Polynomial Fitting for Period Prediction in Sliding-DCT-Based Filtering

Kazuya Ishikawa, Yuto Sumiya, and Norishige Fukushima

Nagoya Institute of Technology, Nagoya, Japan

ABSTRACT

Gaussian filtering (GF) is a fundamental smoothing filter that determines the weights in the kernel according to the Gaussian distribution. GF is an essential tool in image processing and is used in various applications. Therefore, accelerating GF is essential in various situations. The sliding DCT-based GF is one of the fastest methods for approximating GF. The Gaussian kernel is decomposed into multiple cosine kernels using the DCT transform and is approximated by the limited number of kernels. When calculating the period of the DCT for fitting the best length, a linear search method is used; however, the brute-force search has a significant impact on the filtering processing time. In this paper, we accelerate the period estimation by polynomial fitting. Experimental results show that the proposed method has almost the same accuracy as the brute-force approach.

Keywords: Gaussian Filter, Sliding DCT, Period estimation, Polynomial fitting

1. INTRODUCTION

Gaussian filtering (GF) is an essential tool for image processing and is adopted in various applications: pre-filtering, scale-space analysis,[1–4] feature description,[5] saliency map,[6] internal processing of edge-preserving filters (accelerated bilateral filtering,[7–14] guided image filtering,[15,16] and high dimensional Gaussian filtering[17,18]) image quality indices of structural similarity index measure (SSIM),[19,20] blur removal processing,[21–23] and high-dynamic range imaging (HDR).[24] Hence, the accelerated GF is of practical significance for various tasks.

The computational order of GF for the straightforward implementation is $O(R^2)$ per pixel, where $R \in \mathbb{N}$ is the kernel radius, and the implementation is identical to 2D finite impulse response (FIR) filtering. Separable FIR filtering and frequency domain filtering are well-known techniques to speed up the filtering process. The separable FIR filtering reduces GF's order to $O(R)$, and the frequency domain filtering also reduces it to $O(\log N)$, where $N \in \mathbb{N}$ is the number of image pixels. Note that the frequency domain filtering is required to transform an image and the Gaussian kernel to frequency components by a fast Fourier transform (FFT) or discrete cosine transform (DCT); the transformation becomes offset processing.

The sliding-DCT-based GF is a state-of-the-art method.[25–27] The order of the filter is $O(K)$, where $K \in \mathbb{N}$ is the approximated order, which is independent of R. The sliding transform can be realized by various DCT definitions, such as DCT-1 to DCT-8. In particular, the DCT definitions with odd numbers (DCT-1,3,5,7) can be used to approximate the Gaussian kernel.[26] We show the sliding DCT-based GF:

$$g(\sigma, x) \approx \hat{g}_{\text{type}}(\sigma, x, K, T) = \alpha_{\sigma,0} + 2\sum_{k=1}^{K} \alpha_{\sigma,k} \cos\left(\frac{2\pi}{T}kx\right), \tag{1}$$

where g is the convolution kernel and $\hat{g}_{\text{type}}$ is its approximated one of DCT-1, 3, 5, or 7 with the order K. $a_{\sigma,n}$ is the transformed coefficient for each σ and k. Here, we need frequency period T for each DCT type, order, and σ; thus, T is computed by changing these parameters. However, the period computation is cost consuming. Sugimoto et al. [9] proposed a closed-form solution, which is fast but still has room for accuracy improvement reported in their following paper.[26]

This paper solves the limitation of the accuracy and speed of the GF. The contributions is as follows:

- We propose an accurate sliding-DCT-based GF using the optimized period T by using brute-force search.
- Also, we propose its accelerated method by using polynomial fitting.

