IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 3, Ver. III (May - June 2017), PP 17-27
www.iosrjournals.org

Comprehension Bilateral Filtering

J.Sreenivasa Reddy[1] and B.Rama Bhupal Reddy[2]

1. *Department of E.C.E, KSRM College of Engineering (Autonomous), Kadapa, A.P.*
2. *Department of Mathematics, KSRM College of Engineering (Autonomous), Kadapa, A.P.*

Abstract: *The main of the paper presents an efficient constant-time bilateral filter that produces a near-optimal performance tradeoff between approximate accuracy and computational complexity without any complicated parameter adjustment, called a compressive bilateral filter (CBLF). The constant-time means that the computational complexity is independent of its filter window size. Although many existing constant-time bilateral filters have been proposed step by step to pursue a more efficient performance tradeoff, they have less focused on the optimal tradeoff for their own frameworks. It is important to discuss this question because it can reveal whether or not a constant-time algorithm still has plenty room for improvements of performance tradeoff. This paper tackles the question from a viewpoint of compressibility and highlights the fact that state of-the-art algorithms have not yet touched the optimal tradeoff. The CBLF achieves a near optimal performance tradeoff by two key ideas: approximate Gaussian range kernel through Fourier analysis and period length optimization. Experiments demonstrate that the CBLF significantly outperforms state-of-the-art algorithms in terms of approximate accuracy, computational complexity, and usability.*

Key words: *Edge-preserving smoothing, constant-time bi-lateral filtering, compressibility*

I. Introduction

The bilateral filter (BLF), which Aurich and Weulefirst proposed, Smith and Brady rediscovered, and Tomasi and Manduchi also found and coined the name in 1990s, is a fundamental tool for edge-preserving smoothing in image processing, computer vision, and computer graphics. The BLF has flourished in various tasks including denoising, high-dynamic range imaging , stereo matching . This success comes from the following two reasons. First, it is designed based on the clear concept that its filter coefficients are determined from two laterals: pixel position (spatial domain) and pixel intensity (range domain). Second, as studied, it produces high smoothing efficiency without iterative processing in contrast to other edge preserving smoothing filters including the anisotropic diffusion and the weighted least square. The BLF has been actively extended in the past to further enhance the smoothing efficiency or to reduce the computational complexity.

A natural extension of the BLF is the cross/joint extension. In filtering a noisy target image, the original BLF diminishes the smoothing efficiency due to the filter coefficients determined from its noisy pixel intensities, whereas the cross/joint extension addresses this problem by determining them from a guide image captured under a different photographic condition instead. Another algorithmic extension is accelerated algorithms for the BLF including constant-time BLF. The original BLF has computational complexity proportional to the filter window area. This causes unacceptable running time in a recent trend toward higher-resolution or higher dimensional image processing that tends to require a larger filter window. On the other hand, the accelerated algorithms run in logarithmic-time or constant-time with a slight sacrifice of accuracy. This paper focuses on the BLF but our discussion is also applicable to the cross/joint extension.

We review major existing accelerated algorithms of the BLF. Most of them share a general framework that they approximate a BLF by a bunch of spatial linear filters, i.e., convolutions. In dealing with a large filter window, each spatial filter is generally operated by the Fast Fourier Transform (FFT), the extended integral images, or the recursive filters. For instance, Weiss designed a BLF specialized in the box spatial kernel that can be convolved fast by a histogram technique.Durand and Dorsey showed that a BLF with arbitrary spatial/range kernels can be decomposed into a bunch of spatial filters via quantization and interpolation techniques. Yang et al. enhanced that performance by a refined quantization technique. Paris and Durand proposed a more general quantization-based algorithm via the FFT. Chen et al. accelerated it by using a compact data structure and the graphical processing unit (GPU). Chaudhury et al.found that a BLF with a shiftable range kernel, e.g., trigonometric kernel and polynomial kernel, can also be approximated by spatial filters without quantization. Subsequently, Chaudhury significantly reduced the number of the spatial filters through improving its formulation and parameter configuration. As an example with a different framework, Yoshizawa et al. accelerated a BLF with Gaussian spatial/range kernels by reinterpreting it as the Gauss transform to point clouds.

A principal evaluation criterion for the BLF is the performance trade off between computational complexity and approximate accuracy. The existing algorithms have been developed step by step to pursue a more efficient performance trade off. Because the aforementioned framework decomposes a BLF into a bunch of spatial filters, its computational complexity is dominated by the number of the spatial filters M. The larger M

DOI: 10.9790/2834-1203031727

provides the higher approximate accuracy. In short, we aim at a sufficient approximate accuracy by as small M as possible. In terms of the performance trade off, Yang's algorithm, which is based on the quantization technique, and Chaudhury's algorithm, which uses the shiftable kernel technique, are state-of-the-art in the BLF.However, they still cannot satisfy modern applications including online video processing and high-dimensional volume data processing. It is worthwhile for various applications to improve the performance trade off without special hardware.

More importantly, the previous work has never discussed the performance optimality of their own frameworks. For example, it is unclear in Chaudhury's algorithm what shiftable kernel can achieve the optimal performance trade off. It is important to answer this question because it can reveal whether or not the state-of-the-art algorithms still have plenty room for improvements in performance trade off by comparing with the optimal one. As the title suggests, we tackle this question from a viewpoint of compressibility. As first demonstrated, a BLF can be extended to a ratio of convolutions in higher dimensional space spanned by spatial and range domains.

