

Optimal fitting polynomial for linear time bilateral filters

Longquan Dai, Mengke Yuan and Xiaopeng Zhang[✉]

The bilateral filter (BF) has showed great effectiveness for a variety of problems. However, its brute-force implementation is time consuming. One way of accelerating a BF is to approximate the nonlinear range kernel of the BF by a set of linear time shiftable kernels. To achieve this goal, only finite values of the kernel of the BF have been used to perform smoothing due to the quantisation of digital images. Thus, the filtering results are not changed by substituting the range kernel with the function having the same values at finite discrete points. The Lagrange interpolation polynomial can exactly pass through predefined points and therefore can be employed to replace original kernels for accurate by accelerating the BF. To speed up the BF at the cost of small approximation error, two approximation methods are proposed to obtain the optimal fitting polynomial. The performance of the proposed method is validated by extensive experiments.

Introduction: The bilateral filter (BF) [1] is a valuable tool for many computer vision and graphic applications and its output at the pixel location $\mathbf{x} = (x, y)$ is given by

$$\hat{I}_x = \frac{\sum_{y \in \mathcal{N}_x^r} K_s(\|y - \mathbf{x}\|) K_r(I_y - I_x) I_y}{\sum_{y \in \mathcal{N}_x^r} K_s(\|y - \mathbf{x}\|) K_r(I_y - I_x)} \quad (1)$$

where \mathcal{N}_x^r is the square window centred at \mathbf{x} with radius r . For a BF, the kernels $K_s(\cdot)$ and $K_r(\cdot)$ determine how the spatial and intensity differences are treated. The most commonly used kernel in the literature is the Gaussian kernel $G_\sigma(z) = \exp(-z^2/2\sigma^2)$. Other unusual kernels are reported in [2]. Due to the signal-dependent range kernel $K_r(I_y - I_x)$, the BF becomes an edge-aware filter. On the flip side of the power, the complexity of the brute-force implementation is $O(|\mathcal{N}^r||I|)$ that is too intensive for time-critical applications, where $|I|$ is the number of pixels in the entire image and $|\mathcal{N}^r|$ the number of pixels in the window \mathcal{N}_x^r centred at \mathbf{x} . The inefficiency arises in $K_r(I_y - I_x)$ introduces nonlinearity to the BF and therefore rules out lots of fast calculation techniques. One solution is to approximate the nonlinear range kernel by a set of linear time shiftable kernels [2], where the linear time means the computational complexity is $O(|I|)$ which is independent of the window radius r . In the literature, existing accelerating methods of a BF roughly fall into two categories:

- **Series expansion-based BF:** This kind of method employs shiftable kernels [2] to represent the Gaussian range kernel. Chaudhury *et al.* [4] reduce the Gaussian kernel to the raised cosine range kernel. Dai *et al.* [3] propose Hermite polynomials to decompose the Gaussian kernel. Our method also belongs to this kind of method.
- **Histogram-based BF:** BF can be computed in linear time by the fast integral histogram if $K_s(\cdot)$ is the box filter. Zhang *et al.* [5] use the de Moivre-Laplace theorem to decompose the Gaussian kernel into M_s box functions. Gunturk [6] employs least-squares optimisation to find the optimal coefficients β_i^s that minimise the approximation error $(K_s(\|y - \mathbf{x}\|) - \sum_{i=1}^{M_s} \beta_i^s B_{\mathcal{N}_x^r}(\|y - \mathbf{x}\|))^2$. In contrast, Pan [7] exploited sparsity to determined the coefficients β_i^s .

Optimal fitting polynomial: Since the intensity values of digital images are quantised to N_r discrete values r_j , $1 \leq j \leq N_r$, only finite discrete points $K_r(r_j)$ of a BF's range kernel are used to compute the filtering results. We employ the least-squares fitting algorithm to find the optimal fitting polynomial that approximates to the range kernel $K_r(\cdot)$ with the smallest approximation error at the discrete points $(r_j, K_r(r_j))$ and replace the range kernel $K_r(\cdot)$ in (1) by the optimal polynomial. In this way, the nonlinear convolution of $K_r(\cdot)$ can be decomposed into a set of box filters. More importantly, we can compute the box filter in linear time by the fast histogram technique [7].

