Multiple-hypothesis affine region estimation with anisotropic LoG filters

Takahiro Hasegawa
Chubu University
tkhr@vision.cs.chubu.ac.jp

Mitsuru Ambai, Kohta Ishikawa
Denso IT Laboratory, Inc.
{manbai, kishikawa}@d-itlab.co.jp

Gou Koutaki
Kumamoto University
koutaki@cs.kumamoto-u.ac.jp

Yuji Yamauchi, Takayoshi Yamashita, Hironobu Fujiyoshi
Chubu University
yuu@vision.chubu.ac.jp, {yamashita, hf}@cs.chubu.ac.jp

Abstract

We propose a method for estimating multiple-hypothesis affine regions from a keypoint by using an anisotropic Laplacian-of-Gaussian (LoG) filter. Although conventional affine region detectors, such as Hessian/Harris-Affine, iterate to find an affine region that fits a given image patch, such iterative searching is adversely affected by an initial point. To avoid this problem, we allow multiple detections from a single keypoint. We demonstrate that the responses of all possible anisotropic LoG filters can be efficiently computed by factorizing them in a similar manner to spectral SIFT. A large number of LoG filters that are densely sampled in a parameter space are reconstructed by a weighted combination of a limited number of representative filters, called "eigenfilters", by using singular value decomposition. Also, the reconstructed filter responses of the sampled parameters can be interpolated to a continuous representation by using a series of proper functions. This results in efficient multiple extrema searching in a continuous space. Experiments revealed that our method has higher repeatability than the conventional methods.

1. Introduction

Keypoint matching is a fundamental task in applications such as object recognition, image mosaicking, large-scale image searching and 3D reconstruction. It involves finding the correspondence between keypoints in a set of images, each having a different viewing angle, and other imaging conditions. A common matching technique that is robust across conditions is scale-invariant keypoint detection and matching [10, 11, 12, 14, 2]. Scale-invariant keypoint detectors, such as Laplacian-of-Gaussian (LoG) [10, 11] and Difference-of-Gaussian (DoG) [12], estimate scale by determining the parameters at which the response value of the LoG or DoG becomes an extremum from a scale space constructed by Gaussian kernels of different sizes. The Hessian-Laplace and Harris-Laplace keypoint detectors [14] also estimate scale using the same Gaussian scale space as LoG and DoG. In addition, the Speeded-Up Robust Features (SURF) [2] keypoint detector approximates second-order derivative filters by using box filters. The drawback of these scale-invariant methods, however, is that they lose their invariance once anisotropic transformations occur. To overcome this limitation, detecting keypoints that are invariant under projective transformations are needed.

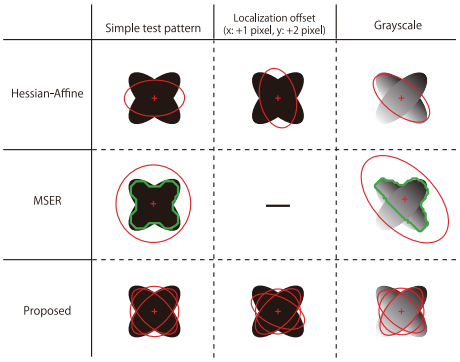

Figure 1. Comparison of multiple affine region estimation results. Left column: detection from binary image. Central column: keypoint localization offsets. Right column: detection from grayscale image.

Affine region estimation, which detects a region invariant to projective transformations, determines affine parameters $(\sigma_x, \sigma_y, \theta)$ by convolving anisotropic Gaussian or LoG filters. The Hessian-Affine and Harris-Affine methods [15] estimate an affine region by using the iterative convolution processing of an anisotropic Gaussian filter. This processing often fails, however, to detect corresponding affine regions between a pair of images when one of them suffers from image deformation or keypoint offset. Furthermore, in many cases, image textures can give multiple affine regions for a single keypoint. Conventional affine region detectors

are designed to find only one affine region, even if multiple regions exist and would fail to match keypoints properly between a pair of images.

A solution to this problem is to detect all affine regions for each keypoint by exhaustively convolving the LoG filters with all keypoints. The naive implementation of this scheme is intractable for most cases, though, because anisotropic LoG filters have three parameters $(\sigma_x, \sigma_y, \theta)$.

