# EFFICIENT KEYPOINT DETECTION AND DESCRIPTION VIA POLYNOMIAL REGRESSION OF SCALE SPACE

*Ryo Okutani, Kenjiro Sugimoto, and Sei-ichiro Kamata*

Waseda University, Graduate School of Information, Production and Systems, JAPAN

## ABSTRACT

Keypoint detection and description using approximate continuous scale space are more efficient techniques than typical discretized scale space for achieving more robust feature matching. However, this state-of-the-art method requires high computational complexity to approximately reconstruct, or decompress, the value at an arbitrary point in scale space. Specifically, it has $O(M^2)$ computational complexity where $M$ is an approximation order. This paper presents an efficient scale space approach that provides decompression operation with $O(M)$ complexity without a loss of accuracy. As a result of the fact that the proposed method has much fewer variables to be solved, the least-square solution can be obtained through normal equation. This is easier to solve than the existing method which employs Karhunen–Loeve expansion and generalized eigenvalue problem. Experiments revealed that the proposed method performs as expected from the theoretical analysis.

***Index Terms—*** Feature extraction, Spectral SIFT, Scale space, Compressed scale space, Image filtering

## 1. INTRODUCTION

Keypoint feature has played an essential role in image processing, computer vision, and pattern recognition. It has flourished in various tasks including structure-from-motion [1], object recognition [2, 3], and image retrieval [4]. In general, the appearance of a visual object in a visual scene geometrically varies due to camera and/or object motion. The keypoint feature approaches enable us to uniformly handle visual objects with appearance variation by using geometric invariance. Many methods have been proposed after SIFT [2, 3] year on year such as SURF [5], PCA-SIFT [6], FAST [7], AGAST [8], CARD [9], FREAK [10], SPADE [11], ORB [12], SIFER [13], DSIFER [14], which are pursuing a higher trade-off between computational complexity, robustness, and stability. We focus on keypoint feature with scale and rotation invariance.

An important theory for scale and rotation invariance is scale space representation of an image. Scale space is a scalar field generated by convolution between an input image and an isotropic kernel with scale parameter $\sigma$. As a primitive idea, Lindeberg [15] validated that extrema in scale space generated by Laplacian-of-Gaussian (LoG) kernel can be used as keypoints robust to scale and rotation changes. However, it is computationally-expensive to straightforwardly compute scale space and its extrema due to its dimensionality elevation, i.e., a $D$-dimensional image leads to a $(D+1)$-dimensional scale space. SIFT achieved realistic complexity by handling LoG kernel as Difference-of-Gaussian (DoG) kernel derived from diffusion equation. SURF enabled online image processing by using, instead of LoG kernel, box-stacked kernel and integral image [16] with a certain sacrifice of rotation invariance. The other existing methods [6–14] also discussed more efficient approaches of scale space for robust keypoint feature. Thus, the performance depends heavily on how to generate and deal with scale space.

A major difficulty of feature keypoint is mainly caused by considerable size of scale space. Although the existing methods roughly describe scale space as a stack of its sliced images along the $\sigma$ axis, this representation results in degrading the stability of keypoints. This is because some extrema in the original scale space may be lost in its over-discretized representation. A remarkable work on this representation problem is Spectral SIFT [17], achieving to describe scale space as a linear combination of several component images without slicing $\sigma$. As inspired from this concept, we call the representation a *compressed scale space* for convenience. This approach is achieved by decomposing a kernel into basis kernels and their weighted function approximated by $M$-order polynomials via continuous Karhunen-Loeve transform (KLT). The compressed scale space provides an operation to approximately-reconstructed, i.e., lossy-decompressed, the value of an arbitrary point in $O(M^2)$ time. This idea significantly improves the stability of keypoint detection because of continuousness and analytical extrema detection. However, the computational time of feature description are still expensive due to the $O(M^2)$ time per point and millions of decompression operations per image.

This paper presents a keypoint detection/description method with lower computational time than but theoretically-equivalent stability to Spectral SIFT. This improvement is achieved by reducing the time of the decompression operation from $O(M^2)$ to $O(M)$ without any loss in accuracy. The key technique is compressed scale space derived from

polynomial regression of an $M$-order polynomial of $\sigma$, which is straightforwardly solvable using normal equation and easy to derive because of fewer unknown variables than Spectral SIFT. Experiments showed that our method reduced 34% running time in total (41% in description process and 19% in detection process) without a loss in accuracy.

