

# UNIVERSAL APPROACH FOR DCT-BASED CONSTANT-TIME GAUSSIAN FILTER WITH MOMENT PRESERVATION

Kenjiro Sugimoto<sup>†</sup>, Seisuke Kyochi<sup>‡</sup> and Sei-ichiro Kamata<sup>†</sup>

<sup>†</sup> Waseda University, 2-7 Hibikino, Wakamatsu, Kitakyushu, Fukuoka, 808-0135, Japan

<sup>‡</sup> The University of Kitakyushu, 1-1 Hibikino, Wakamatsu, Kitakyushu, Fukuoka, 808-0135, Japan

## ABSTRACT

This paper presents a universal approach for constant-time Gaussian filters ( $O(1)$  GF) based on the Discrete Cosine Transform (DCT). It is well known that DCT has the eight types of definitions. Existing methods of  $O(1)$  GF use difference DCT type according to their original concepts. However, all types of DCT have not been studied comprehensively and quantitatively. Unlike existing methods, the proposed approach covers all types of DCT and moment preservation for arbitrary orders, which enables us to clarify differences of  $O(1)$  GF derived from each DCT through a comprehensive analysis. Based on the universal approach, a closed-form solution to optimize weight coefficients is also proposed based on a simple convex analysis. Experiments found that DCT-7 shows the highest approximate accuracy, which is a new conclusion different from existing methods.

**Index Terms**— constant-time Gaussian filter, discrete cosine transform, sliding transform, moments

## 1. INTRODUCTION

Gaussian filter (GF) is one of the fundamental tools in image processing, computer vision and computer graphics. It still has been used in many modern applications of object recognition [1], visual saliency [2], edge-preserving smoothing [3, 4, 5, 6] and so on [7]. These applications generally apply GF to many images, such as video processing, or apply GF to an image many time using a variety of parameter values, such as scale space analysis. These scenarios have demanded to reduce the computational complexity of GF if feasible. In particular, the computational time depending on filter window size is a significant problem for high-resolution image processing, which relatively use a large filter window.

An efficient solution to the large-window problem is *constant-time* GF ( $O(1)$  GF) where constant-time means computational complexity does not depend on filter window size, i.e., it runs in  $O(1)$  time/pixel. The general framework of  $O(1)$  GF is that it approximates a Gaussian kernel by a combination of few efficiently-computable subkernels and then convolves each subkernel in  $O(1)$  time/pixel. Thus,  $O(1)$  GF shows tradeoff between computational complexity and approximate accuracy as a major performance factor. We aim at achieving higher performance tradeoff. Many methods of  $O(1)$  GF have been proposed in the past and widely-known ones in them can be categorized into two groups: recursive approximation [8, 9, 10, 11] and truncated cosine approximation [12, 13, 14, 15]. We discuss the latter because they have so far achieved the state-of-the-art performance in this sense.

This work was supported by JSPS KAKENHI (Grant Number: JP16K16092, JP17H01764).

**Table 1.** Coverage of DCT types and moment preservation.

Method	DCT types	Preserved
Elboher and Werman [12]	DCT-1	
Sugimoto and Kamata [14]	DCT-5	$\mu_0$
Charalampidis [15]	DCT-3	$\mu_0, \mu_2$
<b>Ours</b>	DCT-1,3,5,7	$\mu_0, \mu_2, \mu_4, \dots$

All the cosine-based methods share the following framework. They approximate a (truncated) Gaussian kernel by a linear sum of (truncated) cosine terms and then convolve each cosine term in  $O(1)$  time/pixel by efficient techniques. The approximate kernel is substantially obtained via the Discrete Cosine Transform (DCT) but a variety of DCT types have been used as Table 1 lists. Existing methods conceptually differ in several ways including which DCT type used, how to derive weight coefficients and how to convolve cosine terms. For instance, Elboher and Werman [12] employed DCT-1 for kernel approximation and integral images [16, 17] for cosine convolution. Sugimoto and Kamata [14, 5] accelerated this method by using DCT-5 and sliding transform instead. Charalampidis [15] derived relationship of recursive approximation to truncated cosine approximation using DCT-3,5 and then remarked that DCT-3 produced more accurate Gaussian approximation than DCT-5. Moreover, they indicated that the DCT-5 approach distorted isotropy of two-dimensional GF and reduced this problem by deriving weight coefficients that preserved sum and variance of Gaussian kernel before-and-after approximation by DCT-3. Thus, the existing methods have been designed based on a variety of concepts. However, there exist no comprehensive analysis and quantitative experiments that can cover all types of DCT and arbitrary moments.