This extended space seems efficiently-compressible because they are extensively generated from a target image and low frequency kernels. Yang's algorithm approximates th ese data by quantization and interpolation techniques; we approximate them via lossy compression. Our compression framework makes the performance trade off conceptually equivalent to compression rate in a lossy fashion. This novel insight also covers the theoretical foundations of Chaudhury's algorithm more simply but more broadly. Furthermore, through simple considerations based on compressibility, we also reveal the optimal performance tradeoff of this compression framework and find much room for improvements of Chaudhury's algorithm.

This paper presents an efficient BLF algorithm designed through the above analysis based on compressibility, called compressive BLF (CBLF). The CBLF significantly outperforms the state-of-the-art algorithms in terms of performance trade off both in theory and practice. We first show our compression framework for the BLF and then clarify its optimal performance trade off. This analysis indicates a large potential to improve the performance trade off of Chaudhury'salgorithm. In order to achieve a near-optimal performance trade off in a more simple fashion, we establish two key ideas: an approximate Gaussian range kernel through Fourier analysis and period length optimization. Experiments demonstrate that the CBLF clearly shows a more efficient performance trade off than the state-of-the-art algorithms. Specifically, as compared with Chaudhury's algorithm, the CBLF reduces M by approximately 60% without a loss of accuracy and runs 5–8times faster. In comparison with Yang's algorithm, it succeeds to reduce *M* by roughly 25–50%. It is worth noting that the CBLF requires no special device for hardware acceleration and no assumption on target images. Moreover, it provides a simple-but-precise and user-friendly accuracy control method.

The major contributions of this paper are summarized here:
1) We discuss the BLF algorithms from a viewpoint of compressibility and reveal plenty room for improving their performance tradeoff through simple compression theories.
2) We establish efficient approximate Gaussian kernel through Fourier analysis and the period length optimization. As experimentally-ascertained, our approach attains near-optimalperformance trade-off if we use a significantly-low tolerance.
3) We surely broaden the range of applications for the BLF by significantly improving the performance of the BLF.

II. Digital Image Processing

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays. We implemented pre-processing techniques using MATLAB.

Fundamental Steps in Digital Image Processing

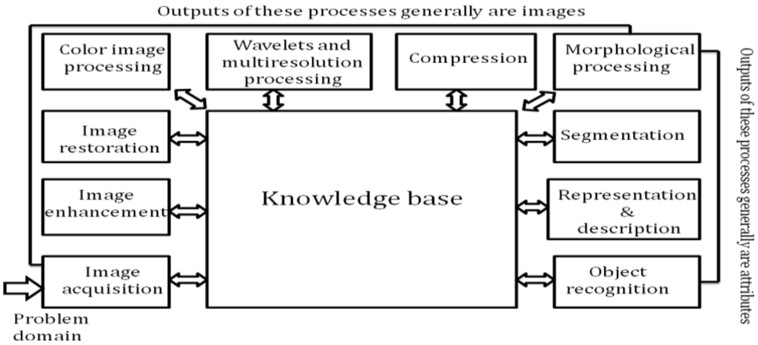


Fig.1 Fundamental Steps In Digital Image Processing

III. Existing System

In the previous section we combined range filtering with domain filtering to produce bilateral filters. We now show that this combination is essential. For notational simplicity, we limit our discussion to black-and-white images, but analogous results apply to multiband images as well. The main point of this section is that range filtering by itself merely modifies the gray map of the image it is applied to. This is a direct consequence of the fact that a range filter has no notion of space. Let be the frequency distribution of gray levels in the input image. In the discrete case is the gray level histogram: - is typically an integer between 0 and 255, and is the fraction of image pixels that have a gray value of -. In the continuous case is the fraction of image area whose gray values are between - and - + d-. For notational consistency, we continue our discussion in the continuous case, as in the section. It is useful to analyse the nature of this gray map transformation in view of our discussion of bilateral filtering. Specifically, we want to show that Range filtering compresses unimodal histograms.

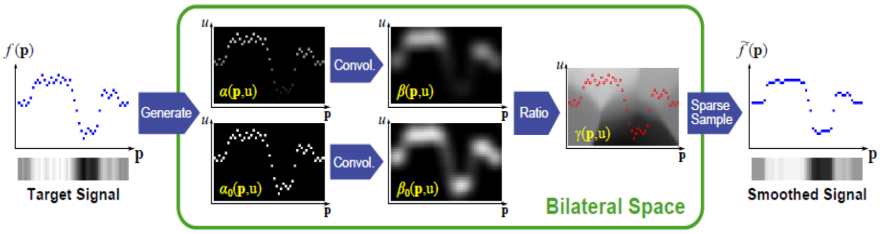


Fig.2 BLF operated via bilateral space

A case of one-dimensional signals is shown to visualize bilateral space as two-dimensional one. In the graph of (**p**; *u*), the red boxes indicate the sampling points to construct the smoothed signal.

