Advanced Image Analysis:
Bilateral and Guided Image Filtering

Sandeep Manandhar
University of Burgundy

2015/22/12

1 Introduction

Edge preserving filtering has been an important topic in computer vision and image processing. Its notion arises in various application like denoising, smoothing, compression, image abstraction, optical flow, etc. Techniques like weighted least squares, anisotropic filtering, bilateral filtering, etc. have been developed over the years.
This report presents matlab implementation of bilateral filtering [5] and guided image filtering [4] and a brief comparative study of both.

2 Gaussian Smoothing

Before We talk about edge preserving filters, we want to discuss about the simple gaussian smoothing. Guassian filtering is a weighted average of intensities in the neighbourhood at a given pixel location. The weighing of intensities is related to the distance from anchoring pixel and not the image content itself.

$$I_p = \sum_{q \in S} G_\sigma(|p - q|) I_q \tag{1}$$

where, I is the image intensity, p is the target pixel location, q is a location in neighbourhood S of p and G_{sigma} is 2D Gaussian kernel mathematically written as:

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-x^2}{2\sigma^2} \tag{2}$$

where x is pixel location and σ is the standard deviation. As we can notice in equation 2, the kernel averages the pixel intensities without considering the adjacent intensities. Hence, it disregards the presence of edges and discontinuities.

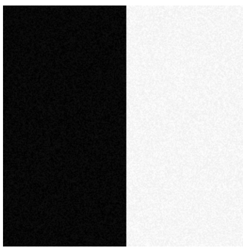

fig. 1: noisy image with edge

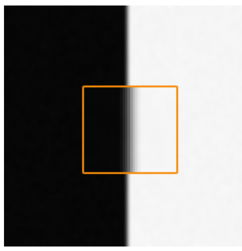

fig. 2: smoothed with gaussian kernel of size 5×5

The yellow bounding box in the figure 2, shows that the gaussian smoothing has deteriorated the edge.

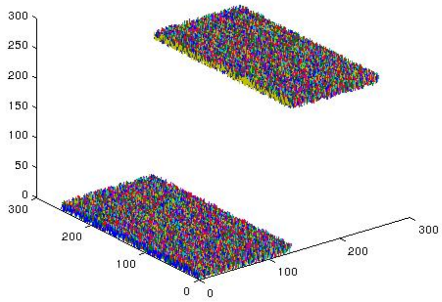

fig. 3: before smoothing: spatial vs intensity

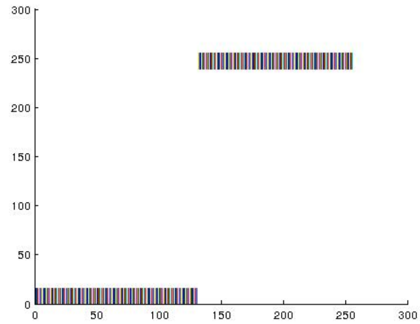

fig. 4: X-Y view: discontinuity in the edge

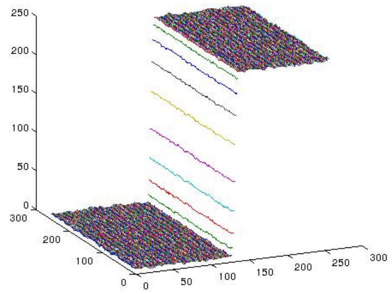

fig. 5: after smoothing: spatial vs intensity

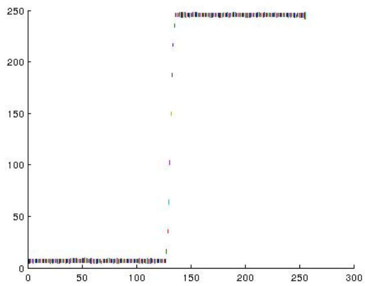

fig. 6: X-Y view: blurring in edge shifts the edges

Comparing the figure 5 and 6, we find that the edge has been effected by the blurring. The edge after blurring shows less discontinuity.

3 Bilateral filtering

Now that we have seen a drawback of simple blurring, we arrive to the necessity of edge preserving filtering. Bilateral filtering is one of such filters that serve our purpose.
In bilateral filtering, we smooth with gaussian function but the key notion here is that the influence of the neighbour should be based on both distance and intensity values. Mathematically, it would appear as:

$$I_p = \frac{1}{W_p} \sum_{q]inS} \underbrace{G_{\sigma_s}(|p-q|)}_{\text{spatial term}} \underbrace{G_{\sigma_r}(|I_p - I_q|)}_{\text{range term}} I_q \tag{3}$$

where W_p is the normalizing factor to ensure sum of weights to be equal to 1. σ_s and σ_r are the standard deviation in spatial and intensity range domain, and are vital for smoothing results.