## 2. RELATED WORK

This section summarizes existing algorithms for keypoint detection and description. Consider convolution between a two-dimensional target image $f(\boldsymbol{p}) \in \mathbb{R}$ and an isotropic kernel $h(\sigma; r) \in \mathbb{R}$ where $\boldsymbol{p} \in \mathbb{Z}^2$ is a pixel position in the image, $r \in \mathbb{R}_+$ is the radius and $\sigma \in \mathbb{R}_+$ is the scale of the kernel. This operation generates

$$H(\sigma; \boldsymbol{p}) = \sum_{\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})} h(\sigma; \|\boldsymbol{q} - \boldsymbol{p}\|)\, f(\boldsymbol{q}), \tag{1}$$

where $\mathcal{N}(\boldsymbol{p}) \subset \mathbb{Z}^2$ indicates pixel positions in the neighborhood of $\boldsymbol{p}$ and $\|\cdot\|$ denotes the $\ell_2$-norm of a vector. This three-dimensional scalar field is generally called a *scale space*. The most common choices of $h(\cdot;\cdot)$ are the Gaussian kernel or the (scale-normalized) LoG kernel defined by

$$h^{(\mathrm{gauss})}(\sigma; r) := \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, \tag{2}$$

$$h^{(\mathrm{LoG})}(\sigma; r) := \frac{r^2 - 2\sigma^2}{2\pi\sigma^4} e^{-\frac{r^2}{2\sigma^2}}. \tag{3}$$

A major difficulty for generating a scale space is the space and computational complexity of (1), which increases in proportion to the window area $|\mathcal{N}(\cdot)|$ determined from $\sigma$.

Most algorithms in computer vision roughly handle a scale space by discretizing $\sigma$ aggressively. In keypoint detection, Lindeberg [15] indicated that extrema in LoG scale space are robust to scale and rotation changes. Inspired by this work, SIFT [2, 3] uses the difference of Gaussian scale space instead of LoG scale space. This is because it approaches asymptotically to the LoG scale space when $\Delta\sigma \to 0$. In local description, SIFT also uses Gaussian scale space to describe some of detected keypoints. Both scale spaces are handled by slicing them in a logarithmic manner of $\sigma$ in order to reduce the complexity of (1). However, this discretization deforms the variation of $H(\sigma; \boldsymbol{p})$ in the direction of $\sigma$.

A remarkable solution to the over-discretization problem is the Spectral SIFT [17]. Based on spectral theory, this method approximates the kernel $h(\cdot;\cdot)$ by

$$\hat{h}(\sigma; r) = \sum_{m=0}^{M} w_m(\sigma)\phi_m(r), \tag{4}$$

where $w_m(\cdot)$ are called weight functions and $\phi_m(\cdot)$ are called basis kernels. Note that $\lim_{M\to\infty} \hat{h}(\sigma; r) = h(\sigma; r)$. By substituting (4) for (1), $H(\cdot;\cdot)$ is approximated by

$$\hat{H}(\sigma; \boldsymbol{p}) = \sum_{m=0}^{M} w_m(\sigma)\Phi_m(\boldsymbol{p}), \tag{5}$$

where $\Phi_m(\cdot)$ are component images defined by

$$\Phi_m(\boldsymbol{p}) = \sum_{\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})} \phi_m(\|\boldsymbol{q} - \boldsymbol{p}\|)\, f(\boldsymbol{q}), \tag{6}$$

which is generated by convolution between the target image and the basis kernels. We call $\hat{H}(\cdot;\cdot)$ a *compressed scale space*. The optimal decomposition of (4) in the least-square manner corresponds to the solution of the continuous Karhunen-Loeve expansion; however, it cannot be solved analytically. Hence, assuming that $w_m(\sigma)$ is sufficiently-smooth within a certain interval $\sigma \in [\sigma_1, \sigma_2]$, this method approximates $w_m(\cdot)$ by the $M$-order polynomial