Further author information: (Send correspondence to N. Fukushima)
N. Fukushima: Web: https://fukushima.web.nitech.ac.jp/en/

2. SLIDING DCT-BASED GAUSSIAN FILTERING

The multi-dimensional GF can be decomposed into multiple one-dimensional GFs due to its separable property. Therefore, we introduce the procedure for sliding DCT-based GF in the one-dimensional case. First, the definition of a Gaussian kernel is expressed as follows:

$$g_n = \eta^{-1} \exp\left(-\frac{n^2}{2\sigma^2}\right), \qquad \eta = \sum_{n=-R}^{R} \exp\left(-\frac{n^2}{2\sigma^2}\right), \tag{2}$$

where σ is the standard deviation of the Gaussian distribution, and R is the filtering radius. The computational order is $O(R)$. Usually, $R = \lceil 3\sigma \rceil$ or $\lceil 6\sigma \rceil$. The FIR convolution of the kernel is defined as:

$$(f * g)_x = \sum_{n=-R}^{R} f_{x+n} g_n, \tag{3}$$

where f_x $(f_x \in \mathcal{R} \subset \mathbb{R}, x \in \mathcal{X} \subset \mathbb{N})$ is input signals, and g_n $(n = -R, \dots, R)$ is Gaussian kernel weights. Note that $\mathcal{R} = \{0, \dots, 255\}$ is a dynamic range, and $\mathcal{X}$ is image area number. We can transform the Gaussian kernel to the linear summation of cosine function by DCT since the Gaussian function is an even function:

$$g_n = \sum_{k=0}^{R} G_k C_n^{(k)}, \quad C_n^{(k)} = \cos(\phi(k + k_0)(n + n_0)), \tag{4}$$

where G_k $(k = 0, 1, \dots, R)$ is the DCT transformed coefficient from g_n, and $\phi = \frac{2\pi}{T}$. Eq. (4) is the inverse DCT expression of the kernel. The variables T, k_0, and n_0 depend on the type of DCT that are summarized in Tab. 1. Note that DCT-1 and -5 have DC components because $k_0 = 0$. Based on (2), G_k can be approximated as:

$$G_k \simeq \frac{c_k e^{-\frac{1}{2}\sigma^2\phi^2(k+k_0)^2}}{\sum_{k=0}^{K} c_k e^{-\frac{1}{2}\sigma^2\phi^2(k+k_0)^2}}, \qquad c_k = \begin{cases} 1 & (k = 0) \\ 2 & (otherwise) \end{cases} \tag{5}$$

Substituting (4) into (3) yields:

$$(f * g)_x = \sum_{n=-R}^{R} \sum_{k=0}^{R} f_{x+n} G_k C_n^{(k)} = \sum_{k=0}^{R} G_k F_k^{(x)}, \tag{6}$$

$$F_k^{(x)} = \sum_{n=-R}^{R} f_{x+n} C_n^{(k)}, \tag{7}$$

where $F_k^{(x)}$ is a DCT coefficient of the partial signals extracted from the window at the coordinate x of input pixel value. Formula (6) indicates that the output value of convolution for coordinate x is the inner product between the partial signals extracted from the window and the kernel, which is equal to the inner product between the transform coefficient of the kernel and the transform coefficient of input signals.

The spatial filtering is accelerated by truncating the DCT coefficients, i.e., setting the approximation order $K (\leq R)$. The approximation is defined as follows:

$$(f * g)_x = \sum_{k=0}^{R} G_k F_k^{(x)} \simeq \sum_{k=0}^{K} G_k F_k^{(x)}. \tag{8}$$

Table 1. Parameters of $T, k_0, n_0$for DCT-1, 3, 5, 7.

DCT-type	T	k_0	n_0
DCT-1	$2R$	0	0
DCT-3	$2R+2$	$\frac{1}{2}$	0
DCT-5	$2R+1$	0	0
DCT-7	$2R+1$	$\frac{1}{2}$	0

The computational order of $F_k^{(x)}$ is $O(R)$ for each k; thus, the total computational order of (8) is $O(KR)$, which is larger than the naïve 1D GF, which is $O(R)$. The sliding transform can calculate $F_k^{(x)}$ in $O(1)$ order;[25] thus, we can reduce the order from $O(KR)$ to $O(K)$.

When $K = \infty$, the period is ∞; however, the optimal value is shorter in the approximated computation. Sugimoto et al. minimize the following closed-form to obtain T for DCT5:

$$E(T; K, \sigma) = \mathrm{erfc}(\frac{2T+1}{2\sigma}) + \mathrm{erfc}(\pi\sigma\frac{2K+1}{2T+1}) \tag{9}$$

$$T_{K,\sigma,\mathrm{type}} = \arg\min_t E(t; K, \sigma, \mathrm{type}), \tag{10}$$

where type = 5. The solution uses approximation for the error function; thus, the accuracy tends to decrease.

