AIA
Bilateral Filtering

Emre Ozan Alkan

{emreozanalkan@gmail.com}

MSCV-5

January 11, 2015

1   Introduction

In the field of Image Processing filtering especially for the use of noise reduction is used to remove unwanted noise, i.e., removing the noise of image sensors to further improve the analysis and processing of the image. Denoising of image traditionally employees a Gaussian filtering technique. This smooths the image over a local region thus removing any noise present. Smaller structures present in the image are completely lost but structures that are more coarser would still be present. We can see formulate the Gaussian filtering as convolution between the image and a Gaussian Kernel as shown below.

$$I_{output} = I_{Input} * h(x, y)$$

where $h(x, y)$ is the Gaussian kernel defined as

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

where σ gives the variance of the Gaussian that determines the degree of blurring or smoothing.

Below we can see a discrete example of Gaussian filtering of an image where the values of the image is just a mean of images around it that is processed in x and y direction

separately.

$$\frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix} * \frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\2 & 4 & 2\\1 & 2 & 1\end{bmatrix}$$

We can clearly see that these traditional techniques do not care about the edges and this information is degraded when applying such methods.

2   Bilateral Filtering

Bilateral filtering is a non linear filtering that preserves strong edges. This primarily work for bilateral filtering was done by Aurich et al 1995 who worked on nonlinear Gaussian filters. This was later refined and was called as Bilateral filter by Tomasi et al. Unlike traditional filtering process where the filtering is done by replacing the pixel value by average values of pixel around it, it also accounts for the pixel intensities around it while performing filtering. This produces a filtering process that uses the domain and range data of an image to produce images where the edge information is retained.

The filter can be dined as below

$$I^{\text{filtered}}(x) = \frac{1}{W_p}\sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

where the weight W_p is defined as

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

Here $I^{Filtered}$ is output image that is filtered. I is the input image and x is the location of the current pixel. ω is the window centered at x where the operation takes place. f_r and g_s are defined as the range and the spatial kernel receptively to smooth the image according to its intensity and coordinate. A visual overview of the process can beem seen below where all the parameters have been visualized.

This approach has a lot of application such as Denoising, Contrast Management, Depth Reconstruction, Data Fusion and demosaicking.

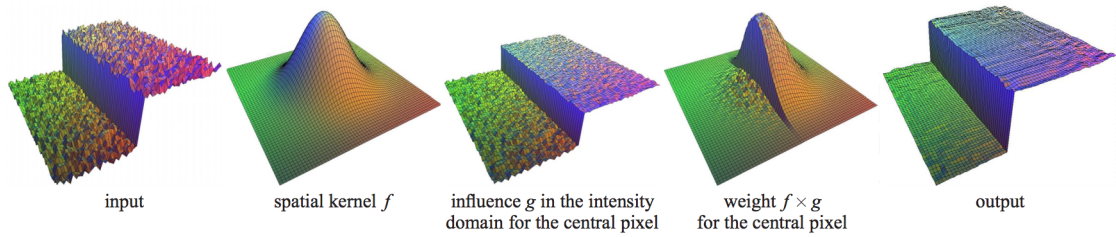The salient feature of this method is that its formulation is very simple as each of the pixel

Figure 1: The complete process visualized

values are a weighted mean of the negligible and this can be tuned with just 2 parameters that tell us the spatial size and the range to get the desired output. With the advent for new algorithm it can be used also in a non iterative manner and at a faster speed making it even real time.

3 Matlab Implementation

In this project I used the implementation of the famous paper - "Bilateral Filtering for Gray and Color Images" by C. Tomasi et al. In this report you find the study of the paper, the code and analysis. Implementation includes bilateral filter for both color and gray images, image abstraction - cartoonizing the image and demo scripts to prepare this report.

4 Application and Results

In this section we discuss the 2 major application of such filtering techniques that we have developed, They are denoising and image abstraction. The denoising was done for both color and gray-scale image and image abstraction was done for color only. First we test the algorithm with synthetic image of a check-board where the noise was added.

Here we can see that the denoising for gray-level image is very well performed. Most of the major edges are preserved while removing all the noise that was added. It is similar for color image with the addition that there is no phantom colors introduced during the step. There is definitely a loss in texture but that is expected from any denosing application in Image processing.