To obtain a versatile fitting polynomial $P_r(\cdot)$, it is reasonable to expect that the optimal $P_r(z) = \sum_{i=0}^{M_r} \beta_i^r z^i$ can exactly pass through some predefined points in the set $\{(r_j, K_r(r_j))\}$ while minimising the approximation error for the remaining points. Hence, we divide $\{(r_j, K_r(r_j))\}$ into two subsets, where the first L points denote the points being used to compute the approximation error and the last $N_r - L$

points represent the predefined constrain points. Considering the two subsets, we employ the constrained quadratic optimisation (2) to compute the coefficients β_i^r of the optimal fitting polynomial $P_r(\cdot)$

$$\begin{aligned} \min_{\beta_i^r} & \sum_{j=1}^L (K_r(r_j) - \sum_{i=0}^{M_r} \beta_i^r (r_j)^i)^2 \\ & \sum_{i=0}^{M_r} \beta_i^r (r_j)^i \geq 0, \quad 1 \leq j \leq L \\ \text{s.t.} & \sum_{i=0}^{M_r} \beta_i^r (r_j)^i = K_r(r_j), \quad L+1 \leq j \leq N_r \end{aligned} \quad (2)$$

The constraints $\sum_{i=0}^{M_r} \beta_i^r (r_j)^i \geq 0$ are added to prevent the negative fitting curve complained of by Chaudhury *et al.* [4]. Moreover, the optimisation can be efficiently solved by the Matlab function 'lsqin'.

Case 1: $L=0$. The optimal polynomial $P_r(\cdot)$ passes through all N_r points in the set $\{(r_j, K_r(r_j))\}$ if the degree M_r of the polynomial $P_r(\cdot)$ is no less than $N_r - 1$. Actually, the closed form solution is nothing but the Lagrange interpolation polynomial (3) which is capable of exactly passing through all N_r points in the set $\{(r_j, K_r(r_j))\}$.

$$P_r(z) = \sum_{i=1}^{N_r} K_r(r_i) \prod_{j=1, j \neq i}^{N_r} \frac{z - r_j}{r_i - r_j} \quad (3)$$

Although substituting $P_r(\cdot)$ with $K_r(\cdot)$ does not change the filtering result, adopting it to speed up the BF is not very efficient because the high degree polynomial increases the computation complexity. For instance, $r_j \in \{-255, \dots, 255\}$ for the 8-bit image. In this situation, the degree of $P_r(\cdot)$ is up to 510, which implies 510+ times linear filtering for accelerating the BF. One possible solution is to relax the exact equality constraints.

Case 2: $L=N_r$. The optimal polynomial does not prefer any special points, and the constrained optimisation (2) is free to decide the optimal polynomial $P_r(\cdot)$ with an arbitrary degree. Although the produced $P_r(r_j)$ is no longer equal to $K_r(r_j)$, the approximation error is usually very small. We can safely replace $K_r(r_j)$ by $P_r(r_j)$ without introducing significant numeric error. More importantly, the degree of $P_r(\cdot)$ will be very low compared with the Lagrange polynomial. However, when $L=N_r$, (2) is not appropriate to compute the optimal fitting curve for the non-smoothing range kernel $K_r(\cdot)$.

Case 3: $0 < L < N_r$. The optimal polynomial $P_r(\cdot)$ exactly passes through the last $N_r - L$ predefined points in the set $\{(r_j, K_r(r_j))\}$ with the smallest approximation error for the first L points in the set $\{(r_j, K_r(r_j))\}$, if the degree $M_r \geq N_r - L - 1$. Note that the optimal polynomial $P_r(\cdot)$ is a compromise for the first two cases. In the first case, $P_r(\cdot)$ produces accurate values at the cost of a high degree (i.e. high computational burden). Although the optimal $P_r(\cdot)$ in the second case achieves the lowest degree, its value is not accurate. The third case makes a compromise between the degree (efficiency) and the number of the accurate values (accuracy).

Frankly, all three cases have pros and cons. Users can choose different solutions according to different requirements. Here, we provide a unified framework that is able to trade off the efficiency and accuracy for different situations.

Our accelerating method: We employ the optimal polynomial $P_r(z) = \sum_{i=0}^{M_r} \beta_i^r z^i$ with degree M_r to approximate the arbitrary range kernel $K_r(\cdot)$ as $P_r(z)$ is able to linearise the nonlinear convolution of $K_r(\cdot)$. Then, substituting the nonlinear convolution of $K_r(\cdot)$ with $P_r(z)$, we have

$$\hat{I}_x = \frac{\sum_{i=0}^{M_r} \sum_{j=0}^i \varphi_{ij}(I_x) \sum_{y \in \mathcal{N}_x^r} K_s(\|y - \mathbf{x}\|) F_y^{ij}}{\sum_{i=0}^{M_r} \sum_{j=0}^i \varphi_{ij}(I_x) \sum_{y \in \mathcal{N}_x^r} K_s(\|y - \mathbf{x}\|)} \quad (4)$$

where $\varphi_{ij}(I_x) = \beta_i^r \binom{i}{j} (-I_x)^{i-j}$, $\psi_{ij}(I_y) = I_y^j$ and $F_y^{ij} = \psi_{ij}(I_y) I_y$. Once again, the most time-consuming operation in (4) is the linear convolution of $K_s(\cdot)$ because $\sum_{i=0}^{M_r} \sum_{j=0}^i$ is the pointwise sum operation.

