

Acceleration of Shiftable $O(1)$ Algorithm for Bilateral Filtering and Non-local means

D.Vishnu Vardhan¹, K. Jayachandra Reddy²

¹Assistant Professor, Department of ECE, JNTUA College of Engineering (Autonomous), Pulivendula 516390, Andhra Pradesh, India

²P.G Scholar, Department of ECE, JNTUA College of Engineering (Autonomous), Pulivendula 516390, Andhra Pradesh, India

Abstract: A direct implementation of the bilateral filter requires $O(\sigma_s^2)$ operations per pixel, where σ_s is the (effective) width of the spatial kernel. A fast implementation of the bilateral filter was recently proposed that require $O(1)$ operations per pixel with respect to σ_s . This is done by using trigonometric functions for the range kernel of the bilateral filter, and by exploiting their so-called shiftability property. In particular, a fast implementation of the Gaussian bilateral filter is realized by approximating the Gaussian range kernel using raised cosines. Later, it is demonstrated that this idea could be extended to a larger class of filters, including the popular non-local means filter. For an image with dynamic range $[0, T]$, the run time scaled as $O(T^2/\sigma_r^2)$ with σ_r . This made it difficult to implement narrow range kernels, particularly for images with large dynamic range. This project discusses this problem and propose some advanced methods to accelerate the implementation, in general, and for small σ_r in particular and also provides some experimental results to demonstrate the acceleration that is achieved using these modifications.

Keywords: Bilateral filter, non-local means, shiftability, Gaussian kernel, truncation, running maximum, max filter, recursive filter, $O(1)$ complexity.

1. Introduction

Image restoration [1] is a process to recover or reconstruct an image that has been degraded by using some prior knowledge of degradation model. Degradation comes in many forms such as motion blur, noise, and camera misfocus. In cases like motion blur, it is possible to come up with a very good estimate of the actual blurring function and "undo" the blur to restore the original image. In cases where the image is corrupted by noise, the best we may hope to do is to compensate for the degradation it caused.

Degradation model:

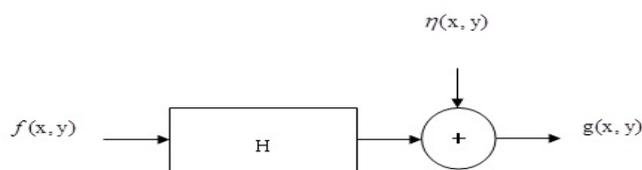


Figure 1: Degradation model

Where:

H: degradation operator.

η: noise.

f(x,y): input image.

g(x,y): degrade output.

The field of image restoration (sometimes referred to as image deblurring or image deconvolution) is concerned with the reconstruction or estimation of the uncorrupted image from a blurred and noisy one. Essentially, it tries to perform an operation on the image that is the inverse of the imperfections in the image formation system. In the use of image restoration methods, the characteristics of the degrading system and the noise are assumed to be known a priori. In practical situations, however, one may not be able to obtain this information directly from the image formation process.

Blurring is a form of bandwidth reduction of an ideal image owing to the imperfect image formation process. It can be caused by relative motion between the camera and the original scene, or by an optical system that is out of focus. When aerial photographs are produced for remote sensing purposes, blurs are introduced by atmospheric turbulence, aberrations in the optical system, and relative motion between the camera and the ground. Noise may be introduced by the medium through which the image is created (random absorption or scatter effects), by the recording medium (sensor noise), by measurement errors due to the limited accuracy of the recording system, and by quantization of the data for digital storage.

Image restoration algorithms distinguish themselves from image enhancement methods in that they are based on models for the degrading process and for the ideal image. For those cases where a fairly accurate blur model is available, powerful restoration algorithms can be arrived at. Unfortunately, in numerous practical cases of interest the modeling of the blur is unfeasible, rendering restoration impossible. The limited validity of blur models is often a factor of disappointment, but one should realize that if none of the blur models described in this chapter are applicable, the corrupted image may well be beyond restoration. Therefore, no matter how powerful blur identification and restoration algorithms are, the objective when capturing an image undeniably is to avoid the need for restoring the image.

The image restoration methods also fall under the class of linear spatially invariant restoration filters. We assume that the blurring function acts as a convolution kernel or point-spread function $d(n_1, n_2)$ that does not vary spatially. It is also assumed that the statistical properties (mean and correlation function) of the image and noise do not change spatially. Under these conditions the restoration process can be carried out by means of a linear filter of which the point-spread function is spatially invariant, i.e., is constant throughout the image.