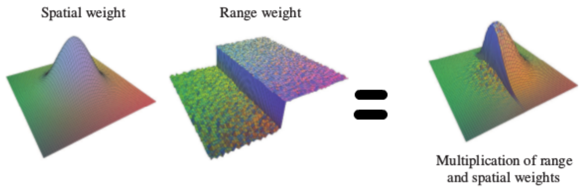

fig. 7: bilateral filter kernel(from: Bilateral Filtering: Theory and Applications, S. Paris[1]

The figure 7 shows the formation of bilateral kernel that has no response on the edges but only on supposedly homogeneous regions. It is to be noted that, the product of the spatial gaussian response, G_{σ_s} and range response, G_{σ_r} gives us the bilateral response. So, the final response is also a gaussian response and the final standard deviation is proportional to the product of σ_r and σ_s [2]. So, one should be cautious about picking near zero values for any of these parameters, otherwise no smoothing would be apparent.

3.1 Results

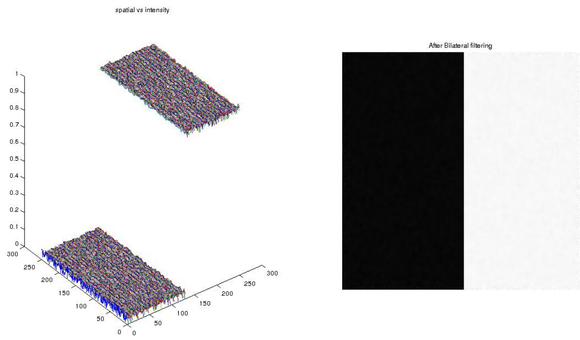

fig. 8: *left*: Spatial vs Intensity; Discontinuity near edge is dominant, *right*: The filtered Image, no blurring of edge

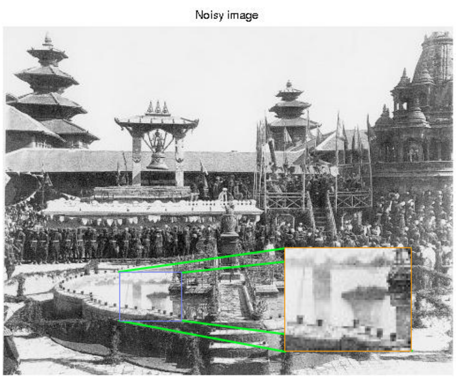

fig. 9: upclose look: noise on the pond and images on it

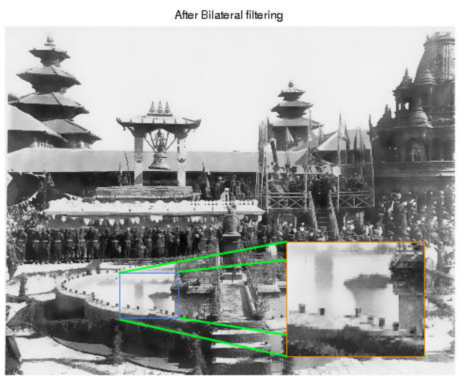

fig. 10: upclose look: smoothed water on pond with preserved edges,

From the results in figures 13, 16 and 19, we conclude that, the more variation we put on σ_r, the more diverse intensities will contribute to the weight, hence increasing the chances of blurring the edges as well. The final gaussian response will be more spread out. For its higher value, the filter produces foggy

fig. 11: Vintage Kathmandu

fig. 12: $\sigma_s = 3\sigma_s = 3, \sigma_r = 0.1$

fig. 13: $\sigma_s = 3, \sigma_r = 0.3$


fig. 14: Original Cat Image


fig. 15: $\sigma_s = 4, \sigma_r = 0.2$


fig. 16: $\sigma_s = 4, \sigma_r = 0.4$

fig. 17: Original Lena Image

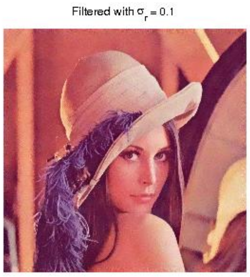

fig. 18: $\sigma_s = 3, \sigma_r = 0.1$

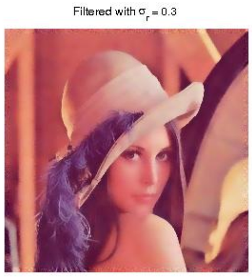

fig. 19: $\sigma_s = 3, \sigma_r = 0.3$

effect where only prominent edges are seen as edges and less significant ones are blurred. Such results would be usual for visual effects. We have implemented color filtering too. Here we have used CIELAB color space and calculated a combined range gaussian response at pixel p as:

$$G_p = \exp \frac{dL^2 + da^2 + db^2}{2 * \sigma_r} \tag{4}$$

where dL, da ,and db are the difference in L, a and b channels between anchor pixel and its neighbourhood.

4 Shortcomings of Bilateral Filter

Bilateral Filter is a non linear filter, i.e., it's weight varies with each pixels. Hence, we need to calculate these weights for each pixels in the image which greatly increases the computation time. Our implementation in particular, is a brute force implementation which has complexity of $O(Nr^2)$. As the kernel radius r increases, the computation time grows quadratically.
There has been many efficient implementation of this algorithm like separable kernel, local histogram,

layered approximation, bilateral grid [1] but they are all approximation and require high degreee of customization to achieve satisfactory results [3].
We have moved on to implementing, Guided Image Filtering which is much faster than bilateral filtering and has many useful application as well.