IV. Proposed System

Edge-preserving smoothing

This introduces a new approach to edge-preserving smoothing of digital images. It is designed for an effective elimination of the image noise within digital images what is an essential step in applications like edge detection or image segmentation. The approach presented here tries to overcome some of the disadvantages of existing smoothing filters and is conceived as an extension of the edge-preserving Maximum Homogeneity Neighbour Filter. The algorithm cleans up the image noise in the homogeneous areas, but preserves all image structures like edges or corners. It is shown that the new filter algorithm combines the advantageous features of different types of filters. The algorithm is not only applicable to grayscale images, but can be extended to multi-channel data, like color images too. The performance of the algorithm is achieved by a more complex and differentiating treatment of the image data compared to conventional concepts. This paper introduces a new approach to edge-preserving smoothing of digital images as a preprocessing step for feature extraction and/or image segmentation. Although there are many filtering algorithms available, the results of these algorithms are not satisfying for all applications. In specific cases of object detection, as to be performed in an ongoing research project, standard algorithms may have deficiencies. Here, the detection process has to work even under bad conditions, making it necessary to have a reliable algorithm. The objects to be detected within the digital images are sometimes very small (eg. down to 3*3 pixels), the contrast is sometimes low and considerable radiometric noise has to be allowed. In every digital image there is a certain amount of white noise. If feature extraction algorithms like interest point operators or line extractors are applied or if an image segmentation will be done, many of the features to be found, like small edge elements, for example, are only expressing the existing image noise. To avoid these annoying effects, which only consume computation time and affect the real image features, firstly a filtering process has to be applied to the original digital image, in order to get rid of the image noise. Then the problem arises, that the smoothing of the image within the homogeneous areas has to be achieved without blurring those gray value changes which are containing the information needed. Especially for following feature extraction steps it is essential to preserve the edges and corners, with respect to their geometrical shape and magnitude of the gradient.

V. Smoothing Algorithms

The most commonly used smoothing algorithms are the n*n-box filter and the Gaussian Kernel Smoothing (Binomial Filter). Here, a square-sized convolution kernel is applied to each pixel. With the parameter "box size" (n) respectively "sigma", the degree of smoothing can be controlled. This smoothing strategy results in very nicely smoothed homogeneous areas, however possessing the disadvantage to blur gray value edges. This leads to three problems:

- The edge region will be extended due to averaging of edge and non edge pixel. So the edge may come closer to other edges with the effect that they are no longer detectable as separate edges.
- Corners are rounded off and small crinkles are blurred. This means that image structures are geometrically damaged, what might be fatal for the following feature extraction.
- The magnitude of the gradients is degraded. In the worst case, the edge cannot be detected at all, because the magnitude falls beyond some given threshold.

Edge-preserving algorithms

Edge-preserving smoothing filters are much more suitable for feature extraction. Some examples of this filter class are:

- Median Filter
- Symmetrical Nearest Neighbour Filter (SNN)
- Maximum Homogeneity Neighbour Filter (MHN)
- Conditional Averaging Filter

These non linear algorithms are calculating the filtered gray value in dependence of the content of a defined neighbourhood. From the list of the neighbourhood pixels, only these are taken for the averaging, which have similar gray values compared to the pixel in consideration. Each edge-preserving filter has its own specific algorithm, but they all have in common, that the effect of this smoothing strategy is to preserve edges. Unfortunately, these smoothing filters have the characteristic not to smooth satisfyingly, because small gray value fluctuations existing in the really homogeneous areas are emphasized and not reduced. In addition, the Symmetrical Nearest Neighbour Filter is unable to produce reliable results in case of small areas.

In order to overcome the problems mentioned above an extended algorithmic concept has been designed which has the following features:

- high degree of smoothing in the homogeneous areas
- preserving of edges, even those ones being represented by small gray value changes
- conservation of very small homogeneous image regions.

The algorithmic extension is based on the edge-preserving Maximum Homogeneity Neighbour Filter, because it meets the last two demands. The concept of this filter is enhanced, taking the strategy of segmentation techniques like region growing into consideration, providing a higher degree of smoothing in the homogeneous areas.

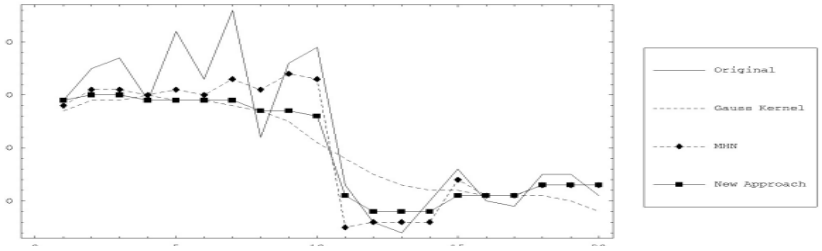


Fig .3 Edge Preserving Graph

Edge Preserving Filters on Color Images

Color images as processed in various applications are recorded by different acquisition devices. Cameras as well as scanners have their specific noise characteristics. Image transmission may as well introduce noise into the image data. Typical models for noise are either Gaussian or salt-and-pepper noise; Gaussian noise is used as a model for sensor errors, drop-outs during transmission and errors on the CCD chip can be modeled by salt-and-pepper noise. In our contribution we use images from a data base and disturb them by artificial noise of varying degree and type. we describe some non-linear smoothing filters, such as edge preserving smoothing algorithm , and extend them to color images where they were defined for gray-level images originally. We apply these filters to the disturbed input images and compare each result with its corresponding original image to compute difference measures. Standard measures are the signal-to-noise ratio (SNR) and maximum differences for color vectors. As all disturbances are modeled in RGB, we compute these measures in RGB as well. More important for human perception than SNR is the so-called ΔE difference which describes the perceived color difference. we conclude our contribution with a summary of the evaluation and the prospective work.Color Filters In research and literature there exist several filters that can be classified into linear and non-linear filters. Filters can either operate in the spatial or in the frequency domain. In the following we compare non-linear filters in the spatial domain and additionally use an AMF (arithmetic mean filter).

Linear Filters

A huge number of filters for single band images has been proposed in the long history of image processing. As color images became affordable with respect to sensors, memory and processor speed, some of these filters have been extended to color. If a linear filter, such as a Gaussian or mean filter, is applied to each channel of an RGB image separately, the resulting image will contain usually color triplets which are not present in the input image. Such artifacts yield perceptional differences which can be avoided by non-linear filtering. On the other hand, additive noise, such as additive Gaussian noise, can be removed by a low-pass filter which averages color vectors.