This paper presents a universal approach in truncated cosine approximation that covers all types of DCT and arbitrary moments such as sum, variance and so on. We also perform quantitative analysis through our universal model. We formalize a Gaussian kernel approximated by arbitrary DCT type as a general form and comprehensively derive a close-form solution to obtain its optimal weight coefficients that can preserve arbitrary moments. This solution provides more clear understanding and slightly-lower computational complexity than that of [15] because it is derived by a simple convex optimization approach. Unlike [15], our experiments validate the fact that DCT-7 is the most suitable in approximate accuracy for GF especially when scale parameter is larger.

## 2. EXISTING WORK

This section describes Gaussian filter and moments of Gaussian, followed by briefly introducing DCT-based  $O(1)$  GF and its remaining problems. We discuss one-dimensional GF throughout the paper but

**Table 2.** A parameter list of all types of DCT.

DCT Type	$T$	$k_0$	$n_0$
DCT-1 (Inverse DCT-1)	$2N - 2$	0	0
DCT-2 (Inverse DCT-3)	$2N$	0	$1/2$
DCT-3 (Inverse DCT-2)	$2N$	$1/2$	0
DCT-4 (Inverse DCT-4)	$2N$	$1/2$	$1/2$
DCT-5 (Inverse DCT-5)	$2N - 1$	0	0
DCT-6 (Inverse DCT-7)	$2N - 1$	0	$1/2$
DCT-7 (Inverse DCT-6)	$2N - 1$	$1/2$	0
DCT-8 (Inverse DCT-8)	$2N + 2$	$1/2$	$1/2$

it is noting worth that multi-dimensional isotropic GF is separable into a multiple of one-dimensional GFs.

### 2.1. Gaussian filter

Let  $x_t \in \mathbb{R}$  ( $t \in \mathbb{Z}$ ) be an input sequence and  $h_n \in \mathbb{R}$  ( $n = -N + 1, \dots, N - 1$ ) a truncated filter kernel where  $N \in \mathbb{N}$  is called window radius. The convolution is defined by

$$(x * h)_t := \sum_{n=-N+1}^{N-1} x_{t+n} h_n. \quad (1)$$

We define the kernel as Gaussian kernel (sampled Gaussian [18]):

$$h_n := \eta^{-1} e^{-\frac{n^2}{2\sigma^2}}, \quad \eta := \sum_{n=-N+1}^{N-1} e^{-\frac{n^2}{2\sigma^2}}, \quad (2)$$

where  $\sigma \in \mathbb{R}_+$  is a scale parameter. The window radius  $N$  has to be determined such that it supports most of Gaussian shape, e.g., commonly  $N = \lceil 3\sigma \rceil$ . Since the computational complexity of (1) is  $O(N)$  time/pixel, the larger  $\sigma$  obviously causes the longer running time. This property is a significant problem for high-resolution image processing, which requires relatively-large  $\sigma$ .

### 2.2. Moments of Gaussian

An important measure for Gaussian function is *moments*. The  $m$ -th order moment of discrete kernel  $h_n$  is defined by

$$\mu_m := \sum_{n=-N+1}^{N-1} n^m h_n. \quad (3)$$

Low-order moments are particularly named *sum*  $\mu_0$ , *mean*  $\mu_1$ , *variance*  $\mu_2$ , *skewness*  $\mu_3$  and *kurtosis*  $\mu_4$ . In the case of Gaussian kernel (2),  $\mu_{2m+1} = 0$  ( $m = 0, 1, \dots$ ) holds thanks to its symmetry. We therefore deal with only  $\mu_{2m}$  in later discussion. It is well-known that ideal continuous Gaussian kernel exactly satisfies  $\mu_0 = 1$ ,  $\mu_2 = \sigma^2$ ,  $\mu_4 = 3\sigma^4$  and so on; by contrast, the discrete Gaussian kernel (2) perturbs the moment values due to discretization and window truncation. As [15] also pointed out, moment preservation is important in image filtering. For instance, if  $\mu_0 \neq 1$ , the filter would change brightness of images and the subjective quality drastically would degrade.