These modeling assumptions can be mathematically formulated as follows. If we denote by $f(n_1, n_2)$ the desired ideal spatially discrete image that does not contain any blur or noise, then the recorded image $g(n_1, n_2)$ is modeled as in figure 1.

$$g(n_1, n_2) = d(n_1, n_2) * f(n_1, n_2) + w(n_1, n_2)$$

$$= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} d(k_1, k_2) f(n_1 - k_1, n_2 - k_2) + w(n_1, n_2) \dots (1)$$

Here $w(n_1, n_2)$ is the noise that corrupts the blurred image. Clearly the objective of image restoration is to make an estimate $\hat{f}(n_1, n_2)$ of the ideal image $f(n_1, n_2)$, given only the degraded image $g(n_1, n_2)$, the blurring function $d(n_1, n_2)$ and some formation about the statistical properties of the ideal image and the noise.

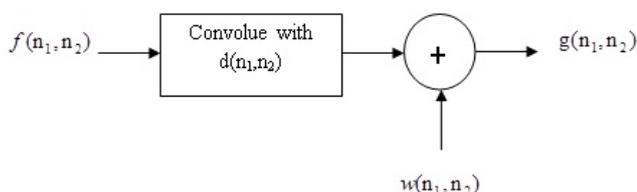


Figure 2: Image formation model in spatial domain

2. Bilateral Filtering

The bilateral filter is an edge-preserving diffusion filter, which was introduced by Tomasi et al [2]. The edge-preserving property comes from the use of a range kernel (along with the spatial kernel) that is used to control the diffusion in the vicinity of edges.

In this work, we will focus on the Gaussian bilateral filter where both the spatial and range kernels are Gaussian. This is given by

$$\tilde{f}(x) = \frac{1}{\eta} \int_{\Omega} g_{\sigma_s}(x-y) g_{\sigma_r}(f(x-y) - f(x)) f(x-y) dy \dots (2)$$

where

$$\eta = \int_{\Omega} g_{\sigma_s}(x-y) g_{\sigma_r}(f(x-y) - f(x)) dy$$

Here, $g_{\sigma_s}(x)$ is the centered Gaussian distribution on the plane with variance σ_s^2 , and $g_{\sigma_r}(s)$ (is the one-dimensional Gaussian distribution with variance σ_r^2 is the support of $g_{\sigma_s}(x)$ over which the averaging takes place. We call $g_{\sigma_s}(x)$ and $g_{\sigma_r}(s)$ the spatial and the range kernel. The range kernel is controlled by the local distribution of intensity. Sharp discontinuities (jumps) in intensity typically occur in the vicinity of edges. This is picked up by the range kernel, which is then used to inhibit the spatial diffusion. On the other hand, the range kernel becomes inoperative in regions with smooth variations in intensity. The spatial kernel

then takes over, and the bilateral filter behaves as a standard diffusion filter. Together, the spatial and range kernels perform smoothing in homogeneous regions, while preserving edges at the same time. The bilateral filter has found widespread use in several image processing, computer graphics, and computer vision applications. More recently, the bilateral filter was extended by Baudes et al [3] in the form of the non-local means filter, where the similarity between pixels is measured using patches centered around the pixel.

The direct implementation of bilateral filter requires $O(\sigma_s^2)$ operations per pixel. This makes the filter slow for real-time applications. Several efficient algorithms have been proposed in the past for implementing the filter in real time. Porikli [4] demonstrated for the first time that the bilateral filter could be implemented using $O(1)$ operations per pixel (with respect to σ_s). This was done for two different settings: (a) Spatial box filter and arbitrary range filter, and (b) Arbitrary spatial filter and polynomial range filter. The author extended (b) to the Gaussian bilateral filter in (2) by approximating $g_{\sigma_r}(s)$ with its Taylor polynomial. The run time of this approximation was linear in the order of the polynomial. The problem with Taylor polynomials, however, is that they provide good approximations of $g_{\sigma_r}(s)$ only locally around the origin. In particular, they have the following drawbacks:

- 1) Taylor polynomials are not guaranteed to be positive and monotonic away from the origin, where the approximation is poor. Moreover, they tend to blow up at the tails.
- 2) It is difficult to approximate $g_{\sigma_r}(s)$ using the Taylor expansion when σ_r is small.