Non-Linear Filters

Filters which are quoted to smooth homogeneous areas while preserving edges are the
EPS (Edge preserving smoothing), presented for gray-level images,
SNN (Symmetric Nearest Neighbor Filter) described,
K-N (Kalahari-Nagao Filter) proposed in for gray-level images,
VMF (Vector Median Filter, also known as CVF (Color Vector Median))

EPS Filter In the so-called edge preserving smoothing algorithm introduced in the selection of gray-level pixels for averaging is done based on statistical principles. The algorithm uses nine different 5 × 5 masks for each pixel; three of them. The pixels marked in the neighborhood are used for the following computations. The symmetrical use results in eight different masks. Each of these masks includes seven points for the calculation of the new gray-level. The contrast mask includes nine elements for the following computations. For each mask we compute the variance.

Constant-time bilateral filtering

The objective of bilateral filtering is to smooth images. It is done by replacing the intensity (color) value of a pixel by the average of the values of other pixels weighted by their spatial distance and intensity similarity to the original pixel used this idea by identifying similar pixels by first detecting edges. They iteratively replaced the intensity of a pixel by the average of all the pixels in a small (3 × 3) neighborhood, and on the same side of the edge as the pixel itself identified similar pixels differently, by choosing those six of the nine pixels in the neighborhood that are closest in intensity to the original pixel, and used their median to obtain the smoothed value proposed a transform to compute the net similarity between a pixel and all other pixels in the image, as well as the direction in which the largest number of a pixel's most similar pixels are located. The latter was captured by computing a force vector at the pixel presented a detailed algorithm for multi-scale image segmentation using the force transform and demonstrated the performance advantages of the similarity measure incorporated in the force transform used the same definition of similarity as proposed and used it for image smoothing. They replaced pixel values with similarity-weighted averages and called it bilateral filtering. Bilateral filter has been demonstrated to be very effective for many computer vision and computer graphics tasks but its brute-force implementation is known to be computationally intensive when the filter kernel is large. Several enable it to be computed at either O(r) or O(log(r)) runtime in the radius of the filter. By filtering on the sub sampled image prove that the runtime decreases as the filter size increases because the sub sampling factor can be increased without significantly impacting the accuracy of the result. This method is relatively slow when the filter size is small. later show that the GPU implementation of (2002) can achieve video rate.

The median filter replaces each pixel p with the median of neighboring pixel values within a box region Ω_p. Let I_p denote the pixel value at p and I_p denote the corresponding median filte red value,

$$I_P^M = \arg\min_x \left(\sum_{q\in\Omega p} \left| I_q - x \right| \right)$$

given the fact that median actually minimizes the sum of absolute error of the given data (Huber and Wiley 1981). When the input signal is an 8-bit digital image, the possible median value only resides in a total of 256 digital numbers ranging from 0 to 255, that is x ∈ [0, 255] . Letting the absolute error be the cost, a cost volume C can then be computed given an 8-bit image:

$$C_{q,x} = \left| I_q - x \right|, \ x \in [0,255]$$

And the median filter is formulated as a standard cost aggregation problem followed by a winner-take-all selection. As a result, the main computational cost of the median filter resides in the cost aggregation step. The integral image enables the cost aggregation to be independent of the size of the filter; and thus the computational complexity of the median filter is linear regarding the number of image pixels and the intensity level. In practice, the cost volume can be down sampled in the intensity domain before aggregation to improve the efficiency. That is, 256 intensity values are quantized into only a total of N < 256 bins, and the cost is computed at each bin and each pixel location. Instead of using a winner-take-all selection, a quadratic interpolation step is used to up sample the intensity domain after cost aggregation. Specifically, let C^A denote the aggregated cost volume. A quadratic function same as the input image and the shape of the aggregated cost is exactly the same as the absolute error function presented. In this case, the ground truth can be obtained by either interpolating the aggregated cost using this absolute error function or the quadratic function presented. However, in real images, the intensity values within a local region are likely to be similar but normally different; thus interpolation with a quadratic function is proved to be more accurate. The eight images used in this experiment were selected from different categories. These images were chosen to be as different as possible to cover a broad spectrum of content including the following:

Artificial image with various edges, frequencies, and white noise;

Architectural picture structured along two main directions; and

Photograph of natural scenes with a more stochastic structure.

A bilateral filter has two filter kernels: a spatial filter kernel F(p, q) that measures spatial distance between two pixel locations p and q and a range filter kernel G(I_p, I_q) that measures the intensity/range distance between two pixel

