U) There are three basic inputs required by the user. First, the image can be of any file type. MATLAB® allows for great versatility in this regard. The next input would be the bit depth of the image. Lastly, the window counter, w, is required as an input. The window size, W, is defined by the simple equation,

$$W = 2w + 1$$

(U) The first task is defining the submatrix window which defines the local neighborhood over which statistics will be gathered. Consider the following matrix below.

$$\begin{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{m1} & \cdots & f_{mn} \end{pmatrix}$$

The submatrix depicts the first 3 by 3 neighborhood within the general matrix of greyvalue intensities, F(m,n).

(U) After defining the neighborhood, one must ensure that the dimensions of the original image are compatible with the chosen neighborhood size. I chose to use the MATLAB "try-catch" function. The code attempts (tries) to form a 3x3 neighborhood around a given pixel. If successful, the program then goes on to calculate the mean and standard deviation. If a neighborhood cannot be defined around a given pixel, (catch) then this greyscale value remains unchanged. It is important to note that pixels that remained unchanged due to an ill defined neighborhood, are deleted in the final filtered image. For example, referring to the matrix displayed above, one cannot define a neighborhood submatrix of size 3x3, centered around greyscale intensity f_{11}.

(U) We then are able to "scan" the entire image, analyzing each neighborhood around every greyscale intensity in a systematic manner. Our program scans each row sequentially. The program calculates the mean and the standard deviation of each neighborhood. The program stores the new transformed intensity values in a matrix, G according to equation 1.

(U) Finally the program displays both the original and transformed image with the acompanying histograms.

2.3 (U) Evaluation Metrics

(U) The problem of image enhancement is inherently qualitative in nature. I sought out a metric to describe the statistical difference filter in a more quantitative way.

(U) First we considered the notion of entropy. Mathematically entropy is defined as,

$$H = -\sum p_i \, log p_i$$

(U) This expression, is the information theoretic definition as defined by Shannon. Physicists also use the notion of entropy in the context of thermodynamics. Roughly speaking, the broader the distribution the higher the entropy. In section 3, results and discussion, we plot entropy as a function of the submatrix window size.

3. (U) Results and Discussion

(U) In this section we discuss the results of the statisical difference filter on various synthetic and real images.

3.1 (U) Lena

(U) The original Lena picture is 8 bits, .gif filetype, and is of size 256x256 pixels. The original image along is displayed in figure 1 below. In figure 2 below we display the histogram of the original Lena image before we apply statistical differencing.

Figure 1. Original greyscale Lena Image [2]

(U) For comparison purposes we display the histogram of greyscale intensities in Figure 2. We calculate the entropy of the original image to be approximately 7.57.

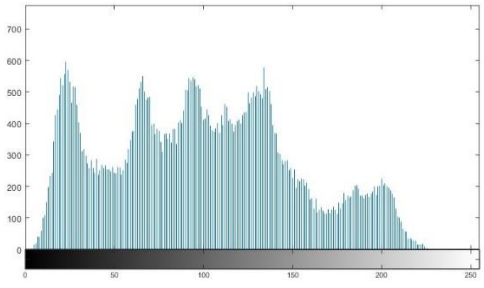

Figure 2. Histogram of Intensities of original Lena Image

(U) Small window sizes shift the histogram to lower intensity values, while bigger window sizes shift the hisogram to higher intensity values. Figure 3 shows the histogram of the processed Lena image with a window size of 5x5.

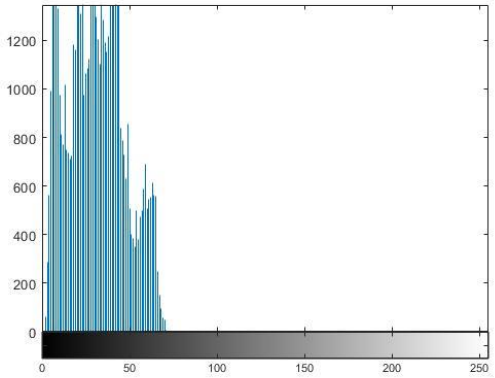

Figure 3. Histogram of processed Lena Image with a window size of 5. Entropy is 5.89.