$$w_m(\sigma) \approx \sum_{n=0}^{M} a_n^{(m)} \sigma^n, \tag{7}$$

and then computes $a_n^{(m)}$ and the corresponding $\phi_m(\cdot)$ via a generalized eigenvalue problem. In the compressed scale space, its extrema in the $\sigma$ direction can be derived by

$$\frac{\partial \hat{H}(\sigma; \boldsymbol{p})}{\partial \sigma} = \sum_{n=0}^{M-1} \left\{ (n+1) a_{n+1}^{(m)} \sum_{m=0}^{M} \Phi_m(\boldsymbol{p}) \right\} \sigma^n = 0, \tag{8}$$

which can be solved by the quadratic formula if $M = 3$. By using a sufficiently narrow interval $\sigma \in [\sigma_1, \sigma_2]$ (c.f. octave strategy), the order parameter $M$ can have a small value.

A remaining problem of the above compressed scale space is the high computational complexity of computing (5), i.e., decompression. Spectral SIFT employs LoG and Gaussian compressed scale space in keypoint detection and local description, respectively. Once the $(M + 1)$ component images are generated in advance, we can compute (5) in $O(M^2)$ time, because of variable separation into scale $\sigma$ and spatial variable $r$ in (4). However, the complexity is still high even if $M$ is small because the local description step requires to decompress various locations in $\hat{H}(\cdot;\cdot)$ numerous times.

## 3. PROPOSED METHOD

We propose a compressed scale space that provides decompression operation with lower computational complexity to reduce the computational time.

### 3.1. Kernel Approximation via Polynomial Regression

Since Gaussian and LoG scale space have a smooth variation in the $\sigma$ direction, our method approximates the kernel $h(\cdot;\cdot)$

and its scale space $H(\cdot;\cdot)$ by the $M$-order polynomials

$$\hat{h}(\sigma;r)=\sum_{m=0}^{M}\sigma^m\phi_m(r),\quad \hat{H}(\sigma;\boldsymbol{p})=\sum_{m=0}^{M}\sigma^m\Phi_m(\boldsymbol{p}) \quad (9)$$

which are understood as a specific case of $w_m(\sigma)=\sigma^m$ in (7). Consider a compressed scale space with $\sigma\in[\sigma_1,\sigma_2]$. The optimal polynomial regression of (9) in the least-square manner is derived by the normal equation $\boldsymbol{A}\boldsymbol{\phi}(r)=\boldsymbol{b}(r)$, where $\boldsymbol{\phi}(r)=[\phi_0(r),\ldots,\phi_M(r)]^\top\in\mathbb{R}^{M+1}$, $\boldsymbol{A}=(A_{k,l})\in\mathbb{R}^{(M+1)\times(M+1)}$ and $\boldsymbol{b}(r)=(b_k(r))\in\mathbb{R}^{M+1}$ with

$$A_{k,l}=\int_{\sigma_1}^{\sigma_2}\sigma^{k+l}\,d\sigma=\frac{\sigma_2^{k+l+1}-\sigma_1^{k+l+1}}{k+l+1}, \quad (10)$$

$$b_k(r)=\int_{\sigma_1}^{\sigma_2}\sigma^k\,h(\sigma;r)\,d\sigma. \quad (11)$$

In the case of Gaussian kernel, if $r\neq 0$, (11) is expanded to

$$b_k^{(\mathrm{gauss})}(r)=-\frac{r^{k-1}}{2^{\frac{k+3}{2}}\pi}\int_{\frac{r^2}{2\sigma_1^2}}^{\frac{r^2}{2\sigma_2^2}}t^{(-\frac{k-1}{2})-1}e^{-t}dt, \quad (12)$$

otherwise,

$$b_k^{(\mathrm{gauss})}(0)=\begin{cases}\frac{1}{2\pi}\left(\log\sigma_2-\log\sigma_1\right) & \text{if } k=1\\ \frac{1}{2\pi(k-1)}(\sigma_2^{k-1}-\sigma_1^{k-1}) & \text{otherwise}\end{cases}. \quad (13)$$