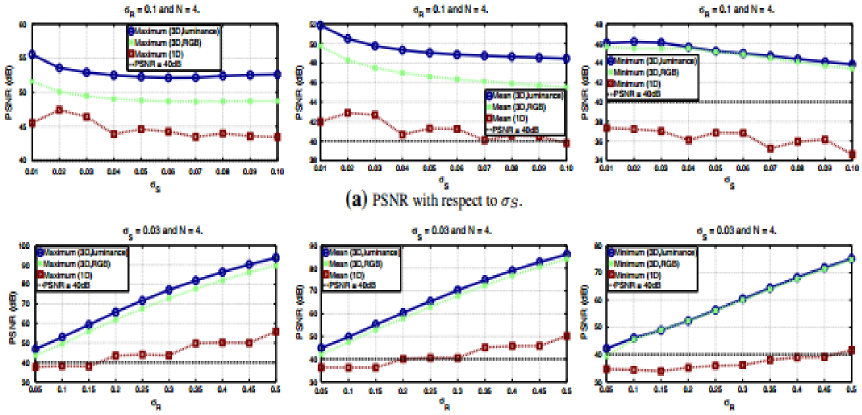
is a box filter the cost volume can be aggregated using an integral image; the computational complexity of the bilateral filter is thus independent of the filter size. Bilateral filtering using other spatial filter kernels can actually be computed in exactly the same manner by filtering the cost volume using the corresponding spatial filter kernels. If the spatial filter kernel has a recursive implementation, the computational complexity of the spatial filter will be independent of the filter kernel size by adjusting the coefficients of the recursive system. Three spatial filters are tested in this project including the box filter, exponential filter, and recursive Gaussian filter. The box filter kernel size is again defined using a single parameter σS∈ [0, 1] : round(2σS H + 1) × round(2σSW + 1), where H and W are the height and width of the input image. The exponential filter and recursive Gaussian filter can be implemented recursively, and the kernel size is controlled by the coefficients of the recursive system determined also by σS. In our experiments, the exponential filter uses exp(− √2/σS) as the feedback coefficient, and Gaussian filter uses σS as the standard deviation. Unlike median filtering that requires a single cost volume, two cost volumes (one for computing Kp and the other for ωp) is used in the proposed bilateral filtering method; thus the computation cost will be theoretically twice higher. However, the numerical comparison presented shows that the performance of the proposed bilateral filter will be higher than the proposed median filter when the same amount of quantization levels N is used. The red curve shows that when N ≥ 14, the minimum PSNR value will be greater than 40 dB which corresponds to almost invisible error, while the red curves in reveal that only N ≥ 6 is required. Only the box spatial filter is analyzed because the performance w.r.t. N is extremely close between different spatial filter kernels. Figures 7 and 8 evaluate the performance of the proposed bilateral filter with respect to the filter kernel size defined by σS∈ [0, 1] and σR∈ [0, 1] 3 when the quantization level N = 8. Two other spatial filters—exponential filter and recursive Gaussian filter - are used. As can be seen, the performance is robust to σS but increases w.r.t. σR. Nevertheless, the minimum PSNR value (shown as red curves) is always over 40 dB when σR ≥ 0.05. The bilateral filtering results of the Tulip test image in are presented. In fact, there is no noticeable error.

6.6 Multi-Channel Bilateral Filtering

If the guidance image T used to compute the range filter kernel is a n-channel image T where n > 1 (e.g. , a color image when n = 3), becomes:

$$k_p = I_P^T.\omega_P = \sum_{q\in\Omega P} 1.\varsigma\left(T_P, T_q\right)I_q$$

Where T_p and T_q are n-channel vectors extracted from the guidance image at pixel p and q, respectively. For 8-bit images, Tp has up to 256n potential vector values. Hence, the same as the one-channel case, multi-channel bilateral filtering can be formulated as a cost aggregation problem, except that the computational complexity will be exponential in the dimensionality n. In practice, the computational complexity can be reduced without introducing significant errors by down sampling the multi-dimensional space. Assuming that a total of N values are used in each dimension, the computational complexity will be O (Nn) at each pixel location. However, the components of the multi-dimensional space are normally highly correlated; thus a relatively smaller N value can be used for a multi-channel bilateral filter as demonstrated. Similar to the one-channel case, a linear interpolation is performed after cost aggregation to obtain the final bilateral filtered value. The linear interpolation is straightforward in theory, except when memory cost is taken into account. Considering the one-channel bilateral filter again, let Tpdenote the pixel value of the guidance image at pixel location p, Ll denote the l-th selected intensity level, ΔL denote the distance between two successive selected intensity levels,



(a) PSNR with respect to σ_S.

Fig.:4 Multi-Channel Bilateral Filtering graphs

SUGIMOTO *et al.*: COMPRESSIVE BILATERAL FILTERING

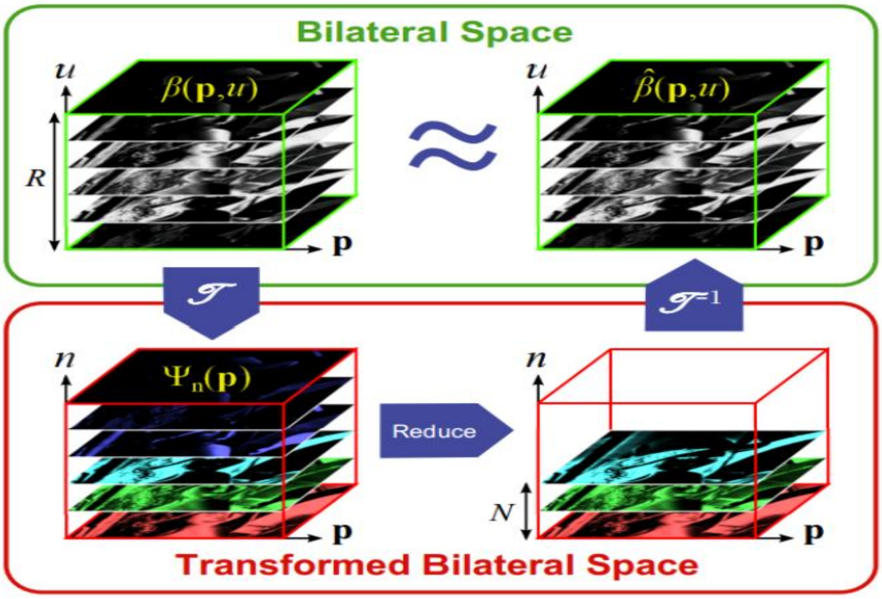


Fig.5 conceptual illustration of compression

A conceptual illustration of our compression approach denoted by^_(p; *u*) = *Cu* [_(p; *u*)]. In the transform domain, the redder layer indicates the higher energy. We decompress ^_(p; *u*) from the most *N* significant transform components *n*(p), generated by *N* spatial filters.