In this study, we efficiently calculate the response values of the anisotropic LoG filter by using the spectral SIFT [7]. The spectral SIFT detector solves eigenvalue problems for Gaussian scale space and LoG space and decomposes each set of filters into compact linear combinations of eigenfilters. It enables scale-continuous filter responses to be calculated from only a few of the eigenfilter convolution results. Spectral SIFT formulates the eigenproblem in the form of a continuous integral equation and solves it by approximating a discrete filter decomposition without an unreasonable amount of error. To approximate a large number of filters, the separable filter [24] uses tensor decomposition. In our proposed method we simply apply singular value decomposition (SVD) to the affine space, which is composed of several thousand different anisotropic LoG filters. We show that the affine space is well reconstructed from few (dozens) eigenfilters. This makes it tractable to compute all possible LoG filter responses. Furthermore, our proposed method can estimate multiple affine regions for a single keypoint by finding the multiple extrema of anisotropic LoG filter responses. We successfully estimated multiple affine regions under varying image conditions, as shown in Fig. 1.

2. Related works

A variety of corner detection methods have been proposed, including Moravec corner detection [17], Harris corner detection [5], Good Features to Track [23], Smallest Univalue Segment Assimilating Nucleus (SUSAN) [25], and Features from Accelerated Segment Test (FAST) [21]. Scale-space filtering has been studied since the 1980s [3]. Perona introduced the approximation of rotational and scale-space filters by SVD [20], which is based on the Steerable filter that uses a linear combination of rotation filters [4]. Scale-invariant keypoint detectors that generate isotropic support regions to compute descriptors have also been designed, such as LoG [10, 11], DoG [12], Hessian/Harris-Laplace [14] and SURF [2]. While these detectors compute scale-invariant keypoints using scale-space theory, some more recent methods simply build an image pyramid and apply a corner detector for each layer to cope with scale change [9, 22, 1, 6], which results in extremely fast keypoint detection. While there has been much research on estimating isotropic and scale invariant local regions, little work has been directed at estimating affine regions since the proposal of the Hessian/Harris-Affine method introduced by Mikolajczyk *et al.* [15]. In the following, we review conventional affine region estimation methods and present our method for estimating multiple affine regions.

2.1. Affine region detectors

Affine-invariant keypoint detection can be thought of as a generalization of scale estimation. The edge-based region (EBR) detector from Tuytelaars and colleagues treats a parallelogram region as an affine region in which each vertex is the point common to two edges crossing a corner determined by the Harris detector [26]. Tuytelaars *et al.* [27] also proposed the intensity extrema-based region (IBR) detector using intensity information in the periphery of a keypoint. The IBR method observes intensity values along rays emanating from a keypoint and determines the difference between those values and the intensity value of the keypoint. An affine region is determined by fitting an ellipse to the shape that joins the positions where that difference takes on a maximum value on each ray. The Maximally Stable Extremal Regions (MSER) approach [13] is another region estimation method, which uses a watershed algorithm to determine connected regions having similar intensity values within an image. It estimates affine-invariant regions by fitting ellipses to the regions so determined. Obdrzalek and Matas [19] extended the MSER approach to extract multiple regions based on parameters such as outline, curvature and so on.

Mikolajczyk *et al.* proposed Hessian-Affine and Harris-Affine [15] methods, which extend isotropic regions detected by Hessian/Harris-Laplace to affine regions. The procedure involves calculating a second-moment matrix with respect to an isotropic region detected by Hessian/Harris-Laplace. They determine an affine region by performing convolution operations on the keypoint while repeatedly varying the shape of an anisotropic Gaussian filter according to the second-moment matrix.

Affine SIFT (ASIFT) [18] generates a set of sample views of the initial image patches, obtained by varying the orientation of the two camera axes. Next, it applies the SIFT descriptor to all images thus generated. The method of Lepetit and Fua [8] also generates a set of all possible appearances under different viewing conditions; it trains randomized trees, using the set to recognize keypoints.

Each of the above methods determines only a single affine region per keypoint. This limitation often results in low repeatability; see Fig. 1. If the Hessian-Affine method (shown in the top row) starts from different initial points, as shown in the upper left and center of this figure, the affine estimation results converge to different locations. In contrast, our idea is to allow multiple affine regions, which can improve repeatability even in difficult situations, such as noisy localization and complicated shading, as shown in the bottom row of Fig. 1.

In this study, we propose a method for determining multiple affine regions by using the anisotropic LoG filter. A major challenge of this idea is that a very large number of anisotropic LoG filters must be convolved with a given image patch to exhaustively search the multiple extrema of filter responses. This requires a massive amount of computational time. Our finding is that the responses of all possible anisotropic LoG filters can be efficiently computed by factorizing them in a similar manner to spectral SIFT.

2.2. Contribution

We propose a method for accurately detecting affine regions from an image keypoint by using a combination of the eigenfilters of anisotropic LoG filters. Our method has the following features.