In the case of LoG kernel, (11) has the form:

$$b_k^{(\mathrm{LoG})}(r)=-2b_k^{(\mathrm{gauss})}(r)+r^2b_{k-2}^{(\mathrm{gauss})}(r). \quad (14)$$

Note that the integral in (12) can be calculated using the incomplete gamma function $\Gamma(a,x)=\int_x^\infty t^{a-1}e^{-t}dt$. Specifically, we obtain $\boldsymbol{\phi}(r)$ by first computing $\boldsymbol{b}(r)$ and then multiplying $\boldsymbol{A}^{-1}$ to it for each $r$. Similar to Spectral SIFT, the extrema in our compressed scale space is derived from

$$\frac{\partial\hat{H}(\sigma;\boldsymbol{p})}{\partial\sigma}=\sum_{m=0}^{M-1}(m+1)\,\Phi_m(\boldsymbol{p})\,\sigma^m=0, \quad (15)$$

which can be solved by the quadratic formula if $M=3$. Figure 1 depicts $m$-th basis kernels of the Gaussian kernel and the LoG kernel where $m=0,1,2,3$. Both higher-order basis kernels converge to almost zero. Hence, $M=3$ is sufficient to well approximate their kernels.

### 3.2. Advantages over Spectral SIFT

Our method has the clear advantage that an arbitrary location in our compressed scale space is decompressed in $O(M)$ time as (9) shows. In other words, our idea has succeeded to sufficiently reduce the $O(M^2)$ complexity of Spectral SIFT.

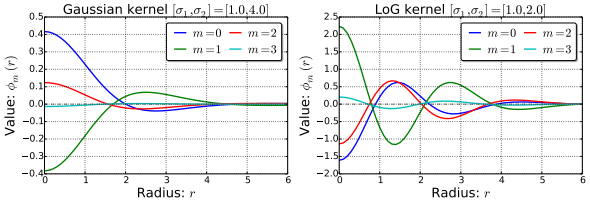



**Fig. 1**: Shape of basis kernels $\phi_m(r)$ of two-dimensional Gaussian kernel (left) and LoG kernel (right).

Moreover, its approximate accuracy is maintained from Spectral SIFT as shown below. By substituting (7) for (5),

$$\hat{H}(\sigma;\boldsymbol{p})=\sum_{m=0}^{M}\left(\sum_{n=0}^{M}a_n^{(m)}\sigma^n\right)\Phi_m(\boldsymbol{p}) \quad (16)$$

$$=\sum_{n=0}^{M}\sigma^n\left\{\sum_{m=0}^{M}a_n^{(m)}\Phi_m(\boldsymbol{p})\right\}. \quad (17)$$

If we redefine $\{\cdot\}$ as a new $\Phi_m(\cdot)$, this is equivalent to (9) of our method. It runs without any loss of accuracy as compared with Spectral SIFT. From a theoretical viewpoint, our solution is considered as a different solution provided by Spectral SIFT via a linear transform, which transfers computing $w_m(\cdot)$ in the decompressing part to computing $\Phi_m(\cdot)$ in the precomputing part. Normal equation is easier than generalized eigenvalue problem to solve because of much fewer parameter. Note that our method eliminates $a_n^{(m)}$ but still provides the essentially same solution. Thus, our faster decompression sufficiently accelerates local description as well as keypoint detection and we also inherited the advantages of Spectral SIFT. Incidentally, our compressed scale space may provide faster filtering than the state-of-the-art constant-time filtering algorithms including recursive filters [18, 19] and frequency-sampling approaches [20–23]. This is because our decompression operation requires $M$ multiplications/pixel only.

## 4. EXPERIMENTS AND DISCUSSION

This section verifies practical performance of our method. The test environment mounts on an Intel Core i7-4770 3.40GHz CPU with 8GB main memory. The competitors are SIFT [2, 3], Spectral SIFT [17], and our method that all are written in C++ with OpenCV 2.4.11 [24]. Their parameters are $L=6$ for SIFT and $M=3$ for both Spectral SIFT and our method where $L$ and $(M+1)$ indicate the number of convolutions per octave. We set $\sigma\in[1.0,4.0]$ with some margin for processing each octave of compressed scale space. The test image set is the Oxford dataset [25], which contains one standard image (Image 1) and its five other-view images (Image 2–6) in each subset ("leuven", "trees", "ubc", "boat", "graf", and "wall").