3. PROPOSED METHOD

In this paper, we directly minimize the difference between the kernel and the approximation function. The period T is computed to minimize the following:

$$E(T; K, \sigma, \mathrm{type}) = \frac{1}{|\mathrm{R}|}\sum_{\mathrm{r}\in\mathrm{R}}(\mathrm{g}(\sigma, \mathrm{r}) - \hat{\mathrm{g}}_{\mathrm{type}}(\sigma, \mathrm{r}, \mathrm{K}, \mathrm{T}))^2, \tag{11}$$

Given type, K, σ, we compute the error of (11) to linearly change t for the brute-force search (Eq. (10)). Figure 1 shows the difference between the period T obtained by Eq. (9), (10) and various coefficients of σ for R. Since the response repeats in cycles, we need a long period enough to cover the Gaussian tail. However, there is a trade-off because the longer period is the more difficult to represent the response with fewer coefficients.

Eq. (11) does not have the closed-form solution without approximation; thus, we full search the candidates, but it is cost-consuming. For acceleration, we approximate the search by polynomial fitting for two variables: K and σ. The two-variable-fitting is usually complex; however, we can use the relationship between radius and σ. For usual FIR filtering, we set $r = \lceil a\sigma \rceil, a \in \mathbb{R}$, i.e., the radius is linear to σ. Directly using the fact, the N-th order polynomial fitting is as follows:

$$T' = \lfloor \sigma \sum_{n=0}^{n=N}(c_n K^n) + 0.5 \rfloor \tag{12}$$

where $c_n, n \in \{0, 1, \ldots, N\}$ are polynomial coefficients, and the coefficients are obtained by solving a Vandermonde matrix. The rounding value is used for the integer radius.

Furthermore, we can add an offset term b for more accuracy.

$$T' = \lfloor \sigma \sum_{n=0}^{n=N}(c_n K^n) + b + 0.5 \rfloor \tag{13}$$

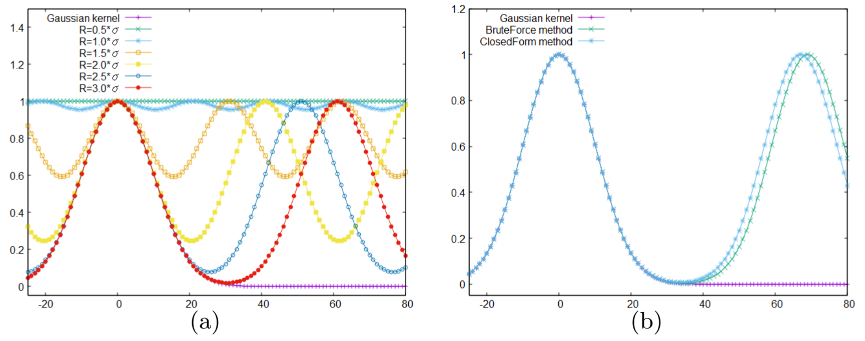


Figure 1. Gaussian kernel approximated by DCT-5 when σ is 10 and the approximation order K is 3. (a) Approximation by DCT-5 when the coefficient for σ of R in the period T is changed from 0.5 to 3. (b) The purple color represents the Gaussian kernel when σ is 10 in Eq. (2), the green color uses the period T obtained by the Eq. (10), and the blue color also obtained by the Eq. (9).

We can obtain these fitting results by solving the following Vandermonde-like matrix, $\boldsymbol{V}\boldsymbol{C} = \boldsymbol{T}$:

