International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.2, Mar-Apr 2012 pp-338-340 ISSN: 2249-6645

Quality improvement of image using adaptive bilateral filter and neural networks

T.Ravi[1], Ch.Mounika[2], Ch. Rajesh Babu[3], T.Prasanth[4]
(Department of Electronics and Communication Engineering, KL University, India)

ABSTRACT

In this paper we present a adaptive bilateral Filter (ABF) with neural networks for enhancing sharpness and suppressing noise. Increasing the slope at edges ABF sharps the image and doesn't contain the halo. Undershoot and overshoot are not produced. Significantly sharper images are restored using ABF when compared with bilateral filter, un sharp mask(USM). It is better than other filters in noise removal also. The back propagation neural network we apply makes more efficient in sharpening the image.

***Keywords*– back propagation neural network, bilateral filter, image restoration, sharpness enhancement**

I. INTRODUCTION

The problem here is to develop a filter that removes noise and also sharpens the edges simultaneously. So we have to design a filter first of all which uses a better method for sharpening without halo and secondly it should be able to remove noise effectively.

We can employ conventional filters for noise removal, which work efficiently in smooth regions but blurring of the image takes place especially at the edges. A lot of effort is put in designing a noise removal by preserving the edges. These efforts were resulted in "SUSAN" filter and "bilateral filter". Bilateral filter adopts low pass Gaussian filter for both domain and range filter. The domain low pass Gaussian filter gives higher weight to pixels that are spatially close to center pixel. The range low pass Gaussian filter gives higher weight to pixels that are similar to center pixel in gray value. By combining domain filter and range filter we can produce bilateral filter which will reduce noise and enhances the sharpness. Gaussian filter that is oriented along the edge will do the averaging along the edges and reduce in gradient direction. For this reason bilateral filter can smooth the noise and preserve the edge details.

In terms of image sharpening un sharp mask (USM) has certain disadvantages: First It sharpens the image by adding halo(undershoot and overshoot). Second it amplifies the noise information present in the image instead of suppressing the noise and reduce the quality of image. To reduce the first problem we have several slope restoration algorithms. Those algorithms modify the edge information normally or horizontally or vertically i.e in 1D only. The ABF will restore the edge slope, without need to locate edge normal's. So ABF with neural networks is efficient to implement. This will produce clean, crisp edges. To reduce the noise levels in an image we use Gaussian low pass filter which will remove the noise.

II. GAUSSIAN FILTER

The impulse response of a Gaussian filter is Gaussian function. Gaussian filter is designed to give no overshoot or undershoot to input image while minimizing the rise and fall time. The output of Gaussian filter is the convolution of input signal with a Gaussian function.

$$G = \exp(-(X^2+Y^2)/(2*\sigma^2))$$

where X is the distance from the origin in the horizontal axis, Y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. In case of 2D, the above formula gives a surface whose outputs are concentric circles with a Gaussian distribution from the center point. The original pixel's value assigns heaviest weight to center pixel (having the highest Gaussian value) and smaller weights to neighboring pixels as their distance from the original pixel increases. The main advantage of Gaussian function is it will be non-zero at every point on the image, it means that the entire image should be included in the calculation of each pixel.

III.ADAPTIVE BILATERAL FILTER

The impulse response of ABF is

$$f[m, n] = \sum_{k}\sum_{l} h[m, n; k, l]g[k, l]$$

f[m,n] is restored image, h[m,n; k, l] is the response at [m, n] to an impulse at [k, l], and g[m, n] is the degraded image.

$$h[m,n;m_0,n_0] = I(\Omega_{m_0,n_0})r^{-1}_{m_0,n_0} e^{-\left(\frac{(m-m_0)^2+(n-n_0)^2}{2\sigma_d{}^2}\right)} \cdot e^{-\frac{1}{2}\left(\frac{g[m,n]-g[m_0,n_0]-\zeta[m_0,n_0]}{\sigma_r[m_0,n_0]}\right)^2}, \quad (2)$$

where [m0, n0] is the center pixel of the window, Ωm0,n0 ={[m, n] : [m, n] ∈ [m0 − N,m0 + N]×[n0 − N, n0 + N]},I(･) denotes the indicator function, and m0,n0 normalizes the volume under the filter to unity.