1. **Compaction of filter parameters by replacing anisotropic LoG filters with major eigenfilters**
   At the detection stage, only a small set of filters—which we call 'eigenfilters'—is used to convolve an image, instead of a large number of different anisotropic LoG filters.

2. **Accurate affine parameter interpolation**
   The eigenfunctions, produced by SVD of anisotropic LoG filters, are fitted by continuous functions, which can be used to interpolate accurate affine parameters.

3. **Detection of multiple affine regions**
   Even when two or more affine shapes are superimposed, the method accurately estimates the affine parameters for each shape, as is demonstrated in Fig. 1. Multiple region detection has not been very successful to date, although overlapping elliptical shapes occurs frequently in natural images.

3. Proposed method

We propose a method for estimating multiple affine regions based on Koutaki *et al.'s* [7] idea of spectral SIFT. Spectral SIFT derives eigensolutions for Gaussian space and LoG space expressed by

$$\begin{aligned} LoG(\sigma) &= \left(\frac{\partial^2}{\partial x^2}G(\sigma)+\frac{\partial^2}{\partial y^2}G(\sigma)\right)\sigma^2 \\ &= \frac{x^2+y^2-2\sigma^2}{2\pi\sigma^6}\sigma^2\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \quad (1) \\ G(\sigma) &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \quad (2) \end{aligned}$$

Equations (1) and (2) represent isotropic filters and thus cannot be used to estimate affine regions. Therefore, we employ anisotropic filters to determine affine regions. Determining the eigenfilters and eigenfunctions (i.e., the eigensolutions) of the anisotropic filters by SVD is an efficient means of defining the affine regions.

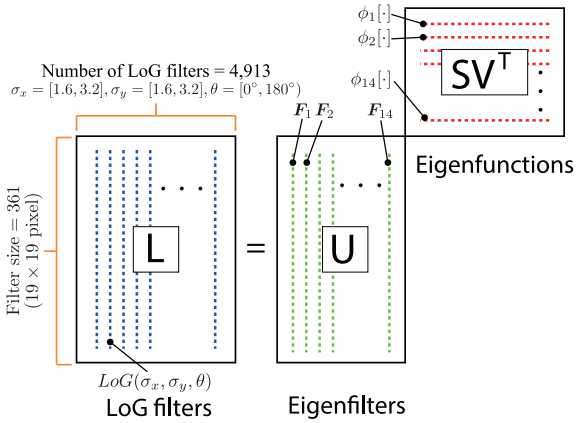

Figure 2. Approximation of anisotropic LoG filters by SVD. The LoG filter is defined by a linear combination of eigenfilters and eigenfunctions.

3.1. Approximation of anisotropic LoG space by SVD

We generate anisotropic LoG filters beforehand using:

$$LoG(\sigma_x, \sigma_y, \theta) = \frac{\partial^2}{\partial x^2}G(\mathbf{\Sigma}) + \frac{\partial^2}{\partial y^2}G(\mathbf{\Sigma}), \quad (3)$$

$$G(\mathbf{\Sigma}) = \frac{1}{2\pi\sqrt{\det\mathbf{\Sigma}}}\exp\left(-\frac{\mathbf{x}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{x}}{2}\right), \quad (4)$$

$$\mathbf{\Sigma} = \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\sigma_x^2 & 0 \\ 0 & \sigma_y^2\end{bmatrix}\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}, \quad (5)$$

where $G(\mathbf{\Sigma})$ denotes an anisotropic Gaussian filter and $\mathbf{x}$ denotes the position(x, y) from the center of the filter. The filter parameters are set as follows: scale σ_x in the x direction $\{1.6, 1.7, \cdots, 3.2\}$, scale σ_y in the y direction $\{1.6, 1.7, \cdots, 3.2\}$, and filter rotational angle θ $\{0°, 5°, \cdots, 175°\}$. The filter size D of an anisotropic LoG filter is 361 (19×19) and the number of different filters N is 4,913. These 4,913 anisotropic LoG filters can be decomposed into three matrices by applying SVD,