In particular, a large order polynomial is required to get a good approximation of a narrow Gaussian, and this considerably increases the run time of the algorithm. Later Wiseman [5] proposed a fast median bilateral filtering.

In this project, it is observed that it is important that the kernel used to approximate $g_{\sigma_r}(s)$ be positive, monotonic, and symmetric. While it is easy to ensure symmetry, the other two properties are hard to enforce using Taylor approximations. It was noticed that, in the absence of these properties, the bilateral filter created strange artifacts in the processed image. It is proposed to fix this problem using the family of raised cosines, namely, functions of the form

$$\phi(s) = \left[\cos\left(\frac{\pi s}{2T}\right) \right]^N \quad (-T \leq s \leq T) \dots (3)$$

Here N is the order of the kernel, which controls the width of $\phi(s)$. The kernel can be made narrow by increasing N . The key parameter in (3) is the quantity T . The idea here is that $[\cos(s)]^N$ is guaranteed to be positive and monotonic provided that is restricted to the interval $[-\pi/2, \pi/2]$. Note that the arguments in (3) takes on the values $|f(x-y) - f(x)|$ as x and y varies over the image.

Therefore, by letting

$$T = \max_x \max_{\|y\| \leq R} |f(x-y) - f(x)|$$

One could guarantee $\phi(s)$ to be positive and monotonic over $[-T, T]$. T was simply set to the maximum dynamic range, for example, 255 for gray scale images. The polynomials suggested were of the form

$$\phi(s) = \left(1 - \frac{s^2}{T^2}\right)^N \quad (-T \leq s \leq T) \dots (4)$$

3. Non Local Means

Non-local means is an algorithm in image processing for image denoising. Unlike "local smoothing" filters, non-local means does not update a pixel's value with an average those of the pixels around it - instead, it updates it using a weighted average of the pixels judged to be most similar. The weight of each pixel depends on the distance between its intensity grey level vector and that of the target pixel. Non-local means algorithm was published by Antoni Buades [6]. If compared with other wellknown denoising techniques, such as the Gaussian smoothing model, the anisotropic diffusion model, the total variation denoising, the neighborhood filters and an elegant variant, the Wiener local empirical filter, the translation invariant wavelet thresholding, the non-local means method noise looks more like white noise. The NL-means algorithm, is defined by the simple formula

$$NL[u](x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{(G_a * |u(x+) - u(y+)|^2)(0)}{h^2}} u(y) dy \dots (5)$$

Where $x \in \Omega$, $c(x) = \int_{\Omega} e^{-\frac{(G_a * |u(x+) - u(y+)|^2)(0)}{h^2}} dz$

is a normalizing constant, G_a is a Gaussian kernel [7] and h acts as a filtering parameter. This formula amounts to say that the denoised value at x is a mean of the values of all points whose gaussian neighborhood looks like the neighborhood of x . The main difference of the NL-means algorithm with respect to local filters or frequency domain filters is the systematic use of all possible self-predictions in the image. In non-local means, the range kernel operates on patches centered around the pixel of interest. A "coarse" form of non-local means was considered, where a small patch neighborhood consisting of the pixels $u_1 \dots u_p$ (where, say, $u_1=0$) was used. In this case, the main observation was that formula for the non-local means can be written in terms of following sums:

$$\int_{\|y\| \leq R} f(x-y) g(f(x+u_1) - f(x-y+u_1), \dots, f(x+u_p) - f(x-y+u_p)) dy,$$

$$\int_{\|y\| \leq R} g(f(x+u_1) - f(x-y+u_1), \dots, f(x+u_p) - f(x-y+u_p)) dy,$$

where $g(s_1, \dots, s_p)$ is an anisotropic Gaussian in p variables, and has a diagonal covariance. This looks very similar to (2), except that we now have a multivariate range kernel. By using the separability of $g(s_1, \dots, s_p)$, and by approximating each Gaussian component by a $O(1)$ algorithm for computing (5) was developed.

Given a discrete noisy image $v = \{v(i) \mid i \in I\}$, the estimated value $NL[v](i)$, for a pixel i , is computed as a weighted average of all the pixels in the image,

$$NL[v](i) = \sum_{j \in I} w(i, j) v(j), \dots (6)$$

where the family of weights $\{w(i, j)\}_j$ depend on the similarity between the pixels i and j , and satisfy the usual conditions $0 \leq w(i, j) \leq 1$ and $\sum_j w(i, j) = 1$. The similarity between two pixels i and j depends on the similarity of the intensity gray level vectors $v(N_i)$ and $v(N_j)$, where N_k denotes square neighborhood of fixed size and centered at a pixel k . This similarity is measured.