Figure 2 shows the repeatability score defined in [26] about the six subsets, which have some visual deformations

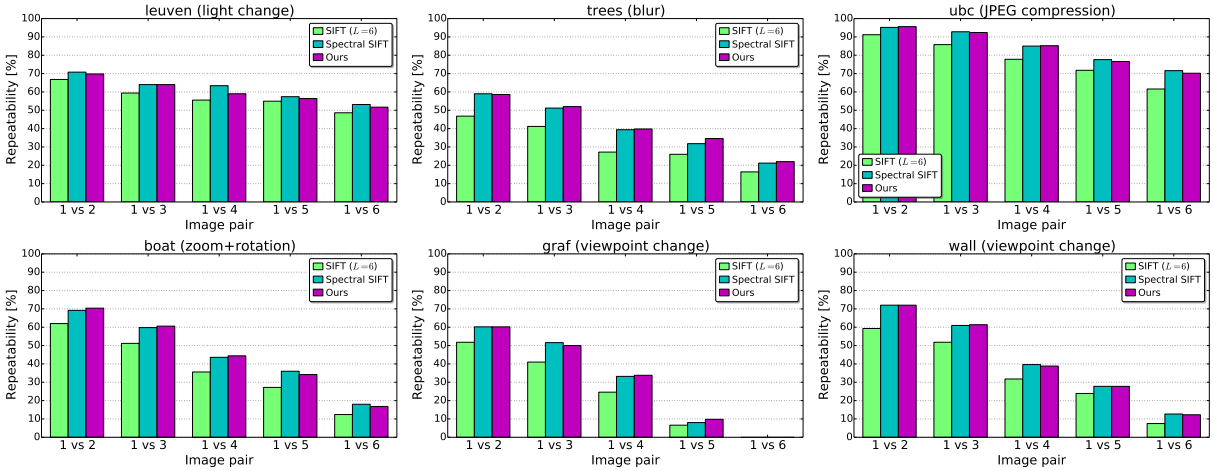
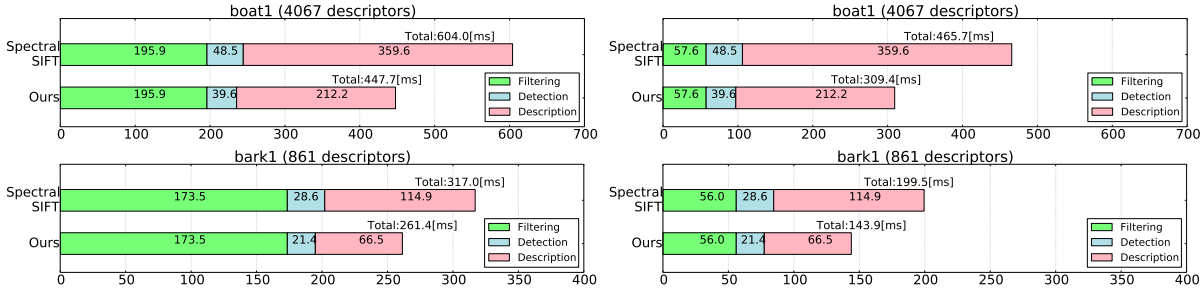



**Fig. 2**: Repeatability of keypoint detection for 50% overlap error in Oxford dataset




(a) Single-core filtering

(b) Multi-core filtering ($\times 4$)

**Fig. 3**: Computational time of Spectral SIFT and our method ($M = 3$)

such as scale, rotation, blur and so on as annotated to each subfigure caption. Evidently, our method achieves repeatability score higher than SIFT and comparable to Spectral SIFT. These results support our theoretical analysis mentioned in Section 3.2. Our method seems to inherit regular characteristics of Spectral SIFT about robustness.