$$\mathbf{L} = \mathbf{USV}^{\mathrm{T}}, \quad (6)$$

where matrix $\mathbf{L} \in \mathbb{R}^{D\times N}$ consists of the above 4,913 LoG filters $\mathbf{L}(\sigma_x, \sigma_y, \theta) \in \mathbb{R}^D$ vectorized to form each column, matrices $\mathbf{U} \in \mathbb{R}^{D\times D}$ and $\mathbf{V} \in \mathbb{R}^{N\times N}$ are D-dimensional and N-dimensional orthogonal matrices, respectively, and matrix $\mathbf{S} \in \mathbb{R}^{D\times N}$ is a rectangular diagonal matrix having singular values. $\boldsymbol{F}_i \in \mathbb{R}^D$ and $\boldsymbol{P}_i \in \mathbb{R}^N$ are i-th column vectors of $\mathbf{U} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \cdots, \boldsymbol{F}_D]$ and $\mathbf{V} = [\boldsymbol{P}_1, \boldsymbol{P}_2, \cdots, \boldsymbol{P}_N]$ respectively. This SVD of anisotropic LoG filters is shown in Fig. 2. Now, denoting the row vectors of the product $\mathbf{SV}^{\mathrm{T}}$ as $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_D\}$, the anisotropic LoG filter $\mathbf{L}(\sigma_x, \sigma_y, \theta)$ for affine parameters $(\sigma_x, \sigma_y, \theta)$ can be defined as

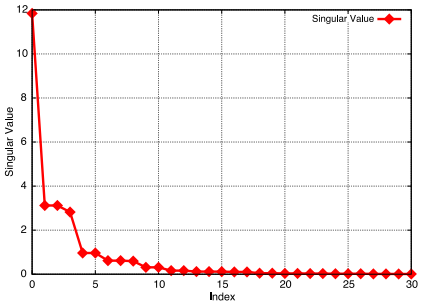

Figure 3. Singular values of diagonal matrix $\mathbf{S}$. From indices 15 and above, the singular values are very close to zero.

$$\mathbf{L}(\sigma_x, \sigma_y, \theta) = \sum_{n=1}^{D} \phi_n[\sigma_x, \sigma_y, \theta] \boldsymbol{F}_n \approx \sum_{n=1}^{14} \phi_n[\sigma_x, \sigma_y, \theta] \boldsymbol{F}_n. \quad (7)$$

After examining the singular values of matrix $\mathbf{S}$ (shown in Fig. 3), we take the major 14 eigenfilters only because the cumulative sum of their singular values reaches 96.7% of the total. This means that an anisotropic LoG filter can be adequately approximated by 14 eigenfilters. In the above, $\phi_n[\sigma_x, \sigma_y, \theta]$ denotes the scalar value of the vector $\boldsymbol{\phi}$ of the parameters $\sigma_x, \sigma_y, \theta$. It is referred to here as an eigenfunction. In addition, $\boldsymbol{F}_n$ can be treated as a two-dimensional filter and is therefore an eigenfilter. All 14 eigenfilters and eigenfunctions are shown in Fig. 4. An eigenfunction is a value in the $(\sigma_x, \sigma_y, \theta)$ 3D space. The eigenfunctions shown in the figure are for $\theta = 45°$.

3.2. Calculation of response value of an anisotropic LoG filter

We now calculate the response value R of an anisotropic LoG filter using eigenfilter $\boldsymbol{F}_n$ and eigenfunction $\phi_n[\cdot]$. The response value R of an anisotropic LoG filter can be calculated by performing a convolution operation between Eq. (7) and patch image I, which is the image corresponding to the keypoint neighborhood (19×19). Here, the result of convolution between the patch image and an eigenfilter is given by $q_n = I * \boldsymbol{F}_n$. Applying the distributive law to Eq. (8) gives

$$R(\sigma_x, \sigma_y, \theta) \approx I * \sum_{n=1}^{14} \phi_n[\sigma_x, \sigma_y, \theta] \boldsymbol{F}_n \quad (8)$$

$$\approx \sum_{n=1}^{14} \phi_n[\sigma_x, \sigma_y, \theta] q_n. \quad (9)$$

Convolution between the patch image and the 14 eigenfilters can be performed in advance as in Eq. (9). The response values for a desired filter can be easily calculated by simply changing eigenfunction values. In our approach, the computationally expensive convolution processing only has to be performed 14 times, making it efficient. The workflow for calculating the response value of an anisotropic LoG filter for parameters $\sigma_x = 1.4, \sigma_y = 1.6, \theta = 45°$ is shown in Fig. 5.