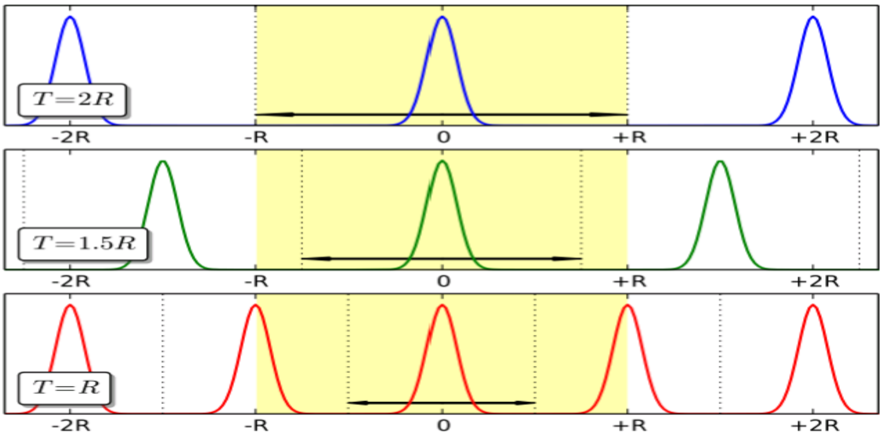


Fig: Periodic Gaussian kernels with three varied period lengths.

The yellow area indicates the domain of the range kernel [□*R;R*] and each black arrow indicates the central period of each periodic kernel. If *T* = 2*R* (top,blue) or *T* = 1:5*R* (middle, green), they preserve accurate Gaussian shapes over the whole domain; whereas, if *T* = *R* (bottom, red), the Gaussian shape is deformed by the two adjoining Gaussian profiles that intrude the domain.

VI. Experimental Results

Bilateral Filtering GPA Result

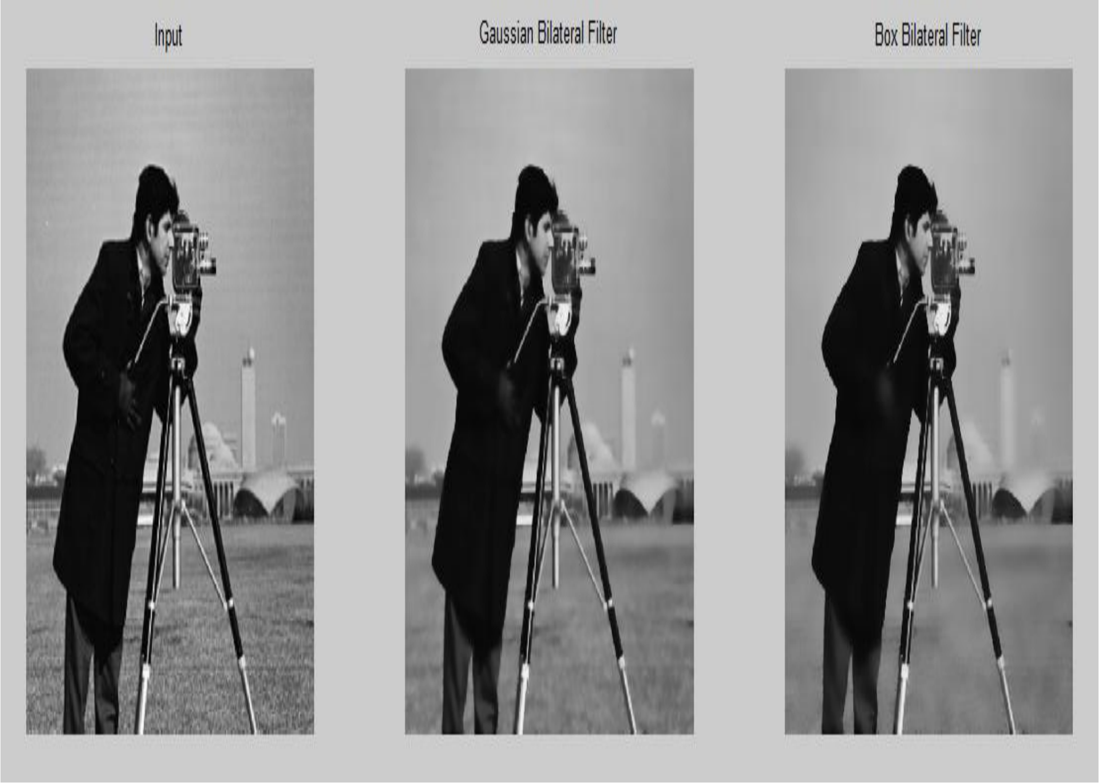
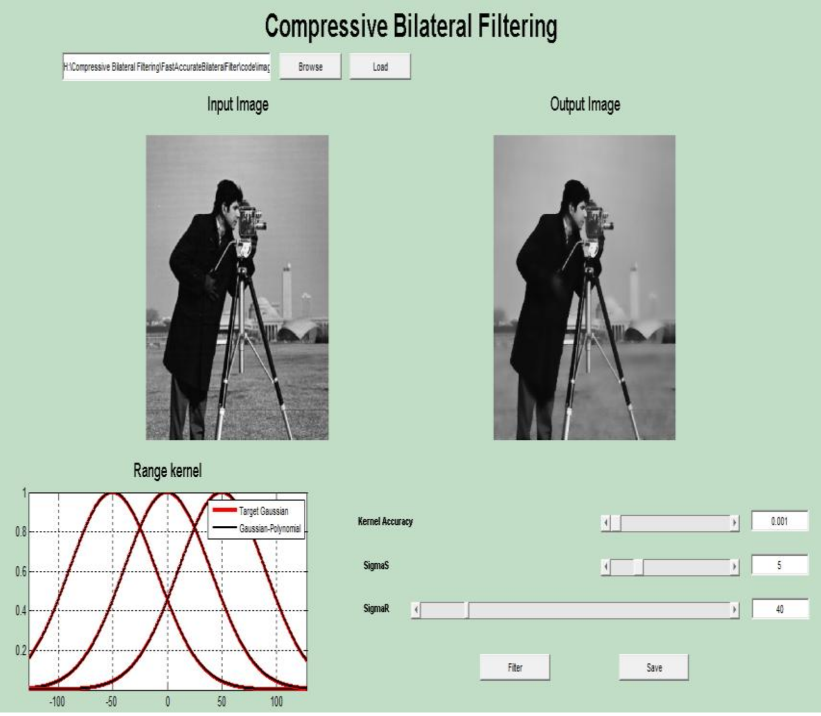