$$\begin{bmatrix} 1 & \sigma_0 & x_{0,0} & x_{0,0}^2 & \cdots & x_{0,0}^N \\ 1 & \sigma_1 & x_{1,0} & x_{1,0}^2 & \cdots & x_{1,0}^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_\ell & x_{\ell,0} & x_{\ell,0}^2 & \cdots & x_{\ell,0}^N \\ 1 & \sigma_0 & x_{0,1} & x_{0,1}^2 & \cdots & x_{0,1}^N \\ 1 & \sigma_1 & x_{1,1} & x_{1,1}^2 & \cdots & x_{1,1}^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_\ell & x_{\ell,1} & x_{\ell,1}^2 & \cdots & x_{\ell,1}^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_0 & x_{0,m} & x_{0,m}^2 & \cdots & x_{0,m}^N \\ 1 & \sigma_1 & x_{1,m} & x_{1,m}^2 & \cdots & x_{1,m}^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_\ell & x_{\ell,m} & x_{\ell,m}^2 & \cdots & x_{\ell,m}^N \end{bmatrix} \begin{bmatrix} b \\ c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} T_{k_0,\sigma_0,\text{type}} \\ T_{k_0,\sigma_1,\text{type}} \\ \vdots \\ T_{k_0,\sigma_\ell,\text{type}} \\ T_{k_1,\sigma_0,\text{type}} \\ T_{k_1,\sigma_1,\text{type}} \\ \vdots \\ T_{k_1,\sigma_\ell,\text{type}} \\ \vdots \\ T_{k_m,\sigma_0,\text{type}} \\ T_{k_m,\sigma_1,\text{type}} \\ \vdots \\ T_{k_m,\sigma_\ell,\text{type}} \end{bmatrix}, \tag{14}$$

where $x_{\ell,m} = \sigma_\ell k_m$; $k_m \in \{1, 2, \ldots, K_{\max}\}$ $\sigma_\ell = \frac{\sigma_{max} - \sigma_{\min}}{\delta}\ell + \sigma_{\min}$. $K_{\max}$ is the max value of in fitting range of K. For σ, we resample the continuous value of σ by interval δ in the possible min/max range: $\sigma_{\min}$ and $\sigma_{\max}$. The size of the matrices are as follow: $\boldsymbol{V}^{(N+2)\times\delta K_{\max}}$, $\boldsymbol{C}^{1\times(N+2)}$, and $\boldsymbol{T}^{1\times\delta K_{\max}}$. For Eq. (12), we can solve the matrix by removing the elements of 1 and b from the matrix. Note that we can use only the resulting coefficients for filtering; thus, we can omit the computation of the matrix solving.

4. EXPERIMENTAL RESULTS

In our experiment, we verified the effectiveness of the proposed method. We changed the truncated parameter: $1 \leq K \leq 15$ sampled at the interval of 1, standard deviation: $2.1 \leq \sigma \leq 10$ sampled at intervals of 0.1. Also, we changed the DCT types: 1, 3, 5, and 7. The tested image size was 512×512. The tested code was optimized by AVX and compiled by Visual Studio 2019 on AMD Threadripper 3970X.

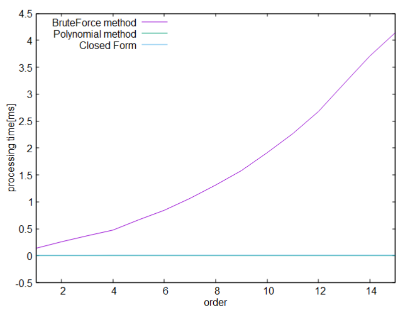
Table 2 shows the sum of absolute difference (SAD) of all samples of K and σ, $\sum_{K,\sigma} |T_{\sigma,K} - T'_{\sigma,K}|$, and the maximum absolute difference (MAD) of $|T_{\sigma,K} - T'_{\sigma,K}|$ in Eq. (12) and Eq. (13) after fitting with $N = 7$. The approximation with offset is more accurate than without offset, and our method can predict radius within floating rounding error, i.e., ± 1.

Figure 2 (a) shows the computation time of the predictions at $\sigma = 10$. The brute-force search takes 4.38ms at the order 15, while our method requires only a few tens of clock cycles, i.e., it is on the order of nanoseconds. Figure 2 (b) shows the processing time of filtering, including the prediction. The previous GF at the order 15 and $\sigma = 10$ takes 5.09ms, while the proposed method takes 0.821ms; thus, the brute-force search takes longer than filtering itself. Figure 2 (c) shows the PSNR for evaluating the approximation accuracy. The proposed and brute-force methods have higher accuracy than the closed-form solution. Also, the proposed method has almost the same processing speed as the closed-form solution.