### 2.3. $O(1)$ Gaussian filter using DCT and its remaining problems

A state-of-the-art solution to the large-window problem is  $O(1)$  GF using DCT. Gaussian kernel (2) can be well-approximated by a linear sum of few cosine terms via DCT because it is an even

**Table 3.** Metric matrices for each DCT type.

DCT Type	$M_n \in \mathbb{R}^{N \times N}$	$M_k \in \mathbb{R}^{N \times N}$
DCT-1	$\text{diag}(\frac{1}{2}, 1, \dots, \frac{1}{2})$	$\text{diag}(2, 1, \dots, 2)$
DCT-2	$\text{diag}(\frac{1}{2}, 1, \dots, 1)$	$I$
DCT-3	$I$	$\text{diag}(2, 1, \dots, 1)$
DCT-4	$I$	$I$
DCT-5	$\text{diag}(\frac{1}{2}, 1, \dots, 1)$	$\text{diag}(2, 1, \dots, 1)$
DCT-6	$\text{diag}(1, \dots, 1, \frac{1}{2})$	$\text{diag}(2, 1, \dots, 1)$
DCT-7	$\text{diag}(\frac{1}{2}, 1, \dots, 1)$	$\text{diag}(1, \dots, 1, 2)$
DCT-8	$I$	$I$

function whose spectra decay exponentially. As mentioned in detail later, each cosine term can be convolved in  $O(1)$  time/pixel by sliding transform [19]. Hence, this overall approach runs in  $O(1)$  time/pixel, that is independent of window radius  $N$ .

This paper sheds light on the following two remaining problems (see Table 1). i) Although the existing methods use different types of DCT, it is still unclear which in all types of DCT achieves the highest performance in terms of a quantitative measure. Actually, each existing method has argued just one type of DCT. ii) Many state-of-the-art methods except for [15] have not attempted moment preservation explicitly. Moment preservation is strongly demanded in DCT-based  $O(1)$  GF because the cosine approximation basically distorts moments much more than the truncated Gaussian kernel (2). It is also important for a fair approximate accuracy comparison among all types of DCT. Regarding these two points, [15] stated that variance  $\mu_2$  is also important for preserving the isotropy of two-dimensional GF. However, their discussion only covered how to preserve  $\mu_0$  and  $\mu_2$  for DCT-3. In order to reach to more in-depth understanding, we should discuss the above two points from a more general viewpoint.

## 3. PROPOSED UNIVERSAL APPROACH

We present a universal approach for DCT-based  $O(1)$  GF that covers all types of DCT and supports moment preservation (see Table 1).

### 3.1. General form of DCT and its sliding transform

Gaussian kernel is an even function whose spectra exponentially decay to zero. From this fact, we consider to approximate the Gaussian kernel (2) by a linear sum of  $K$  cosine terms:

$$h_n \approx \sum_{k=0}^{K-1} \hat{h}^{(k)} \cos\left(\frac{2\pi}{T}(k+k_0)(n+n_0)\right) \quad (4)$$

where  $\hat{h}^{(k)} \in \mathbb{R}$  ( $k = 0, 1, \dots, N - 1$ ) are weight coefficient that exponentially decay to zero, to be discussed in Section 3.3. The approximate kernel (4) can be interpreted as an (inverse) DCT. Importantly, it covers all types of DCT by appropriately selecting the parameters  $T$ ,  $k_0$  and  $n_0$  as listed in Table 2. Substituting (4) for (1),

$$(x * h)_t \approx \sum_{k=0}^{K-1} \hat{h}^{(k)} \hat{x}_t^{(k)}, \quad \hat{x}_t^{(k)} := \sum_{n=-N+1}^{N-1} x_{t+n} C_n^{(k)}, \quad (5)$$

where we introduced  $C_n^{(k)} = \cos\left(\frac{2\pi}{T}(k+k_0)(n+n_0)\right)$  for simplicity. The variable  $\hat{x}_t^{(k)}$  indicates the  $k$ -th short-time transform coefficient at time  $t$  of the target sequence  $\hat{x}_t$ . Importantly, we can sequentially compute it using a relational expression called that holds between three consecutive short-time transform coefficients  $\hat{x}_{t\pm 1}^{(k)}$

**Table 4.** Comparison of approximate accuracy and weight coefficients derived by existing and proposed approaches. Parameter are set to  $\sigma = 2$ ,  $N = 7$  (window size  $2N - 1 = 13$ ),  $K = 3$  and  $M = 2$ , equivalent to [15] for sanity check. We assume the  $\pm 6\sigma$ -supported Gaussian kernel to be ideal shape. Ideal moment and error values are emphasized in bold.