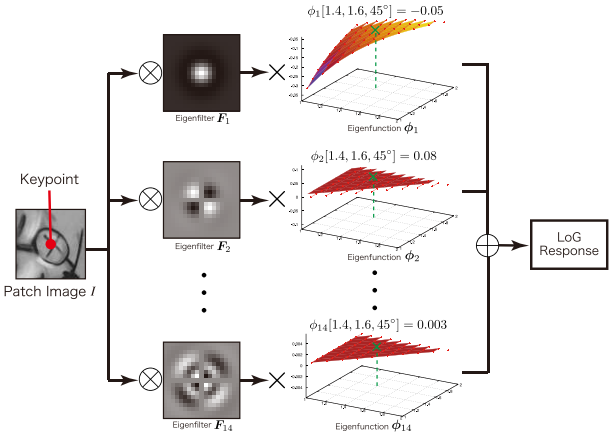

Figure 5. Workflow of the anisotropic LoG filter response calculation; parameter values: $\sigma_x = 1.4, \sigma_y = 1.6, \theta = 45°$.

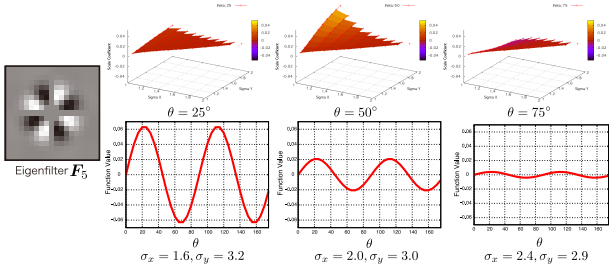

Figure 6. Examples of eigenfunctions with fixed parameters. The upper plots are the eigenfunction values for each θ. The eigenfunctions for fixed σ_x, σ_y are shown in the lower plots.

3.3. Continuous function fitting of eigenfunctions

The response values of 4,913 filters can be approximated by 14 eigenfunctions using Eq. (9). These eigenfunctions, however, take on discrete values, thus the response values can only be approximated for the 4,913 combinations of filter parameters generated before decomposition. We consider the fitting of eigenfunctions to continuous functions to solve this problem. As shown in Fig. 4, eigenfunction $\phi[\cdot]$ is a value in a 3D parameter space. The task, therefore, is to solve the error minimization problem between eigenfunction $\phi[\cdot]$, obtained by SVD, and the fitting model $\varphi(\cdot)$, based on a 3D function. The fitting model uses a continuous function in power series form for the scale parameters in question. However, the eigenfunction for rotational angle θ is a periodic function waveform that considers the shape of the eigenfilter. Numerical values of eigenfunction $\boldsymbol{\phi}_5$ are shown in Fig. 6. Eigenfilter $\boldsymbol{F}_5$ changes shape according to the filter's rotational angle θ. This causes the eigenfunction

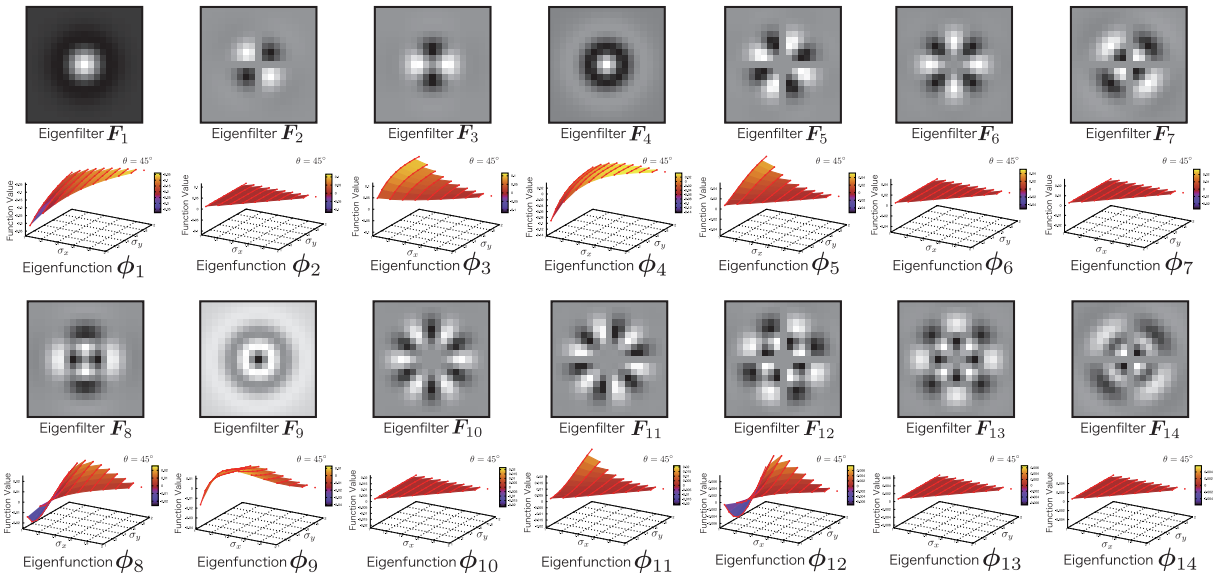

Figure 4. Eigenfilters and eigenfunctions. The eigenfunctions have $\theta = 45°$.