The main two modifications in ABF compared to bilateral filter is

1. The offset ζ is introduced in range filter
2. Both offset ζ and σr are locally adaptive.

If ζ is equal to zero and by making σr constant we get bilateral filter.

For domain filter the value of σd= 1.0 will make the filter adaptive.

By varying the values of ζ and σr we get a powerful filter which will produce both smoothing and sharpening to the given noisy image. It sharpens the image by increasing the slope of the edges. To understand the ABF working completely we should know the role of ζ and σr.

The range filter can be assumed as one dimensional filter which will process the histogram of image. For conventional bilateral filter, the range filter is located on the histogram at the gray value of the current pixel and removes the pixel values which are farther away from the center pixel value.

By adding the offset ζ to the range filter, we can shift the range filter on the histogram. As before, let Ωm0,n0 denote the set of pixels in the (2N + 1) × (2N + 1) window of pixels centered at [m0, n0]. Let MIN, MAX, and MEAN denote the operations of taking the minimum, maximum, and average
value of the data in Ωm0,n0 , respectively.
Let Δm0,n0 =g[m0, n0] − MEAN(Ωm0,n0).
We can define the ABF in terms of Gaussian filter as
1.No offset (conventional bilateral filter):
ζ[m0, n0] = 0,
2.Shifting towards the MEAN:ζ[m0,n0] = −Δm0,n0
3.Shifting away from the MEAN, to the MIN/MAX:
ζ[m0, n0] =

MAX(Ωm0,n0) − g[m0, n0], if Δm0,n0 > 0,
MIN(Ωm0,n0) − g[m0, n0], if Δm0,n0 < 0,
0, if Δm0,n0 = 0. – (3)

By shifting the range filter towards MEAN(Ωm0,n0) will blur the image. By Shifting the range filter away from MEAN(Ωm0,n0) will sharpen the image. In the case of operation No. 3, the range filter is shifted to the MAX or the MIN depending on Δm0,n0 . The reason behind these observations is the transformation of the histogram of the input image by the range filter. The conventional bilateral filter without offset does not significantly alter the histogram or edge slope. Shifting the range filter to MEAN(Ωm0,n0) at each pixel will redistribute the pixels towards the center of the histogram. On the other hand, if we shift the range filter further away from the MEAN(Ωm0,n0), pixels will be compressed and the slope will be increased, We need to point out here that shifting the range filter based on Δm0,n0 is very sensitive to noise.

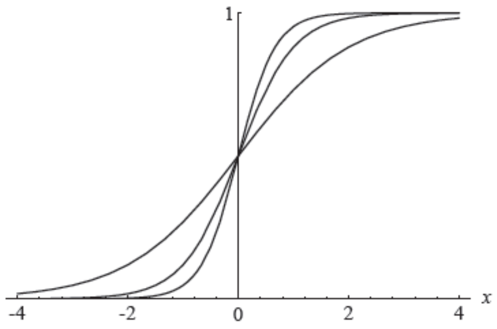
The parameter σr of the range filter controls the width of the range filter. If σr is large compared to the range of the data in the window, the range filter will assign similar weights to every pixel in the range. Therefore, it does not have much
effect on the overall bilateral filter. On the other hand, a small σr will make the range filter dominate the bilateral filter. By making ζ and σr adaptive and jointly optimizing both parameters, we transform the bilateral filter into a much more powerful and versatile filter. To smooth a pixel, we can shift the range filter towards MEAN(Ωm0,n0), and/or use a large σr which enables the spatial Gaussian filter to take charge in bilateral filtering. To sharpen a pixel, we can shift the range filter away from MEAN(Ωm0,n0) towards MAX(Ωm0,n0) or MIN(Ωm0,n0), depending on whether it is above or below the midpoint of the edge slope. At the same time, we reduce σr accordingly. With a small σr, the range filter dominates the bilateral filter and effectively pulls up or pushes down the pixels on the edge slope.