Fig .6 Bilateral Filtering GPA Results

Compressive Bilateral Filtering



H:\Compressive Bilateral Filtering\FastAccurateBilateralFilter\code\imag

Kernel Accuracy 0.001

SigmaS 5

SigmaR 40

Fig.7 Compressive Bilateral Filtering Output

VII. Conclusion

This Paper presented an BLF algorithm that provides the most efficient performance tradeoff of computational complexity and approximate accuracy without any complicated parameter adjustment. We designed a general framework for the BLF from a viewpoint of compressibility and discussed the optimal performance tradeoff that is achieved via the KLT. Next, we aimed at a near-optimal performance tradeoff by establishing two key ideas: approximate Gaussian range kernel through Fourier analysis and the period length optimizationfor its parameter setting. These simple-but-effective techniques enabled us to further enhance the performance tradeoff for the BLF. From the results of our various experiments,we conclude that our compressive approach is surely effective to bilateral filtering and believe that the CBLF will be the first choice for many applications with the BLF in image processing, computer vision, and computer graphics.

Finally, our future work is outlined as follows:

1) The CBLF should be generalized to arbitrary range kernels. Although its general framework seems to be straightforwardly extended to arbitrary range kernels, our current algorithm is effective only against range kernels with narrowband spectrum such as Gaussian range kernel due to compression through the Fourier series.

2) Designing an efficient CBLF using the KLT willbe an important remaining problem for achieving the optimal performance tradeoff. This will enable us to outperform Yang’s algorithm over the whole range of σ_r.

3) Parallelization of the CBLF through hardware acceleration will be an inevitable task for filtering volume data on a time scale of milliseconds. Actually, as reported, even BLF of voxel data has generally taken on the ten-second time scale at the present day.

4) Our key ideas will be also applicable to generalizations of the BLF, e.g., the trilateral filter , the non-local means filter , and the guided filter We believe that the CBLF has a large potential to make the modern filtering algorithms more practical for and more applicable to image processing applications.

VIII. Future Scope

- To enhance the visibility of digital images.
- To reduce the random noise from the images.
- To remove the fog or haze from the images.
- To filter in such a way that it can preserves the edges.

As it is known bilateral filter is unable to remove salt and pepper noise so in near future we will extend this research work by integrating the bilateral filter with median filter. Because we know that the median filter can remove salt and pepper noise.

References

[1] V. Aurich and J. Weule, “Non-linear Gaussian filters performing edge preserving diffusion,” in Mustererkennung 1995, 17. DAGM-Symposium, 1995, pp. 538–545.

[2] S. M. Smith and J. M. Brady, “SUSAN — a new approach to low level image processing,” Int. J. Comput. Vis. (IJCV), vol. 23, no. 1, pp. 45–78, 1997.

[3] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Jan. 1998, pp. 839–846.

[4] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, “Digital photography with flash and no-flash image pairs,” ACM Trans. Graph. (Proc. SIGGRAPH), vol. 1, no. 212, pp. 664–672, Aug. 2004.

[5] E. Eisemann and F. Durand, “Flash photography enhancement via intrinsic relighting,” ACM Trans. Graph.(Proc. SIGGRAPH), vol. 1, no. 212, pp. 673–678, 2004.

[6] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” Multiscale Modeling and Simulation, vol. 4, no. 2, pp. 490–530, Jan. 2005.

[7] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” ACM Trans. Graph.(Proc. SIGGRAPH), pp. 257–266, 2002.

[8] J. Chen, S. Paris, and F. Durand, “Real-time edge-aware image processing with the bilateral grid,” ACM Trans. Graph. (Proc. SIGGRAPH), vol. 26, no. 3, pp. 103.1–103.9, 2007.

[9] J. Kuang, G. M. Johnson, and M. D. Fairchild, “iCAM06: a refined image appearance model for HDR image rendering,” J. VisualCommunication and Image Representation, vol. 18, no. 5, pp. 406–414, Oct. 2007.