Method	$\hat{h}^{(0)}$	$\hat{h}^{(1)}$	$\hat{h}^{(2)}$	$\mu_0$	$\mu_2$	$\mu_4$	RMSE
Convolution ( $N = \lceil 6\sigma \rceil$ )				<b>1.000</b>	<b>4.000</b>	<b>48.000</b>	<b>0</b>
Convolution ( $N = \lceil 3\sigma \rceil$ )				<b>1.000</b>	3.808	39.708	$7.93 \times 10^{-4}$
[15] with DCT-3 (from original paper)	0.12916412	0.05765049	0.01155671	<b>1.000</b>	<b>4.000</b>	47.470	$5.73 \times 10^{-4}$
[15] with DCT-3 (by our impl.)	0.12916355	0.05762658	0.01151695	<b>1.000</b>	<b>4.000</b>	47.429	$5.73 \times 10^{-4}$
<b>Ours</b> with DCT-3	0.12916422	0.05765508	0.01156434	<b>1.000</b>	<b>4.000</b>	47.477	$5.73 \times 10^{-4}$

and  $\hat{x}_t^{(k)}$  called a shift property. Our approach is inspired from sliding DCT based on second-order shift property [19]. Specifically,

$$\hat{x}_{t-1}^{(k)} + \hat{x}_{t+1}^{(k)} = 2\mathcal{C}_{1-n_0}^{(k)} \hat{x}_t^{(k)} + \Delta_t, \quad (6)$$

where  $\Delta_t = \mathcal{C}_{-N+1}^{(k)} x_{t-N} + \mathcal{C}_{N-1}^{(k)} x_{t+N} - \mathcal{C}_{-N}^{(k)} x_{t-N+1} - \mathcal{C}_N^{(k)} x_{t+N-1}$ . Using (6), a new  $\hat{x}_{t+1}^{(k)}$  can be consecutively obtained from already-computed  $\hat{x}_t^{(k)}$  and  $\hat{x}_{t-1}^{(k)}$  in  $O(1)$  time/pixel. Moreover, the left-hand side of (5) have the computational complexity of  $O(K)$ . Hence, the overall filtering process runs in constant-time regardless of window length.

### 3.2. Universal analysis on all types of DCT

First, we discuss the symmetry of Gaussian kernel and the symmetry of its approximate kernel. Since Gaussian kernel (2) is an even function symmetric w.r.t. the axis of  $n_0 = 0$ , (4) also has to satisfy this condition to accurately approximate the Gaussian kernel. This means that only DCT-1,3,5,7 ( $n_0 = 0$ ) are appropriate for the approximation but DCT-2,4,6,8 ( $n_0 = 1/2$ ) are not. We therefore focus only on DCT-1,3,5,7 in later discussion.

Second, in the shift property (6), more arithmetic operations can be reduced by factorizing phase of cosines with the same values. Specifically, in DCT-1,  $\mathcal{C}_{\pm(N-1)}^{(k)} = (-1)^k$  and  $\mathcal{C}_{\pm N}^{(k)} = (-1)^k \mathcal{C}_1^{(k)}$  hold; in DCT-3,  $\mathcal{C}_{N-1}^{(k)} = \mathcal{C}_{-(N-1)}^{(k)}$  and  $\mathcal{C}_{\pm N}^{(k)} = 0$  hold; in DCT-5,  $\mathcal{C}_{\pm(N-1)}^{(k)} = \mathcal{C}_{\pm N}^{(k)}$  holds; in DCT-7,  $\mathcal{C}_{\pm(N-1)}^{(k)} = -\mathcal{C}_{\pm N}^{(k)}$  holds. Using these equations,  $\Delta_t$  in (6) can be specified in each DCT type as

$$\Delta_t^{(\text{DCT-1})} = (-1)^k \{(x_{t-N} + x_{t+N}) - \mathcal{C}_1^{(k)}(x_{t-N+1} + x_{t+N-1})\},$$

$$\Delta_t^{(\text{DCT-3})} = \mathcal{C}_{N-1}^{(k)}(x_{t-N} + x_{t+N}),$$

$$\Delta_t^{(\text{DCT-5})} = \mathcal{C}_{N-1}^{(k)}(x_{t-N} + x_{t+N} - x_{t-N+1} - x_{t+N-1}),$$

$$\Delta_t^{(\text{DCT-7})} = \mathcal{C}_{N-1}^{(k)}(x_{t-N} + x_{t+N} + x_{t-N+1} + x_{t+N-1}).$$