IV.BACK PROPAGATION ALGORITHM

The main advantage of multilayered networks is they are capable of computing a wide range of functions than networks with a single layer networks. The number of iterations increases as the no. of parameters increases. But advantage of Back propagation algorithm is it will reduce the no. of iterations to be computed. Back propagation algorithm will minimize the error by using gradient descent. So in this method we calculate gradient of each step. The activation function used here is sigmoid.

$$s_c(x) = \frac{1}{1 + e^{-cx}}.$$

Here c is a constant. By varying the value of c we can get different curves



The above curves are for values c=1,c=2,c=3.
The sigmoid values vary from 0 to1 strictly. The output expression is given by

$$\frac{1}{1 + e^{\sum wixi - \theta}}$$

Where W_i =weights i=1,2,3……..,n
X_i = Inputs
$-\theta$ = bias

The algorithm in back propagation is divided into four steps
1. Feed forward computation
2. Back propagation to output layer
3. Back propagation to hidden layer
4. Weight updates

The iterations are stopped when the error function value became small. Back propagation is the Most common method of obtaining the many weights in the network. In this we use supervised training
This algorithm depends on minimizing the error of the network using the derivatives of the error function whose properties are
► Simple
► Slow
► Prone to local minima issues
In back propagation we use mean square error
E = (target – output)2
Where E is the error.
Partial derivatives of the error with respect to the weights:
Output Neurons:
let: $\delta j = f'(net_j)\ (target_j - output_j)$
$\partial E/\partial wji = -outputi\ \delta j$

j = output neuron
i = neuron in last hidden
Hidden Neurons:

let: $\delta j = f'(netj)\ \Sigma(\delta kwkj)$

$\partial E/\partial wji = -outputi\ \delta j$

j = hidden neuron
i = neuron in previous layer
k = neuron in next layer

Here the calculation of derivatives move backwards so it is called back propagation. In the direction of derivatives we have maximum increase in error. Where as the maximum decrease in error results in back ward direction. The maximum decrease in error function is given as

$$wnew = wold - \alpha\ \partial E/\partial wold$$

where α is the learning rate.

The learning rate α plays an important role in the convergence. If learning rate is too small then it converge extreme slowly. If it is too large it may not converge.

Momentum

It try to converge the network. The main advantage is it applies smoothed averaging to change in weights.

$$\Delta new = \beta\Delta old - \alpha\ \partial E/\partial wold$$

$$wnew = wold + \Delta new$$

β is the momentum coefficient

It acts as a low-pass filter and reduce the rapid fluctuations

V. RESULTS

Fig: Original image

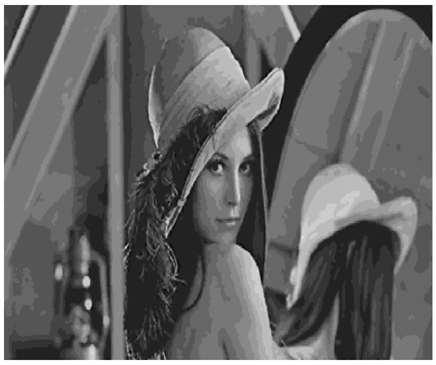


Fig: After applying Adaptive Bilateral Filter

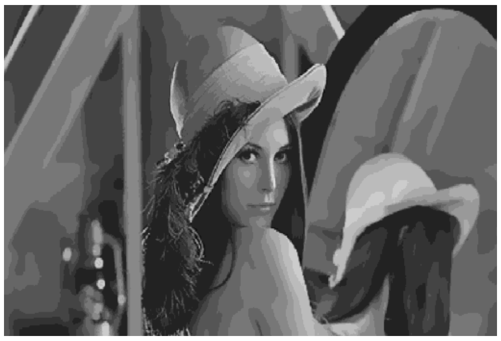


Fig: After applying Back Propagation network.