[10] Q. Yang, R. Yang, J. Davis, and D. Nister, “Spatial-depth super resolution for range images,” in Proc. IEEE Conf. Comput. Vis. PatternRecognit. (CVPR), Jun. 2007.

[11] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling,” IEEE Trans. Pattern Anal. Mach.Intell., vol. 31, no. 3, pp. 492–504, Mar. 2009.

[12] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, “A gentle introduction to bilateral filtering and its applications,” in ACMSIGGRAPH classes, 2008.

[13] M. Elad, “On the origin of the bilateral filter and ways to improve it,” IEEE Trans. Image Process., vol. 11, no. 10, pp. 1141–1151, Jan. 2002.

[14] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 7, pp. 629–639, Jul. 1990.

[15] R. L. Lagendijk, J. Biemond, and D. E. Boekee, “Regularized iterative image restoration with ringing reduction,” IEEE Trans. Acoust., Speech,Signal Process., vol. 36, no. 12, pp. 1874–1888, 1988.

[16] B. Weiss, “Fast median and bilateral filtering,” ACM Trans. Graph. (Proc. SIGGRAPH), vol. 25, no. 3, pp. 519–526, Jul. 2006.

[17] F. Porikli, “Constant time O(1) bilateral filtering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2008, pp. 1–8.

[18] [18] S. Paris and F. Durand, “A fast approximation of the bilateral filter using a signal processing approach,” Int. J. Comput. Vis. (IJCV), vol. 81, no. 1, pp. 24–52, Jan. 2009.

[19] Q. Yang, K. H. Tan, and N. Ahuja, “Real-time O(1) bilateral filtering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), no. 1, Jun. 2009, pp. 557–564.

[20] S. Yoshizawa, A. Belyaev, and H. Yokota, “Fast Gauss bilateral filtering,” Computer Graphics Forum, vol. 29, no. 1, pp. 60–74, Mar. 2010.

[21] K. N. Chaudhury, D. Sage, and M. Unser, “Fast O(1) bilateral filtering using trigonometric range kernels,” IEEE Trans. Image Process., vol. 20, no. 12, pp. 3376–82, Dec. 2011.

[22] K. N. Chaudhury, “Constant-time filtering using shiftable kernels,” IEEE Signal Processing Letters, vol. 18, no. 11, pp. 651–654, Nov. 2011.

[23] “Acceleration of the shiftableO(1) algorithm for bilateral filtering and nonlocal means,” IEEE Trans. Image Process., vol. 22, no. 4, pp. 1291–1300, Apr. 2013.

[24] F. C. Crow, “Summed-area tables for texture mapping,” ACM Trans. Graph. (Proc. SIGGRAPH), vol. 18, no. 3, pp. 207–212, 1984.

[25] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR), vol. 1, 2001, pp. 511–518.

[26] M. Hussein, F. Porikli, and L. Davis, “Kernel integral images: A framework for fast non-uniform filtering,” in Proc. IEEE Conf. Comput.Vis. Pattern Recognit. (CVPR), Jun. 2008, pp.1–8.

[27] E. Elboher and M. Werman, “Cosine integral images for fast spatial and range filtering,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2011, pp. 89–92.

[28] R. Deriche, “Fast algorithms for low-level vision,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 1, pp. 78–87, 1990.

[29] I. T. Young and L. J. van Vliet, “Recursive implementation of the Gaussian filter,” Signal Process., vol. 44, no. 2, pp. 139–151, Jun. 1995.

[30] L. van Vliet, I. Young, and P. Verbeek, “Recursive Gaussian derivative filters,” in Proc. Int. Conf. Pattern Recognition (ICPR), vol. 1, no. 1, 1998, pp. 509–514.

[31] G. Farnebäck and C. Westin, “Improving Deriche-style recursive Gaussian filters,” J. Math. Imaging and Vis., vol. 26, no. 3, pp. 293–299, Nov. 2006.

[32] F. Porikli, “Integral histogram: a fast way to extract histograms in Cartesian spaces,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR), Jun. 2005, pp. 829–836 vol. 1.

[33] B. K. Gunturk, “Fast bilateral filter with arbitrary range and domain kernels,” IEEE Trans. Image Process., vol. 20, no. 9, pp. 2690–2696, Sep. 2011.

[34] S. Pan, X. An, and H.He, “Optimal bilateral filter with arbitrary spatial and range kernels using sparse approximation,” MathematicalProblems in Engineering, vol. 2014, pp. 1–11, 2014.

[35] W. Ray and R. Driver, “Further decomposition of the Karhunen-Loève series representation of a stationary random process,” IEEE Trans.Information Theory, vol. 16, no. 6, pp. 663–668, Nov. 1970.

[36] K. R. Rao and P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press, 1990.

[37] K. Sugimoto and S. Kamata, “Fast Gaussian filter with second-order shift property of DCT-5,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2013, pp. 514–518.

[38] Efficient constant-time Gaussian filtering with sliding DCT/DST- 5 and dual-domain error minimization,” ITE Trans. on Media Technologyand Applications, vol. 3, no. 1, pp. 12–21, 2015.

[39] P. Choudhury and J. Tumblin, “The trilateral filter for high contrast images and meshes,” in Proc. Eurographics Workshop on Rendering, 2003, pp. 186–196.

[40] A. Buades, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR), vol. 2, Jun. 2005, pp. 60–65.

[41] K. He, J. Sun, and X. Tang, “Guided image filtering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[42] P. Milanfar, “A tour of modern image filtering: new insights and methods, both practical and theoretical,” IEEE Signal Process. Mag., vol. 30, no. 1, pp. 106–128, Jan. 2013.