Figure 3 shows computational time of Spectral SIFT and our method in "boat1" (850×680 pixels) and "bark1" (765×512 pixels) subsets. We test two cases of single-core and multi-core processing with OpenMP for filtering via the FFT implemented in OpenCV where both methods require $(2M + 2)$ convolutions to generate LoG and Gaussian compressed scale space. In all cases, our method sufficiently outperforms Spectral SIFT in terms of total time and, in particular, description time. The case of multi-core filtering shows that our method reduces the total time (and the description time) by 34% (and 41%) in "boat1" and by 28% (and 43%) in "bark1". Our method achieves a sufficient reduction rate since decompression operation occupies the largest portion of the keypoint description process, which is obviously dominant in the whole process. A limitation of our method is its reduction rate lower than the expectation of our theoretical analysis, i.e., $O(M^2)$ to $O(M)$ where $M = 3$. This is because it contains many other subprocesses including solving quadratic equations and generating orientation histograms. Hence, our method runs faster if more keypoints are detected as "boat1" has revealed.

## 5. CONCLUSIONS

This paper presented an efficient method for keypoint detection and description based on compressed scale space. The major idea was to simplify variations in the $\sigma$ direction of scale space as low-order polynomials, which are solved via normal equation in the least-square sense. As compared with Spectral SIFT, it runs significantly faster with comparable robustness to visual deformation. As future work, we will design more efficient algorithms by enhancing the robustness to more complex visual deformations [15, 27].


### Acknowledgement

The authors would like to thank Gou Koutaki for valuable discussions about Spectral SIFT [17] and its source code.

## 6. REFERENCES


[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 72–79, 2009.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, 1999.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis. (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[4] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, pp. 2161–2168, 2006.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, , no. September, 2008.

[6] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, pp. 2–9, 2004.

[7] E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 35, 2008.

[8] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," *Europian Conf. Comput. Vis. (ECCV)*, 2010.

[9] M. Ambai and Y. Yoshida, "CARD : Compact And Real-time Descriptors," *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 97–104, 2011.

[10] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 510–517, 2012.

[11] M. Ambai and I. Sato, "SPADE : Scalar Product Accelerator by Integer," *Europian Conf. Comput. Vis. (ECCV)*, pp. 267–281, 2014.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 2564–2571, 2011.

[13] P. Mainali, G. Lafruit, Q. Yang, B. Geelen, L. V. Gool, and R. Lauwereins, "SIFER: Scale-invariant feature detector with error resilience," *Int. J. Comput. Vis. (IJCV)*, vol. 104, no. 2, pp. 172–197, 2013.

[14] P. Mainali, G. Lafruit, K. Tack, L. Van Gool, and R. Lauwereins, "D-SIFER: Derivative-based Scale Invariant Image Feature Detector with Error Resilience.," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2380–2391, 2014.

[15] T. Lindeberg, *Scale-space theory in computer vision*, Kluwer Academic publisher, 1994.

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

[17] G. Koutaki and K. Uchimura, "Scale-space processing using polynomial representation," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2744–2751, 2014.

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

[19] L.J. Van Vliet, I.T. Young, and P.W. Verbeek, "Recursive Gaussian derivative filters," *in Proc. IEEE Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, no. August, pp. 509–514, 1998.

[20] C.M. Rader and B. Gold, "Digital filter design techniques in the frequency domain," *Proceedings of the IEEE*, vol. 55, no. 2, 1967.

[21] K. Sugimoto and S. Kamata, "Fast image filtering by DCT-based kernel decomposition and sequential sum update," *in Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 125–128, 2012.

[22] K. Sugimoto and S. Kamata, "Fast gaussian filter with second-order shift property of DCT-5," pp. 514–518, 2013.

[23] K. Sugimoto and S. Kamata, "Efficient Constant-time Gaussian Filtering with Sliding DCT/DST-5 and Dual-domain Error Minimization," *ITE Trans. Media Tech. Appl.*, vol. 3, no. 1, pp. 12–21, 2015.

[24] "Open Source Computer Vision Library," `http://opencv.org/`.

[25] "Oxford Dataset," `http://www.robots.ox.ac.uk/~vgg/research/affine/`.

[26] K. Mikolajczyk, T. Tuytelaars, C. Schmid, a. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis. (IJCV)*, vol. 65, no. 1-2, pp. 43–72, 2005.

[27] K. Mikolajczyk and C Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis. (IJCV)*, vol. 60, no. 1, pp. 63–86, 2004.