Each sliding transform can be computed with roughly two multiplications by appropriately using look-up tables for cosine values and so on. Because the approximation order  $K$  has a very small value in practice and the weight coefficient  $\hat{h}^{(k)}$  are precomputed, the overall filtering process runs fast.

The sliding transforms derived above are basically the same as those of existing work in some cases but new in the other cases. The results of DCT-3 and DCT-5 are basically equivalent to those of [15] and [14], respectively; however, those of DCT-1 and DCT-7 have not been explored in the literature. Evidently, our universal approach provides comprehensive understanding about all types of DCT for  $O(1)$  GF. Moreover, we should mention to avoid misunderstanding that short-time weight coefficients  $\hat{x}_t^{(k)}$  are equivalent to short-time DCT coefficients in the cases of DCT-5,6,7 ( $T = 2N - 1$ ); by contrast, they are very similar but different otherwise. This is caused by

a mismatch of period length  $T$  to window size  $2N - 1$ . For this reason, the above sliding transforms essentially differ from the sliding DCTs derived in [19].

### 3.3. Optimization with moment preservation

We describe how to appropriately compute  $\hat{h}^{(k)}$  for each of DCT-1,3,5,7. Given scale  $\sigma$  and approximation order  $K$ , the optimal values of  $\hat{h}^{(k)}$  that minimize approximate error of Gaussian kernel can be computed as follows: We first introduce the following vector form for explanation:  $\mathbf{h} = [h_0, \dots, h_{N-1}]^T \in \mathbb{R}^N$ ,  $\hat{\mathbf{h}} = [\hat{h}^{(0)}, \dots, \hat{h}^{(K-1)}]^T \in \mathbb{R}^K$ ,  $\mathbf{C} = \{\mathcal{C}_n^{(k)}\}_{n=0, k=0}^{N-1, K-1} \in \mathbb{R}^{N \times K}$  and  $\mathbf{W} = \text{diag}(\frac{1}{2}, 1, \dots, 1) \in \mathbb{R}^{N \times N}$ , which is a weight matrix for dealing with even symmetry of kernel. We also define some vectors about moments:  $\boldsymbol{\mu} = [\mu_0, \mu_2, \dots, \mu_{2M-2}]^T \in \mathbb{R}^M$  and  $\mathbf{P} = \{n^{2m}\}_{n=0, m=0}^{N-1, M-1} \in \mathbb{R}^{N \times M}$  where  $M$  is the number of moments to be preserved.

We derive a close-form solution that provides the optimum values of  $\hat{h}^{(k)}$ . The optimization can be formalized as the quadratic problem with moment constraints

$$\hat{\mathbf{h}}_* := \arg \min_{\hat{\mathbf{h}}} f(\hat{\mathbf{h}}) \quad \text{subject to} \quad \mathbf{g}(\hat{\mathbf{h}}) = \mathbf{0}, \quad (7)$$

$$f(\hat{\mathbf{h}}) := \frac{1}{2}(\mathbf{h} - \mathbf{C}\hat{\mathbf{h}})^T \mathbf{W}(\mathbf{h} - \mathbf{C}\hat{\mathbf{h}}) \in \mathbb{R}, \quad (8)$$

$$\mathbf{g}(\hat{\mathbf{h}}) := \mathbf{P}^T \mathbf{W}\mathbf{C}\hat{\mathbf{h}} - \frac{1}{2}\boldsymbol{\mu} \in \mathbb{R}^M. \quad (9)$$