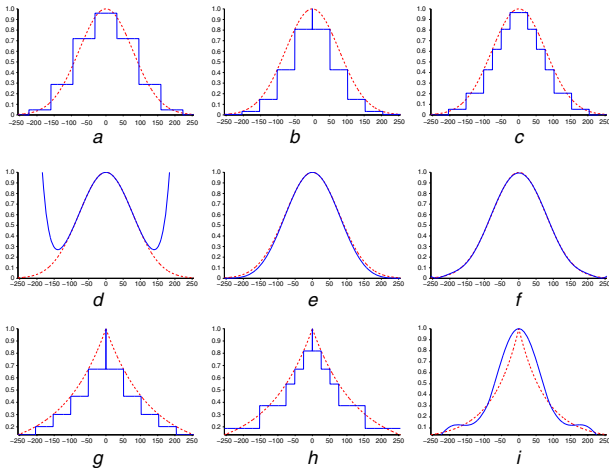


Fig. 1 Gaussian kernel approximations (first and second rows) and exponential decay kernel $\exp(-|x|/\sigma)$ approximations (third row), where variance $\sigma = 80$ a Zhang [5]; b Gunturk [6]; c Pan [7]; d Dai [3]; e Chaudhury [4]; f Ours; g Gunturk [6]; h Pan [7]; i Ours

In the literature, the Fast Fourier transform is exploited to fast compute the linear convolution. We can also employ Pan and He's multiple boxes approximation strategy [7] to compute the filtering results. According to Pan and He's method, we decompose the spatial kernel $K_s(\|\mathbf{y} - \mathbf{x}\|)$ into M_s box functions $K_s(\|\mathbf{y} - \mathbf{x}\|) \approx \sum_{i=1}^{M_s} \beta_i^s B_{N_x^i}(\|\mathbf{y} - \mathbf{x}\|)$, where $B_{N_x^i}(\|\mathbf{y} - \mathbf{x}\|) = 1$ when $\mathbf{y} \in N_x^i$; otherwise $B_{N_x^i}(\|\mathbf{y} - \mathbf{x}\|) = 0$. Then, replacing the nonlinear convolution by $\sum_{i=1}^{M_s} \beta_i^s B_{N_x^i}(\|\mathbf{y} - \mathbf{x}\|)$, we can obtain the fast equivalent filtering form (5)

$$\hat{I}_x = \frac{\sum_{i=0}^{M_r} \sum_{j=0}^i \varphi_{ij}(I_x) \sum_{k=1}^{M_s} \beta_k^s B_{F^{ij}}^{N_x^k}}{\sum_{i=0}^{M_r} \sum_{j=0}^i \varphi_{ij}(I_x) \sum_{k=1}^{M_s} \beta_k^s B_E^{N_x^k}} \quad (5)$$

The integral histogram technique can compute the box filtering results $B_{F^{ij}}^{N_x^k}$ and $B_E^{N_x^k}$ of F^{ij} and the matrix E of ones. Moreover, the sum operations $\sum_{i=0}^{M_r} \sum_{j=0}^i \cdot$, $\sum_{k=1}^{M_s}$ are pointwise. Hence, the overall computational complexity of (5) is $O(IJ)$.

Kernel approximation: BF accelerating methods decompose the BF into a set of linear time filters by substituting the original kernel in the BF with its approximation. Thereby, the accuracy of the approximation determines the accuracy of the final results. In the literature, Zhang *et al.* [5], Gunturk [6] and Pan [7] all resort to a set of box functions to approximate the Gaussian kernel. The major difference is that the three methods use different techniques to determine the radii r_i of box functions and the coefficients β_i^s . However, due to the non-smooth property, the box functions can only obtain rough approximations for the Gaussian kernel as illustrated in Fig. 1. In contrast, Dai *et al.* [3] and Chaudhury *et al.* [4] take advantage of the Hermite polynomials and raised cosines, respectively, to approximate the Gaussian kernel. Nevertheless, the success of the series expansion methods depends on the smoothness of the target kernel. For arbitrary non-smooth kernels, there is no guarantee for the existence of series expansion. Even worse, the approximation for the Gaussian kernel with small variance is not satisfactory. The approximation errors of both series expansion methods are significant using a low-order expansion, when the variance of the Gaussian kernel is small. The approximation using the box function is also not very good for the small variance. Since there are no special points on the curve of the smooth Gaussian kernel, we choose the second case in the above Optimal fitting polynomial Section to compute the optimal polynomial of the Gaussian kernel and illustrate the optimal polynomial with degree 6 for the 511 discrete points $\{(r_i, G_\sigma(r_i))\}$, $r_i \in \{-255, \dots, 255\}$ in Fig. 2.