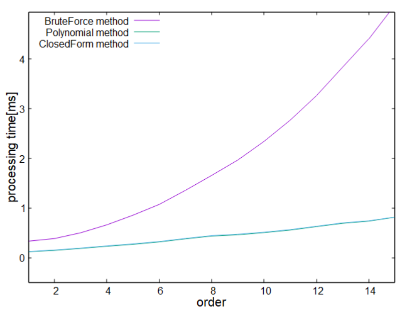
Figure 3 shows that predicting result for each parameter with equation (13). We can predict the same value in most cases, and even if the predicted value deviates from the correct one, the difference between the two values is so slight that it does not significantly affect the output image.

Table 2. Errors with and without offset ($N = 7$) for each DCT type.

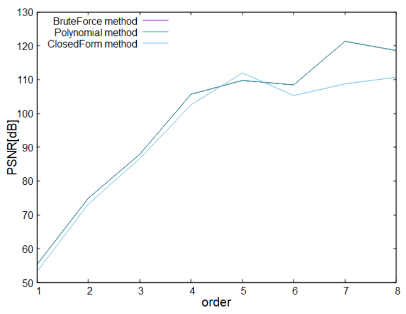
		DCT 1	DCT 3	DCT 5	DCT 7
without offset (12)	SAD	53	286	215	226
	MAD	1	1	1	1
with offset (13)	SAD	46	61	68	81
	MAD	1	1	1	1



(a) time: period computing

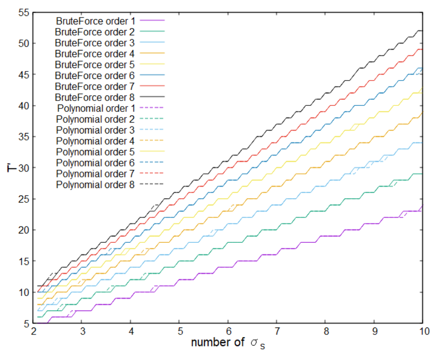


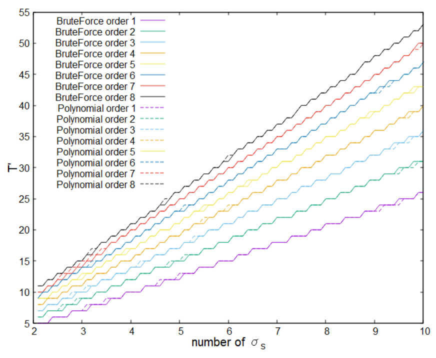
(b) time: filtering

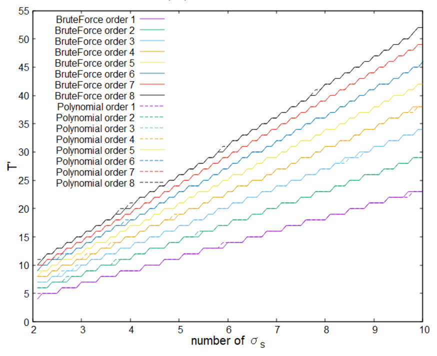


(c) PSNR

Figure 2. (a) processing time of only predicting T. (b) the filtering processing time including the predicting T;the lines of the closed-form and proposed are covered for (a) and (b). (c) PSNR for each method; the lines of the brute-force and proposed are covered. (DCT-5, $\sigma = 10$ for each plot).


(a) DCT-1


(b) DCT-3


(c) DCT-5

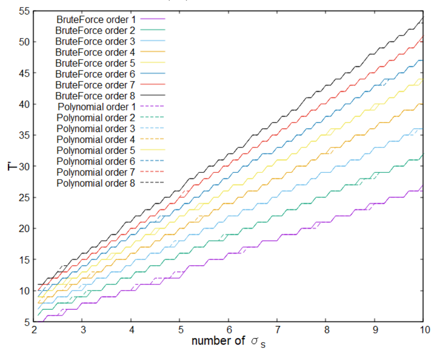

(d) DCT-7

Figure 3. Predicting results for each type and parameters σ_s in $K = \{1, 2, \ldots, 8\}$.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI (21H03465).

REFERENCES

[1] Witkin, A., "Scale-space filtering: A new approach to multi-scale description," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **9**, 150–153 (1984).