This problem can be solved by the method of Lagrange multiplier. Specifically,  $F(\hat{\mathbf{h}}, \boldsymbol{\lambda}) := f(\hat{\mathbf{h}}) - \boldsymbol{\lambda}^T \mathbf{g}(\hat{\mathbf{h}})$  where  $\boldsymbol{\lambda} \in \mathbb{R}^M$  is a vector of Lagrange multipliers. From  $\partial F / \partial \hat{\mathbf{h}} = \mathbf{0}$  and  $\partial F / \partial \boldsymbol{\lambda} = \mathbf{0}$ , we obtain the linear equation

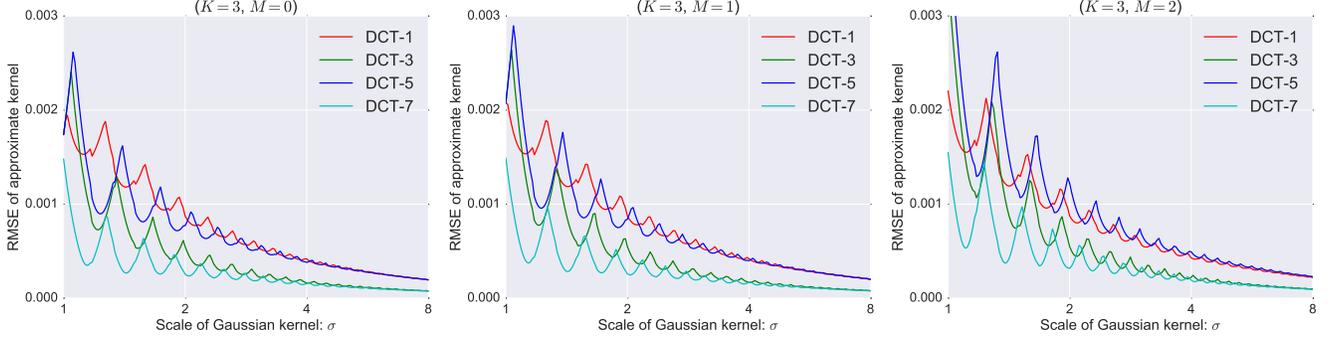
$$\begin{bmatrix} \mathbf{C}^T \mathbf{W}\mathbf{C} & \mathbf{U} \\ \mathbf{U}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{h}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{C}^T \mathbf{W}\mathbf{h} \\ \boldsymbol{\mu}/2 \end{bmatrix}. \quad (10)$$

where  $\mathbf{U} = \mathbf{C}^T \mathbf{W}\mathbf{P}$ . Using the least-square solution without moment constraints  $\hat{\mathbf{h}}_{\text{LS}} = (\mathbf{C}^T \mathbf{W}\mathbf{C})^{-1} \mathbf{C}^T \mathbf{W}\mathbf{h}$ , i.e.,  $M = 0$ , we can solve (10) as

$$\hat{\mathbf{h}}_* = \hat{\mathbf{h}}_{\text{LS}} + (\mathbf{C}^T \mathbf{W}\mathbf{C})^{-1} \mathbf{U}\mathbf{S}^{-1} \left( \mathbf{U}^T \hat{\mathbf{h}}_{\text{LS}} - \frac{1}{2}\boldsymbol{\mu} \right). \quad (11)$$

where  $\mathbf{S} = \mathbf{U}^T (\mathbf{C}^T \mathbf{W}\mathbf{C})^{-1} \mathbf{U}$ . This closed-form solution can be understood as the one projected from  $\hat{\mathbf{h}}_{\text{LS}}$  onto linear constraint space  $\mathbf{g}(\hat{\mathbf{h}}) = \mathbf{0}$  by oblique projections.

The above computation can be preformed negligibly fast. This is because most vectors/matrices in (11) basically depend on  $K$  and  $M$  that have very small values in practice, e.g.,  $K = 4$  and  $M = 2$  at most. The two inverse matrices in (11) are regular and easy-to-compute as shown below. We introduce  $\mathbf{M}_n$  and  $\mathbf{M}_k$  as shown in



**Fig. 1.** Approximate error where  $K = 3$  and  $M = 0, 1, 2$  (columns). The most left column ( $M = 0$ ) corresponds to least-square solution without moment preservation. Obviously, DCT-3,7 are superior to DCT-1,5 and DCT-7 outperforms DCT-3 especially when  $\sigma < 3$ .