The exponential decay kernel $\exp(-|x|/\sigma)$ is not differentiable at zero. The methods of Zhang *et al.* [5], Dai *et al.* [3] and Chaudhury *et al.* [4] cannot deal with this situation. Although the box function can be used to approximate the kernel in the frameworks proposed by Gunturk [6] and Pan [7], the approximation curve is still very rough. In contrast, employing the third case in the Optimal fitting polynomial Section, our method is able to produce an approximation polynomial exactly passing through the non-differentiable point as illustrated in Fig. 2.

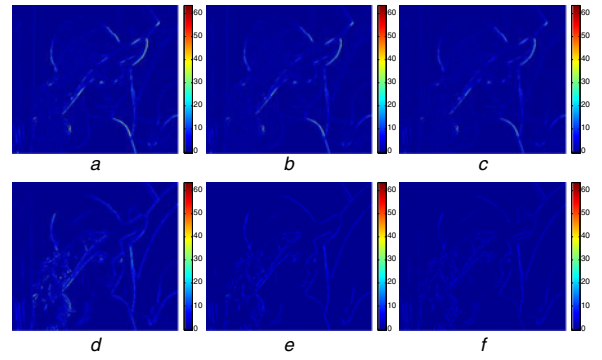


Fig. 2 Approximation error illustration. Spatial kernel and range kernel of prototype BF chosen as the Gaussian kernel and we assign $\sigma_r = 80$ for Gaussian range kernel

a Zhang [5]; b Gunturk [6]; c Pan [7]; d Dai [3]; e Chaudhury [4]; f Ours

Accuracy evaluation: In this Section, we directly compute the linear convolution of K_s and evaluate the PSNR index which reflects the accuracy of different nonlinear convolution decomposition methods to verify the approximation ability of our method. This is because the BF accelerating methods adopt various techniques to decompose the nonlinear convolution of the range kernel into a set of linear convolutions and employ linear time algorithms to fast compute the linear convolutions. If we take the brute-force method to compute linear convolution, the approximation error can only be caused by the nonlinear convolution decomposition method. To provide an intuitive illustration, we visualise the approximation error of the Lena image in Fig. 2. Both the spatial kernel and the range kernel of the prototype BF are chosen to be the Gaussian kernel $G_\sigma(\cdot)$. The results are obtained by naive implementation of the accelerating methods of Zhang *et al.* [5], Gunturk [6], Pan and He [7], Dai *et al.* [3], Chaudhury *et al.* [4] and ours without using the brute-force linear convolution. Let I^{bf} denote the result of the naive implementation of the BF and I^{ac} represent the result of the accelerating methods; we compute the absolute error by $|I^{bf} - I^{ac}|$ and illustrate the filtering results and the corresponding absolute error in Fig. 2. It is easy to see that our method achieves the smallest absolute error among the six comparative accelerating methods.

Conclusion: We propose a novel BF accelerating method. The major advantages of our method are that it is able to approach both differential and non-differential range kernels and our accelerating algorithm only needs addition and multiplication operations to calculate filtering results.

© The Institution of Engineering and Technology 2015

Submitted: 4 May 2015 E-first: 21 July 2015

doi: 10.1049/el.2015.1543

One or more of the Figures in this Letter are available in colour online.

Longquan Dai, Mengke Yuan and Xiaopeng Zhang (NLP, Institute of Automation Chinese Academy of Sciences, Beijing, People's Republic of China)

✉ E-mail: xiaopeng.zhang@ia.ac.cn

References

- Tomasi, C., and Manduchi, R.: 'Bilateral filtering for gray and color images'. Int. Conf. on Computer Vision, 1998
- Chaudhury, K.N.: 'Constant-time filtering using shiftable kernels', *IEEE Signal Process. Lett.*, 2011
- Dai, L., Yuan, M., and Zhang, X.: 'Accelerate bilateral filter using Hermite polynomials', *Electron. Lett.*, 2014
- Chaudhury, K.N., Sage, D., and Unser, M.: 'Fast o(1) bilateral filtering using trigonometric range kernels'. *Trans. Image Process.*, 2011
- Zhang, K., Lafuit, G., Lauwereins, R., and Van Gool, L.: 'Constant time joint bilateral filtering using joint integral histograms'. *Trans. Image Process.*, 2012, **21**, (9), pp. 4309–4314
- Gunturk, B.: 'Fast bilateral filter with arbitrary range and domain kernels'. *Trans. Image Process.*, 2011, **20**, (9), pp. 2690–2696
- Pan, S.D., An, X.J., and He, H.: 'Optimal o(1) bilateral filter with arbitrary spatial and range kernels using sparse approximation'. *Math. Probl. Eng.*, 2014