[2] Witkin, A. P., "Scale-space filtering," in *Readings in Computer Vision*, 329–332, Elsevier (1987).

[3] Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M., "Pyramid methods in image processing," *RCA engineer* **29**(6), 33–41 (1984).

[4] Lindeberg, T., "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of applied statistics* **21**(1-2), 225–270 (1994).

[5] Lowe, D., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)* **60(2)**, 91–110 (2004).

[6] Itti, L., Koch, C., and Niebur, E., "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis Machine Intelligence* **20**, 1254–1259 (1998).

[7] Chaudhury, K. N., "Acceleration of the shiftable o(1) algorithm for bilateral filtering and nonlocal means," *IEEE Transactions on Image Processing* **22**, 1291–1300 (2013).

[8] Chaudhury, K. N., Sage, D., and Unser, M., "Fast o(1) bilateral filtering using trigonometric range kernels," *Proc. IEEE Transactions on Image Processing* **20**(12), 3376–3382 (2011).

[9] Sugimoto, K. and Kamata, S., "Compressive bilateral filtering," *IEEE Transactions on Image Processing* **24**(11), 3357–3369 (2015).

[10] Sugimoto, K., Fukushima, N., and Kamata, S., "200 fps constant-time bilateral filter using svd and tiling strategy," in *Proc. IEEE International Conference on Image Processing (ICIP)*, (2019).

[11] Yang, Q., Tan, K. H., and Ahuja, N., "Real-time o(1) bilateral filtering," in *Proc. IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, (2009).

[12] Deng, G., "Fast compressive bilateral filter," *Electronics Letters* **53**, 150–152 (2017).

[13] Sumiya, Y., Fukushima, N., Sugimoto, K., and Kamata, S., "Extending compressive bilateral filtering for arbitrary range kernel," in *Proc. IEEE International Conference on Image Processing (ICIP)*, (2020).

[14] Paris, S. and Durand, F., "A fast approximation of the bilateral filter using a signal processing," in *Proc. European conference on computer vision (ECCV)*, (2006).

[15] K. He, J. S. and Tang, X., "Guided image filtering," in *Proc European Conference on Computer Vision (ECCV)*, (2010).

[16] Fukushima, N., Sugimoto, K., and Kamata, S., "Guided image filtering with arbitrary window function," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2018).

[17] Nair, P. and Chaudhury, K. N., "Fast high-dimensional kernel filtering," *IEEE Signal Processing Letters* **26**, 377–381 (2019).

[18] Miyamura, T., Fukushima, N., Waqas, M., Sugimoto, K., and Kamata, S., "Image tiling for clustering to improve stability of constant-time color bilateral filtering," in *Proc. IEEE International Conference on Image Processing (ICIP)*, (2020).

[19] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004).

[20] Sasaki, T., Fukushima, N., Maeda, Y., Sugimoto, K., and Kamata, S., "Constant-time gaussian filtering for acceleration of structure similarity," in *Proc. International Conference on Image Processing and Robotics (ICIPRoB)*, (2020).

[21] Richardson, W. H., "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America* **62**(1), 55–59 (1972).

[22] Lucy, L. B., "An iterative technique for the rectification of observed distributions," *The Astronomical Journal* **79**, 745 (1974).

[23] Irani, M. and Peleg, S., "Improving resolution by image registration," *Graphical models and image processing* **53(3)**, 231–239 (1991).

[24] Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J., "Photographic tone reproduction for digital images," in *Proc. Annual Conference on Computer Graphics and Interactive Techniques*, 267–276 (2002).

[25] Sugimoto, K. and Kamata, S., "Fast gaussian filter with second-order shift property of dct-5," in *Proc. IEEE International Conference on Image Processing (ICIP)*, (2013).

[26] Sugimoto, K., Kyochi, S., and Kamata, S., "Universal approach for dct-based constant-time gaussian filter with moment preservation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2018).

[27] Otsuka, T., Fukushima, N., Maeda, Y., Sugimoto, K., and Kamata, S., "Optimization of sliding-dct based gaussian filtering for hardware accelerator," in *Proc. International Conference on Visual Communications and Image Processing (VCIP)*, (2020).