Image abstraction was done using the idea provided by Winnemoller et al. This can be
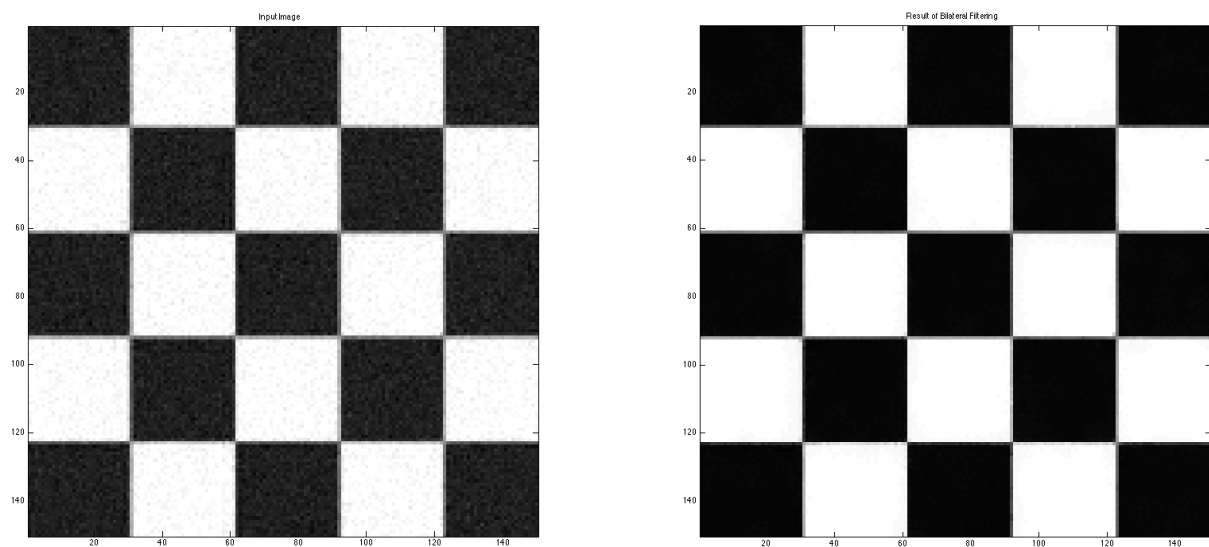
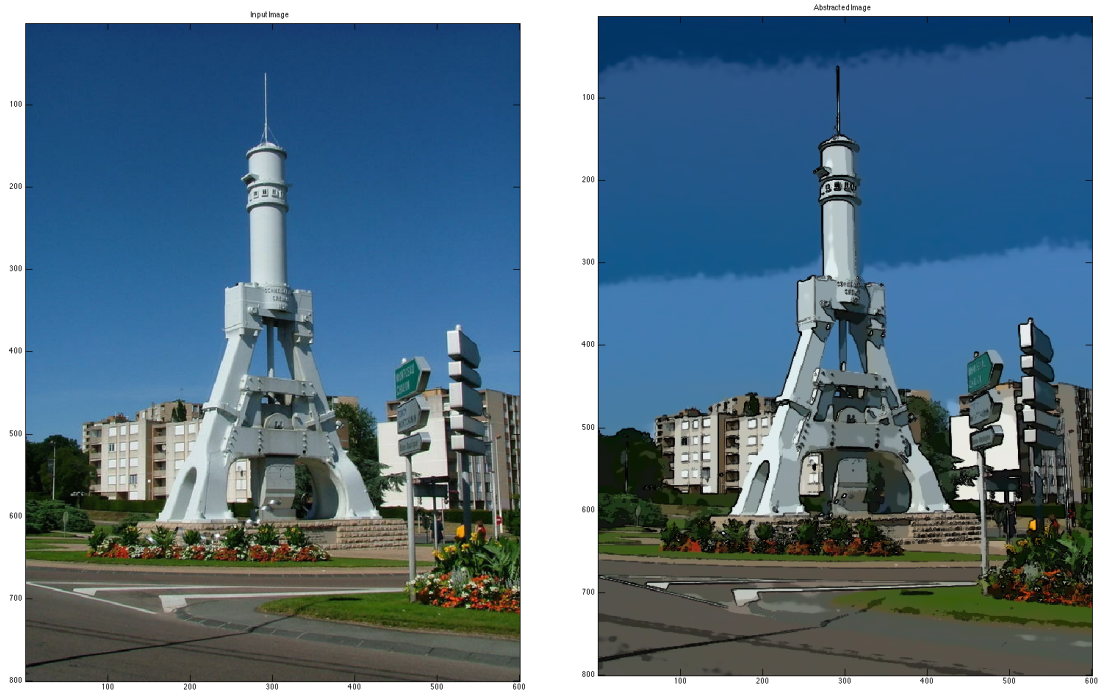Figure 2: Denoising for synthetic Image

Figure 3: Denoising for gray-scale Image

viewed as another application of bilateral filtering.

Figure 4: Denoising for color Image

(a) Original Image

(b) Abstracted Image

Figure 5: Image Abstraction

Here below we can see the parameter changes effecting on the cameraman image. There are two parameters we experimented: w - window size and σ - standard deviation.

(a) w = 5, σ = 2

(b) w = 5, σ = 5

(c) w = 7, σ = 2

(d) w = 7, σ = 5

(e) w = 9, σ = 2

(f) w = 9, σ = 5

Figure 6: Cameraman Image

5   Conclusion

We conclude that bilateral filtering is very simple and elegant algorithm image denoising that preserves edges. This was achieved by the addition of intensity range term to classical image filtering algorithm produce the desired result. The range and the spatial coefficients

could be varied accordingly to use for any specific application. In future work we could focus on making this algorithm more efficient as currently it take $O(n^2)$ time to filter an image.