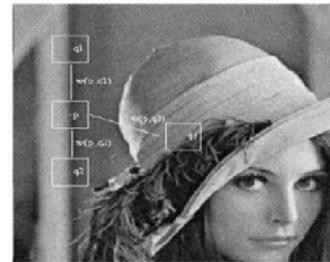


Fig 3: Scheme of Non Local Means strategy. Similar pixel neighborhoods give a large weight, $w(p,q1)$ and $w(p,q2)$, while much different neighborhoods give a small weight $w(p,q3)$ as a decreasing function of the weighted Euclidean distance

$$\|v(N_i) - v(N_j)\|_{2,a}^2$$

a , where $a > 0$ is the standard deviation of the Gaussian kernel. The application of the Euclidean distance to the noisy neighborhoods raises the following equality

$$E \|v(N_i) - v(N_j)\|_{2,a}^2 = \|u(N_i) - u(N_j)\|_{2,a}^2 + 2\sigma^2 \dots (7)$$

This equality shows the robustness of the algorithm since in expectation the Euclidean distance conserves the order of similarity between pixels. The pixels with a similar grey level neighborhood to $v(N_i)$ have larger weights in the average, see Figure 3.

These weights are defined as,

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}}$$

where $Z(i)$ is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}}$$

and the parameter h acts as a degree of filtering. It controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances. The NL-means not only compares the grey level in a single point but the the geometrical configuration in a whole neighborhood. This fact allows a more robust comparison than neighborhood filters. Figure 1 illustrates this fact, the pixel q_3 has the same grey level value of pixel p , but the neighborhoods are much different and therefore the weight $w(p, q_3)$ is nearly zero.

4. Fast O(1) Implementation Using Shiftable Kernels

For completeness, we now explain how the above kernels [8] can be used to compute (2) using O(1) operations. As observed in (3) and (4) are essentially the simplest kernels that have the so-called property of shiftability. This means that, for a given N , we can find a fixed set of basis functions $\phi_1(s), \dots, \phi_N(s)$ and coefficients c_1, \dots, c_N , so that for any translation, we can write

$$\phi(s - \tau) = c_1(\tau)\phi_1(s) + \dots + c_N(\tau)\phi_N(s) \dots (8)$$

The coefficients depend continuously on, but the basis functions have no dependence on. For (3), both the basis functions and coefficients are cosines, while they are polynomials for (4). This shiftability property is at the heart of the O(1) algorithm. Let $\bar{f}(x)$ denote the output of the Gaussian filter $g_{\sigma_s}(x)$ with neighborhood Ω ,

$$\bar{f}(x) = \frac{1}{\eta} \int_{\Omega} g_{\sigma_s}(x-y)f(y) dy \dots (9)$$

Note that, by replacing $g_{\sigma_s}(s)$ with $\phi(s)$, we can write (1) as

$$\tilde{f}(x) = \frac{1}{\eta} \left[\int_{\Omega} c_1(f(x))\bar{F}_1(x) + \dots + c_N(f(x))\bar{F}_N(x) \right] \dots (10)$$

where we have set $F_i(x) = \int_{\Omega} \phi_i(f(x))f(x) dx$. Similarly, by setting $G_i(x) = \int_{\Omega} \phi_i(f(x))f(x) dx$,

we can write

$$\eta = c_1(f(x))\bar{G}_1(x) + \dots + c_N(f(x))\bar{G}_N(x) \dots (11)$$

Now, it is well-known that certain approximation of (2.5) can be computed using just O(1) operations per pixel. These recursive algorithms are based on specialized kernels, such as the box and the hat function, and the more general class of box splines. Putting all these together, we arrive at the following O(1) algorithm for approximating (2):

- 1) Fix N , and approximate $g_{\sigma_s}(s)$ using (3) or (4).
- 2) For $i = 1, 2, \dots, N$, set up the images $F_i(x) = \int_{\Omega} \phi_i(f(x))f(x) dx$ and $G_i(x) = \int_{\Omega} \phi_i(f(x))f(x) dx$, and the coefficients $c_i(f(x))$.

- 3) Use a recursive O(1) algorithm to compute each $F_i(x)$ and $G_i(x)$.
- 4) Plug these into (10) and (11) to get the filtered image.