Now consider a larger window size. When the window size is 29x29, one sees a shift towards higher intensity values. Figure 4 displays the processed lena image with a window size of 29x29. No window levelling has been performed.

Figure 4. Processed Lena Image with a window size of 29x29. Entropy: 6.20.

(U) We apply the statistical differencing filter with varying window sizes and plot the entropy in Figure 5.

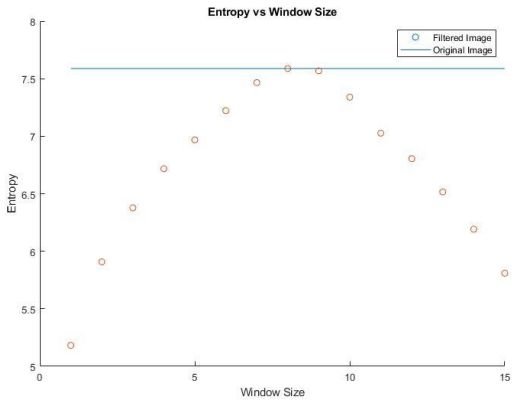

Figure 5. Entropy vs. Submatrix window size for a image of Lena, greyscale.

(U) The first observation is that the graph of the entropy versus window size is concave down. Therefore, there is a particular window size which maximizes the entropy. When choosing a window size of 8, the entropy of the original image matches the entropy of the filtered image. Note, this does not mean the images are equivalent.

3.3 (U) Camera man

(U) The original camera man picture is 8 bits, .gif filetype, and is of size 512x512 pixels. The original image along is displayed in figure 6 below. In figure 7 we display the statistical differencing filter over varying window sizes and graph the calculated entropy vs. window size.

Figure 6. Original greyscale Camera man Image. Entropy is 7.23 [2]

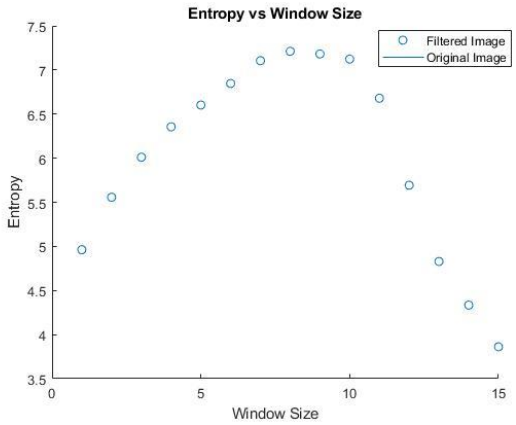

Figure 7. Results of Statistical Differencing, Entropy versus Window Size for the Camera man Image

One observes the same concave down behavior as the Lena image.

3.4 (U) Building

(U) The original building picture is 8 bits, .jpg filetype, and is of size 359x479 pixels. The original image along is displayed in figure 8 below. In figure 9 we display the statistical differencing filter over varying window sizes and graph the calculated entropy vs. window size.

Figure 8. Original greyscale building Image. Entropy is 6.9451 [2]

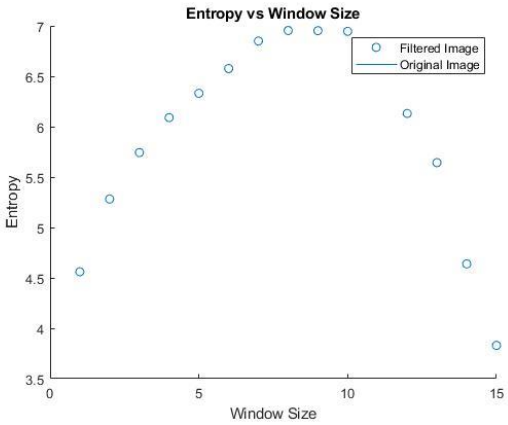

Figure 9. Results of Statistical Differencing, Entropy versus Window Size for the building image.