Table 3. Let us consider

$$\mathbf{C}^\top \mathbf{W} \mathbf{C} = \mathbf{C}^\top \mathbf{M}_n \mathbf{C} + \mathbf{C}^\top (\mathbf{W} - \mathbf{M}_n) \mathbf{C}. \quad (12)$$

The first term can be simplified using the orthogonality of DCT matrix:  $\mathbf{C}^\top \mathbf{M}_n \mathbf{C} = \frac{T}{4} \mathbf{M}_k$ . In the case of DCT-3,5,7, the second term is vanished since  $\mathbf{W} = \mathbf{M}_n$ . In the case of DCT-1,  $\mathbf{W} - \mathbf{M}_n = \text{diag}(0, 0, \dots, \frac{1}{2})$  holds.

$$\mathbf{C}^\top (\mathbf{W} - \mathbf{M}_n) \mathbf{C} = \frac{1}{2} \mathbf{s} \mathbf{s}^\top. \quad (13)$$

where  $\mathbf{s} = \{(-1)^k\}_{k=0}^{K-1} \in \mathbb{R}^K$ . From these results, using the Sherman-Morrison formula,  $(\mathbf{C}^\top \mathbf{W} \mathbf{C})^{-1}$  equals to

$$\text{DCT-1: } \frac{4}{T} \left\{ \mathbf{M}_k^{-1} - \frac{(\mathbf{M}_k^{-1} \mathbf{s})(\mathbf{M}_k^{-1} \mathbf{s})^\top}{N - 1 + \text{tr}(\mathbf{M}_k^{-1})} \right\}, \quad (14)$$

$$\text{DCT-3,5,7: } \frac{4}{T} \mathbf{M}_k^{-1}. \quad (15)$$

Obviously, this is a regular matrix in both cases. Moreover,  $\mathbf{S}$  is non-singular because of properties of Vandermonde matrix and Gram matrix. Computing  $\mathbf{S}^{-1} \in \mathbb{R}^{M \times M}$  has low computational complexity since  $M = 1$  or  $M = 2$  in practice, which is slightly more efficient than the approach of [15], which requires inverse operation of a  $K \times K$  matrix. Thus, as compared with the existing methods, our universal approach is a generalization of the existing methods but provides a tighter solution based on theoretical understanding. Similar to [14], we determine an optimal value for  $N$  from given  $\sigma$  and  $K$  by binary search according to truncation error. This computation runs negligibly fast as compared with main filtering process since  $K$  and  $M$  have very small values in practice.

#### 4. EXPERIMENTS AND DISCUSSIONS

We comprehensively compare the performance of Gaussian approximation using each type of DCT via our universal approach. Table 4 shows weight coefficients and approximate error where  $\sigma = 2$  and  $N = 7$  (window size: 13),  $K = 3$  and  $M = 2$ . This experimental condition followed the one indicated in [15] for sanity check. Here, we supposed  $\pm 6\sigma$ -supported Gaussian kernel as the ideal kernel. Approximate error is quantified by Root-Mean-Square Error (RMSE) between the ideal and approximate kernels. These results showed that our approach exactly preserved sum  $\mu_0$  and variance  $\mu_2$ . Different parameter setting also showed the same tendency. An important fact is that  $\pm 3\sigma$ -supported Gaussian convolution shows not only variance distortion but also larger approximate error than

the two DCT-based  $O(1)$  GF. Hence, we state that our approach can perform GF with higher accuracy than naive GF.

Figure 1 plots the approximate error of DCT-1,3,5,7 where  $\sigma \in [1, 8]$ ,  $K = 3$  and  $M = 0, 1, 2$ . The saw-shape curves were caused by variation of optimum window size to be integer. The larger  $M$  produced the slightly larger error. This is intuitive because the more constraints cause the smaller search space. More importantly, DCT-3,7 significantly outperformed DCT-1,5 in approximate accuracy as larger  $\sigma$  prominently showed. According to [15], DCT-3 shows higher accuracy than DCT-5. However, more exactly, we found that on average DCT-7 produces the most accurate approximation. This important fact was firstly revealed through our universal approach. As a whole, DCT-5 showed the worst accuracy and DCT-1 was the next but it showed somewhat accurate results when  $\sigma$  was small. We consider that these behaviors come from the shape of the first cosine basis of each type of DCT. Specifically, the basis of  $k = 0$  in DCT-1,5 is a DC component  $C_n^{(0)} = 1$  but DCT-3,7 have an unimodal shape because of  $k_0 = \frac{1}{2}$ . This shape match can promote to approximate the Gaussian shape.