values also to change according to θ (upper part of Fig. 6). Furthermore, the eigenfunctions become periodic when fixing parameters σ_x, σ_y (lower part of Fig. 6). For this reason, the eigenfunction fitting model uses a continuous function consisting of trigonometric functions of θ. Accordingly, fitting model $\varphi(\cdot)$ takes the form

$$\varphi(\sigma_x, \sigma_y, \theta) = \sum_{i=0}^{e}\sum_{j=0}^{f}\sum_{k=1}^{g}(\alpha_{ijk}\sigma_x^i\sigma_y^j \sin k\theta + \beta_{ijk}\sigma_x^i\sigma_y^j \cos k\theta), \quad (10)$$

where α and β are unknown coefficients for which the square error E between the fitting model $\varphi(\cdot)$ and discrete eigenfunction $\phi[\cdot]$ (obtained by SVD) is minimal;

$$E = \sum_{\sigma_x}\sum_{\sigma_y}\sum_{\theta}(\phi[\sigma_x, \sigma_y, \theta] - \varphi(\sigma_x, \sigma_y, \theta))^2. \quad (11)$$

Fitting eigenfunctions in this way enables the filter response value R to be represented as a continuous function,

$$R(\sigma_x, \sigma_y, \theta) \approx \sum_{n=1}^{14}\varphi_n(\sigma_x, \sigma_y, \theta)q_n. \quad (12)$$

This makes it possible to calculate filter response values for any parameters $(\sigma_x, \sigma_y, \theta)$. By applying the gradient method, it is possible to search for the extrema of response values efficiently because the continuous function of the eigenfunction $\varphi(\cdot)$ is differentiable. The process of searching for the extrema is described next.

3.4. Estimation of multiple affine regions

Given that multiple affine regions exist in an input image, there will also be multiple instances of the parameter combination $(\sigma_x, \sigma_y, \theta)$ at which a filter response value is an extremum. It is therefore necessary to detect multiple extrema in filter response values in the 3D parameter space. For the ellipsoidal patterns in Fig. 1, one affine region is determined with respect to the scale directions σ_x, σ_y. In the θ direction, however, the existence of multiple affine regions can be considered. For this situation, therefore, we determine one affine region in the σ_x, σ_y directions and estimate multiple affine regions in the θ direction. The method used here to search for extrema is illustrated in Fig. 7. First, after discretizing the θ axis, Newton's method is employed to search for different θ in the extrema in the 2D space (σ_x, σ_y) (denoted by a black $\times$-mark in the figure). Next, using these 2D extrema for different θ, extrema are detected in the 3D space $(\sigma_x, \sigma_y, \theta)$ (denoted by a blue $\circ$-mark in the figure). When there are more than one extremum in 3D space, all extrema with a response value that is close to the largest extremum are adopted as the affine regions of the keypoint. In the example of Fig. 7, three extrema exist in 3D space but only two extrema are used as the affine regions. The results of keypoint affine estimation using our proposed method are shown in Fig. 8. These results demonstrate that the proposed method determines multiple affine regions for a single keypoint. It is also clear that these affine regions are the same as for a pair of images.

3.5. Simple pattern testing

In Fig. 9, the left part of the simple test pattern in each subfigure depicts ellipses and intersecting ellipses drawn in black, while the right part depicts the same shapes after gradation processing. The parameter range of each ellipse is 25-30 pixels for the major axis, 16-19 pixels for the minor axis, and $0° - 60°$ for the rotational angle. As shown in Fig.

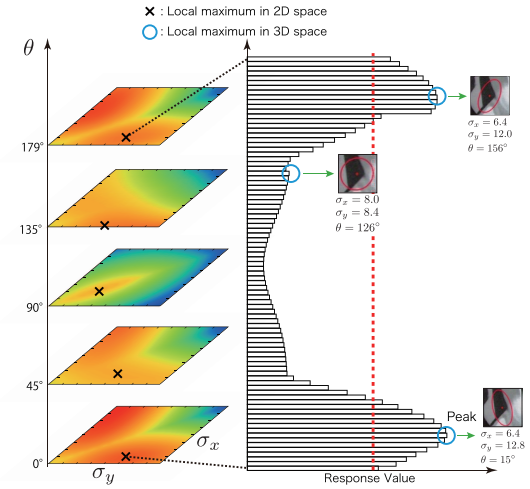

Figure 7. Searching for a multiple-hypothesis affine region.