One observes the same concave down behavior as the Lena image.

3.5 (U) Flower

(U) The original flower picture is 8 bits, .jpg filetype, and is of size 240x320 pixels. The original image along is displayed in figure 10 below. In figure 11 we display the statistical differencing filter over varying window sizes and graph the calculated entropy vs. window size.

Figure 10. Original greyscale flower Image. Entropy is 7.47 [2]

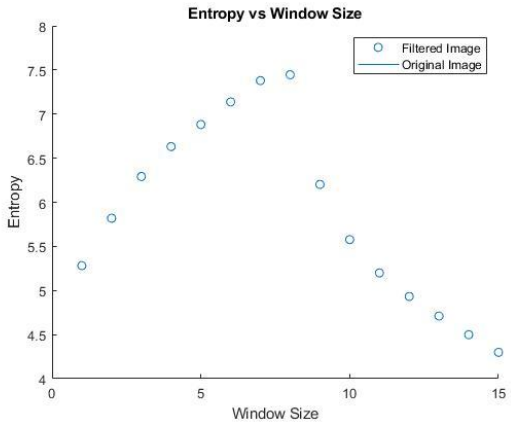

Figure 11. Results of Statistical Differencing, Entropy versus Window Size for the flower image.

One observes the same concave down behavior as we have been exhibiting thus far.

3.6 (U) Miscellaneous Army Items

(U) Figures 12 are the results of the statisical difference filter using a radiograph of an M1 shell as a template. The original radiograph is 16 bits greyscale, .dcm type, and is of size 1025x725 pixels.

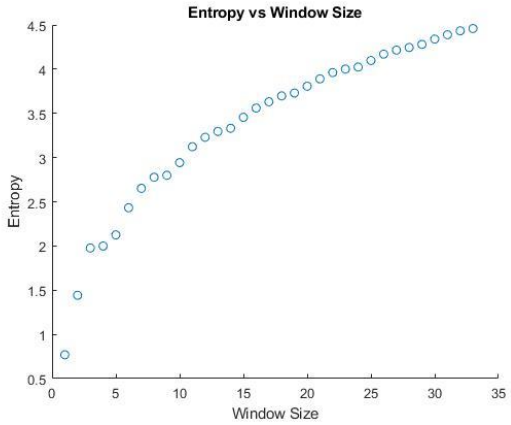

Figure 12. Entropy vs. Submatrix window size for a image of an M1 shell.

Figure 12 shows an increasing function at a decreasing rate. If program was run for larger window sizes, I believe we would get the same concave down behavior we have been presenting earlier. This required considerable computational time and so that is why I chose not to run at larger window sizes. It is important to note that there are ways to make this filter computationally more efficient. This was not the objective of this report and can be explored in future work.

4. (U) Conclusion

(U) In this technical report we considered the statistical difference filter and a couple of its variants from a theoretical point of view. We implemented the statistical difference filter in MATLAB and discussed a quantitative metric associated with the processing of such an image. Traditionally entropy is used in the context of image compression, but we see that entropy can be a useful macroscopic variable when performing statistical differencing. The entropy is a way of summarizing the histogram in a single number.

(U) An understanding of how these algorithms are constructed will be increasingly important in the future. Being able to work with contractors and government facilities on image processing tasks will continue to be pervasive throughout industrial radiography. It is important to note that image processing, and more generally signal processing has applications outside of industrial radiography that are relevant to the department of defense.

(U) While off the shelf image processing tools are available, research applications often call for new innovative approaches to be taken.

(U) The code for the Wallis filter is also included in the appendix. Free variable input is at the discretion of the user.

5. (U) Future Work

(U) It might be instructive to gather entropy statistics based on the variety of free parameters that are defined for the Wallis operator. One might also want to construct different geometrical shapes (as opposed to a square) for gathering local neighborhood statistics.

(U) As mentioned, this is only one of many different types of filters one can implement for image enhancement. Each filter has its merits and drawbacks. Frequency domain techniques as well as other spatial domain techniques are widely used and will be considered in future technical reports.