It is clear that better approximations are obtained when N is large. On the other hand, the run time scales linearly with N . One key advantage of the above algorithm, however, is that the $F_i(x)$ and $G_i(x)$ can be computed in parallel. For small orders ($N < 10$), the serial implementation is found to be comparable, and often better, than the state-of-the-art algorithms. The parallel implementation, however, turns out to be much faster than the competing algorithms, at least for $N < 50$. Henceforth, we will refer to the above algorithm as SHIFTABLE-BF, the shiftable bilateral filter.

5. Proposed Algorithm

In this project, we address the above problem, namely that N_0 grows as $O(T^2/\sigma_r^2)$ with σ_r . In Section, we propose a fast algorithm for determining T exactly. Besides cutting down N_0 , this is essential for determining the (local) dynamic range of a gray scale image that has been deformed, e.g., by additive noise. Setting $T = 255$ in this case can lead to artefacts in the processed image.

Next, this algorithm concentrated on simple and practical means of reducing the order, which leads to quite dramatic reductions in the run time of SHIFTABLE-BF. These modifications are also applicable to the shiftable algorithms proposed. Finally, this algorithm provides some experimental results to demonstrate the acceleration that is achieved using these modifications. We also compare our algorithm with the Porikli's [9] algorithms, both in terms of speed and accuracy.

6. Conclusion

As expected, Gaussian filtering performed poorly on all the test cases. The resulting images show little detail and still contain noise. The method noise for the Gaussian filter contained extensive structure and detail from the image. The Wiener filter performed marginally better than the Gaussian filter. More noise was removed by this method, but the images were still blurry. Once again the method noise contained detail and structure from the image. The non-local means method performed exceptionally well. As expected, the non-local means did a better job of preserving edges than the other methods. It performed best on periodic textures like the stripped pants from the Barb test case. In the Camera and Walter test cases where no noise was added, the denoised images looked clear and smooth. In all test cases for the non-local means method, the method noise contained little structure from the image. The non-local means algorithm accomplished its goals of removing noise and preserving detail.

7. Future Scope

This constant time algorithm can be extended to the color images. Adaptive Non Local Means algorithm can be used to improve the image quality.

Acknowledgement

I would like to express my sincere gratitude to my guide Sri D. Vishnu Vardhan, Garu Assistant Professor in Dept. of ECE, JNTUA College of Engineering Pulivendula for his continuous support and valuable guidance. I thank him for his meticulous efforts to guide me throughout my project work.

References

- [1] M.R. Banham and A.K. Katsaggelos, "Digital Image Restoration," IEEE Signal Processing Magazine, vol.14, no.2, pp.24-41, March 1997.
- [2] C. Tomasi, R. Manduchi, "Bilateral filtering for gray and color images", In Proc. International Conference on Computer Vision, 839846. 1998.
- [3] A. Buades, B. Coll, and J Morel. On image denoising methods. Technical Report 2004- 15, CMLA, 2004.
- [4] F. Porikli, "Integral histogram: a fast way to extract histograms in Cartesian spaces", In Proc. Computer Vision and Pattern Recognition, vol.1, 829-836, 2005.
- [5] B. Weiss, "Fast median and bilateral filtering", In Proc. SIGGRAPH, 2006.
- [6] A. Buades, B. Coll, and J Morel. A non-local algorithm for image denoising. IEEE International Conference on Computer Vision and Pattern Recognition, 2005.
- [7] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction,"IEEE Trans. Image Process., vol. 16, no. 2, pp. 349-366, Feb. 2007.
- [8] K. N. Chaudhury, Daniel Sage, and M. Unser, "Fast O(1) bilateral filtering using trigonometric range kernels," IEEE Transactions on Image Processing, vol. 20, no. 12, pp. 3376-3382, 2011.
- [9] F. Porikli, "Integral histogram: a fast way to extract histograms in Cartesian spaces", In Proc. Computer Vision and Pattern Recognition, vol.1, 829-836, 2005

Bibliography



D. Vishnu Vardhan working as a Assistant Professor in the department of ECE at JNTU college of Engineering, Pulivendula. He has 10 years of research and teaching experience in various domains. He is a life member of Indian Society for Technical Education (ISTE), India.



K. Jayachandra Reddy currently pursuing M.Tech, in Digital Electronics and Communication Systems at JNTUA College of engineering, Pulivendula, INDIA. His interested areas are Embedded systems and Image processing.