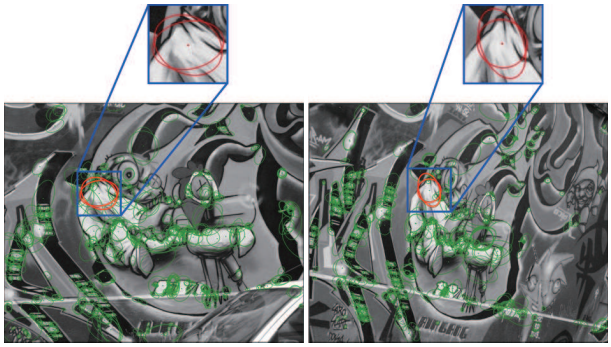

Figure 8. Result of affine regions estimation with the proposed method. Red affine regions are estimated to be the same region in each of two images.

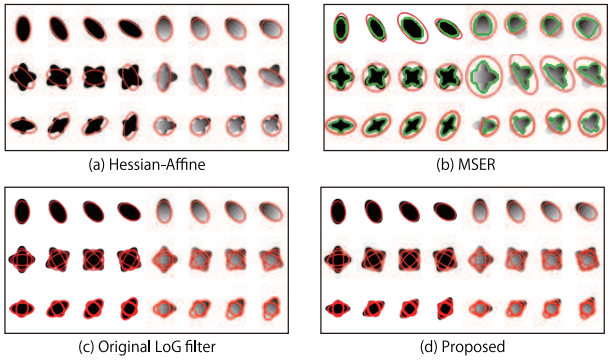

Figure 9. Results of the affine region estimation for a simple test pattern. The input pattern depicts simple ellipses and intersecting ellipses drawn in black and gradation.

9(a), the Hessian-Affine method can detect simple ellipses but cannot correctly detect intersecting ellipses. Thus, for several elliptical shapes, this method detected different ellipses between the left and right parts of the test pattern. The MSER method, by comparison, fits ellipses using region partitioning and therefore predicted elliptical patterns surrounding the combined shapes. It is insensitive to changes in intensity and consequently failed in detecting the elliptical patterns in the right part of the test pattern (Fig. 9(b)). The proposed method, however, detected all ellipses—even those in an intersecting pattern, as it is designed to estimate multiple affine regions. The regions that it detected in the left part of the test pattern were thus the same as those in the right part, as shown in Fig. 9(d). The results of estimating affine regions by using the original LoG filter prior to SVD are shown in Fig. 9(c). It can be seen that these results agree with those produced by the proposed method, thus the proposed method approximates the original LoG filter.

4. Experiments

This experiment evaluates the repeatability of affine-region detection between two images and compares the results of the proposed method, Hessian-Affine, MSER, and DoG.

4.1. Datasets

For this evaluation experiment, we used conventional Affine Covariant Regions Datasets[1] and Spectrum magazine datasets created by us. The Affine Covariant Regions Datasets consist of image datasets covering eight scenes (Graffiti, Wall, Boat, Bark, Bikes, Trees, Leuven and UBC). Each scene comprises six images, each of which represents a certain type of change in the appearance of that scene. These changes in appearance are summarized in Table 1. The Spectrum magazine datasets consist of image

Table 1. Change in appearance for each scene type.

Change in appearance	Scene
Projective transformation	Graffiti, Wall
Rotation and scale change	Boat, Bark
Blur	Bikes, Trees
Other	Leuven, UBC

datasets from three magazines (Spectrum1, Spectrum2 and Spectrum3). Each magazine has image sets corresponding to pitch angle rotations of $0°, 10°, 20°$ and $30°$, and each pitch angle set consists of 73 images having rotations in roll angle. Examples of images in the Spectrum1 dataset are shown in Fig. 10. Homography matrices $\boldsymbol{H}$ between image pairs were determined beforehand for the above datasets.

4.2. Repeatability

Repeatability is calculated as the ratio of the number of corresponding points ($corresp.\ points$) in a pair of images to the minimum number of keypoints ($keypoints1, keypoints2$) detected in those two images;

[1] http://www.robots.ox.ac.uk/~vgg/data/data-aff.html

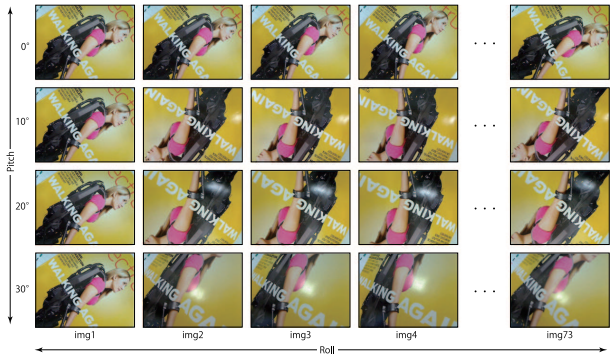

Figure 10. Example from the Spectrum magazine dataset.