6. (U) Appendix

6.1 (U) Statistical Difference Filter

Shown below is the MATLAB® code for the statistical difference filter.

```
% Inputs: X, w, b

X = dicomread('MC00AL7B_2.dcm');
%Read in Image


X_uint = uint16(X);

max_w = 50; %Max window Counter

entropy_f = zeros(1,max_w);

for w = 1:max_w

W = 2*w + 1; %Odd Numbers only

[m,n] = size(X_uint);

G = zeros(m,n);


for i = 1:m

    for j = 1:n

        try

        N = X(i-w:i+w,j-w:j+w);

        S = 0;

        SS = 0;

%Calculate Mean of Neighborhood
             for i_n = 1:W

                for j_n = 1:W

                    S = S +
N(i_n,j_n);

                end

             end

             M = 1/W^2 * S;

 %Calculate Standard Deviation of
Neighborhood
             for i_n = 1:W

                  for j_n = 1:W

                      SS = SS +
(N(i_n,j_n) - M)^2;

                  end

             end

             if SS == 0

                 SS = 1;

             end

             SS = double(SS);

             SS = 1/W*sqrt(SS);
%Estimated Standard Deviation

             G(i,j) = X_uint(i,j)/SS;

         catch

         G(i,j) = X_uint(i,j);

         end

    end

end

G_2 = G(max_w+1:m-max_w,max_w+1:n-
max_w);

G_2 = uint16(G_2);

X_2 = X(max_w+1:m-max_w,max_w+1:n-
max_w);

entropy_X = entropy(X_2);

entropy_G = entropy(G_2);

entropy_f(w) = entropy_G;

end
```

```
x = [1:1:max_w];
y = entropy_X;
scatter(x,entropy_f)
hold on
line(x,y)
title('Entropy vs Window Size')
xlabel('Window Size')
ylabel('Entropy')
legend('Filtered Image','Original
Image')
```

6.1 (U) Wallis Filter

Shown below is the MATLAB® code for the Wallis filter.

```
cd 'C:\Users\'

X = imread(.gif)

X_uint = uint8(X);

p = 0.2; %Mean Proportionality
Factor

D_d = .5; %Standard Deviation
Factor

M_d = 1.2; %Average Mean Factor

A_max = 2; %Maximum Gain Factor

%Window; Square Neighborhood

w = 15;

W = 2*w + 1; %Odd Numbers only

[m,n] = size(X_uint);

G = zeros(m,n);


for i = 1:m

    for j = 1:n

        try

        N = X(i-w:i+w,j-w:j+w);

        S = 0;

        SS = 0;

%Calculate Mean of Neighborhood
            for i_n = 1:W

                for j_n = 1:W

                    S = S +
N(i_n,j_n);

                end

            end

            M = 1/W^2 * S;

 %Calculate Standard Deviation of
Neighborhood
           for i_n = 1:W

                for j_n = 1:W

                    SS = SS +
(N(i_n,j_n) - M)^2;

                end

           end

           if SS == 0

               SS = 1;

           end

           SS = double(SS);

           SS = 1/W*sqrt(SS);
%Estimated Standard Deviation

           G(i,j) = (X_uint(i,j)-
M)*(A_max*D_d)/(A_max*SS+D_d) +
(p*M_d + (1-p)*M);

        catch

        G(i,j) = 0;

        end
```

```
    end

end

G_2 = G(w+1:m-w,w+1:n-w);

X = uint8(X);

G_2 = uint8(G_2);

entropy_X = entropy(X);

entropy_G_2 = entropy(G_2);

figure(1)
imshow(X)
imcontrast

figure(2)
imshow(G_2)
imcontrast
```

References

1. Pratt, William. Introduction to Digital Image Processing, CRC Press, September 2013

2. Katsaggelos, Aggelos. Fundamentals of Digital Image and Video Processing, Northwestern University Course Notes, (Available through Coursera)