We also confirmed actual running time of naive convolution and our  $O(1)$  GFs. The test image was "baboon" (512×512 pixels, 8-bit grayscale). The test environment mounted on Intel Core i5-6200U 2.30GHz CPU with 8GB main memory. All the implementations were written in C++ and compiled by VC++2015 with "/O2" option. For fair comparison, we made all the methods share the similar number of multiplications per pixel:  $N = 4$  ( $\sigma = 1$ ) in  $\pm 3\sigma$ -supported Gaussian convolution and  $K = 2$  in DCT-based  $O(1)$  GF. As a result, the convolution took roughly 2.8–3.1 [ms] and DCT-based  $O(1)$  GF took approximately 1.5–1.7 [ms] regardless of scale  $\sigma$  and DCT type. The major reason why DCT-based  $O(1)$  GF is much faster than convolution with small-sized window probably comes from memory access cost. The performance of modern computers is dominated by memory access, not CPU operations in most cases. For instance, (1) requires  $(2N - 1)$  pixels but  $\Delta_t$  in (6) requires just 4 pixels.

#### 5. CONCLUSIONS

This paper presented a universal approach for DCT-based  $O(1)$  GF and analyzed its performance comprehensively. Our universal approach covered all types of DCT and arbitrary moments and provided a closed-form solution for optimal coefficients. In spite of its wider coverage, our solution for moment preservation was tighter than that of [15]. Through our discussion, we achieved a generalized conclusion different from [15]: DCT-7 produces the highest accuracy in DCT-1,3,5,7. Future work will extend this discussion to multi-dimensional cases and other filters.

## 6. REFERENCES

- [1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis. (IJCV)*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [2] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [3] Q. Yang, K. H. Tan, and N. Ahuja, “Real-time O(1) bilateral filtering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2009, number 1, pp. 557–564.
- [4] K. N. Chaudhury, “Acceleration of the shifttable O(1) algorithm for bilateral filtering and nonlocal means,” *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1291–1300, Apr. 2013.
- [5] K. Sugimoto and S. Kamata, “Compressive bilateral filtering,” *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3357–3369, Nov. 2015.
- [6] G. Deng, “Fast compressive bilateral filter,” *Electronics Letters*, vol. 53, pp. 150–152, Feb. 2017.
- [7] B. Cai, X. Xing, and X. Xu, “Edge/Structure preserving smoothing via Relativity-of-Gaussian,” *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 250–254, Sept. 2017.
- [8] R. Deriche, “Fast algorithms for low-level vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 78–87, 1990.
- [9] I. T. Young and L. J. van Vliet, “Recursive implementation of the Gaussian filter,” *Signal Process.*, vol. 44, no. 2, pp. 139–151, June 1995.
- [10] L. J. van Vliet, I. T. Young, and P. W. Verbeek, “Recursive Gaussian derivative filters,” in *Proc. Int. Conf. Pattern Recognition (ICPR)*, 1998, vol. 1, pp. 509–514.
- [11] 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.
- [12] E. Elboher and M. Werman, “Cosine integral images for fast spatial and range filtering,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2011, pp. 89–92.
- [13] K. Sugimoto and S. Kamata, “Fast Gaussian filter with second-order shift property of DCT-5,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2013, pp. 514–518.
- [14] K. Sugimoto and S. Kamata, “Efficient constant-time Gaussian filtering with sliding DCT/DST-5 and dual-domain error minimization,” *ITE Trans. Media Technol. Appl.*, vol. 3, no. 1, pp. 12–21, 2015.
- [15] D. Charalampidis, “Recursive implementation of the Gaussian filter using truncated cosine functions,” *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3554–3565, 2016.
- [16] F. C. Crow, “Summed-area tables for texture mapping,” *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 18, no. 3, pp. 207–212, July 1984.
- [17] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, vol. 1, pp. 511–518.
- [18] P. Getreuer, “A survey of Gaussian convolution algorithms,” *Image Processing On Line*, vol. 3, pp. 286–310, 2013.
- [19] V. Kober, “Fast algorithms for the computation of sliding discrete sinusoidal transforms,” *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1704–1710, June 2004.