5 Guided Image filtering

Guided Image filtering is a recent technique developed by Kaiminget al. which can behave more than just an edge-preserving smoothing operator. It can transfer image structure, fuse images, enhance details, compress, etc. It is currently one of the fastest edge-preserving filters[]. It uses additional guiding filter, which can also be the input image itself to filter the input image. In such case it behaves as a bilateral filter[].

5.1 Algorithm

The key notion of the algorithm is that filter output q is a linear combination of guiding image I in a window w_k centered at k.

$$q_i = a_k I_i + b_k \in w_k \tag{5}$$

where (a_k, b_k) are the linear coefficients unique for w_k. This model conveys that the output image will have edges depending upon the edges in guiding image I. To compute a_k and b_k, [3] specifies a cost function to be minimized given as:

$$E(a_k, b_k) = \sum_{i \in w_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2 \tag{6}$$

where p is the input image and ϵ is a regularization parameter for large values of a_k. The solution for a_k and b_k is given as:

$$a_k = \frac{\frac{1}{|w|} \sum_{i \in w_k} (I_i p_i - \mu_k \overline{p}_k)}{\sigma_k^2 + \epsilon} b_k = \overline{p}_k - a_k \mu_k \tag{7}$$

where μ_k and σ_k are the mean and variance of guided image I in w_k, $|w|$ is the number of pixels in w_k and $\overline{p}_k$ is the mean of input image in w_k. From there we arrive to the ouput image q as:

$$q_i = \frac{1}{|w|} \sum_{k, i \in w_k} (a_k I_i + b_k) = \overline{a}_i I_i + \overline{b}_i \tag{8}$$

where $\overline{a}_i = \frac{1}{|w|} \sum_{k \in w_k} a_k$ and $\overline{b}_i = \frac{1}{|w|} \sum_{k \in w_k} b_k$.

5.2 Results

We can see the figure 20, the result being equivalent to the result after bilateral filtering 12. We have used the same input image as the guiding image. The figure 21 shows the edge response stored in coefficient matrix a_k and the low frequency components stored in b_k. With their linear combination on guiding image, we produce the output at 20. The guided image drives the algorithm to output edges that are significant on it. Furthermore, we have tried to fuse two images using two different images of

fig. 20: $\sigma_s = 3\sigma_s = 3, \sigma_r = 0.2$
filtered image

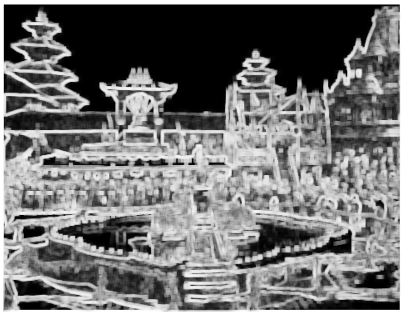

fig. 21: $\sigma_s = 3\sigma_s = 3, \sigma_r = 0.2$
high frequency map

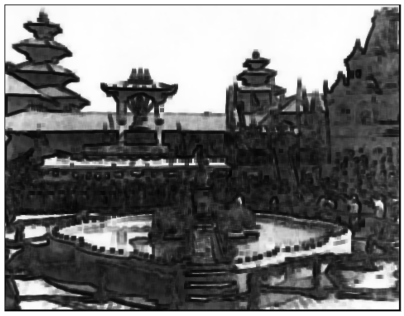

fig. 22: $\sigma_s = 3, \sigma_r = 0.2$
low frequency map

same scene for input and guiding images. The figure on the left of 23 is the guiding image where edges are prominent. The middle figure has a big patch of blur which could be thought of as a haze or a fog that hinders the edges to be captured. The right figure shows the fusion of these images using guided image filtering. Here the edges has been recovered to some extent with a bit of a presence of thin fog like effect.

fig. 23: Fusing two images: $left$: guiding image
$middle$: Input image, $right$: output fused image, $k = 3, \sigma_s = 2, \sigma_r = 0.1$

6 Conclusion

The bilateral filtering is able to smooth the edges while preserving the edges. But the high computation time (complexity of $O(Nr^2)$) makes it less appealing. Eventhough, it has been modified to approximate and faster versions, the guided filter with complexity of $(O(N))$ and independence from window radius r makes it far more efficient. We did not study the compressive behaviour of these filters but [3] states that the gradient inversion in edges that is common error in bilateral filtering is not present in guided image filtering. The later one is much faster and can provid flexibility via the use of guiding image which is indeed much preferred than the other one.

References

[1] Bilateral Filtering: Theory and Applications, Sylvain Paris, Pierre Kornprobst, Jack Tumblin, Fredo Durand, 2008

[2] Products and Convolutions of Gaussian Probability Density Functions, P.A. Bromiley, 2003

[3] Guided Image Filtering, Kaiming He, Jian Sun, Xiaoou Tang, Microsoft Research Asia

[4] Guided Image Filtering, Kaiming He, Jian Sun and Xiaoou Tang, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 35, 2013

[5] Bilateral Filtering for Gray and Color Images, C. Tomasi and R. Manduchi. ,In Proc. of the IEEE ,International Conference on Computer Vision, 1998.