i.e.,

$$Repeatability = \frac{corresp.\ points}{\min(keypoints1, keypoints2)} \times 100. \quad (13)$$

When determining corresponding points, we calculate the error between the two images in the overlapping of the corresponding keypoint affine regions [16] by using the homography matrix $\boldsymbol{H}$,

$$overlap\ error = \left(1 - \frac{A_1 \cap \boldsymbol{H}_i^{\mathrm{T}} A_2 \boldsymbol{H}_i}{A_1 \cup \boldsymbol{H}_i^{\mathrm{T}} A_2 \boldsymbol{H}_i}\right) \times 100, \quad (14)$$

where A_1 and A_2 are keypoint affine regions detected in the two images and $\boldsymbol{H}_i$ is a linear homography matrix. If the overlap error of corresponding affine regions is less than the threshold T, the corresponding points are counted. In this experiment, we set $T = 40\%$ and compared repeatability among the proposed method, Hessian-Affine, MSER, and DoG. To put the evaluations of the proposed method and Hessian-Affine on equal footing, we used the same keypoints used in the estimation of affine regions in both methods.

4.3. Experiment results

The repeatability results obtained when using the Affine Covariant Regions Datasets and Spectrum magazine datasets are shown in Figures 11 and 12, respectively. The results show that the proposed method has higher repeatability than do the conventional methods. We attribute this to the proposed method estimating multiple affine regions, thereby reducing error in the estimation of affine regions between a pair of images. Thus the affine region estimation employed in the proposed method exhibits high accuracy.

For the Affine Covariant Regions Datasets, these results also show that repeatability improved by using the proposed method even for datasets that did not involve a projective transformation (namely, Boat, Bark, Bike, Trees, Leuven, and UBC). This indicates that the proposed method was capable of determining invariant affine regions for various types of changes in appearance. Examples of keypoint matching using SIFT descriptors with the Hessian-Affine and the proposed method are shown in Fig. 13. It is clear that a high matching rate can be achieved with the proposed method.

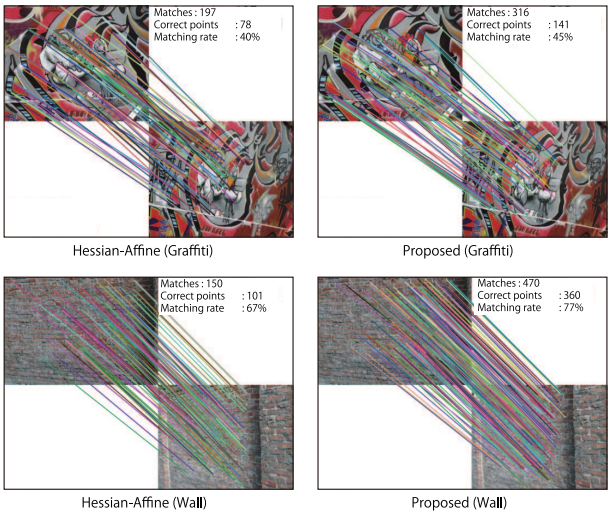

Figure 13. Examples of keypoint matching by Hessian-Affine and proposed method.

4.4. Computational time

We compare the processing times per image (640×480 pixels) of the Hessian-Affine, Original LoG, and proposed method. The computer used in the experiment had a 3.33-GHz Intel Xeon X5470 CPU and 32 GB of RAM. All code was written in C++. The results of the experiment are listed in Table 2. The proposed method was 87.2 times faster than the Original LoG. This was possible because it approximates the convolution processing of 4,913 filters with only 14 eigenfilters.

Table 2. Computational time (s) per image (640×480 pixels).

	Hessian-Affine	Original LoG	Proposed
Total time	4.091	198.654	2.277

5. Conclusion

We proposed a method for estimating multiple-hypothesis affine regions using anisotropic LoG filters. This method efficiently calculates the response values of anisotropic LoG filters by applying singular value decomposition. We showed that estimating multiple affine regions for each keypoint in an image improves repeatability. The proposed method also represents filter response values in continuous function form, enabling response values for any parameter to be found. These features make for high-accuracy affine region estimation. In future work, we plan to increase the efficiency of extrema searching to increase the speed of high-speed affine region estimation.

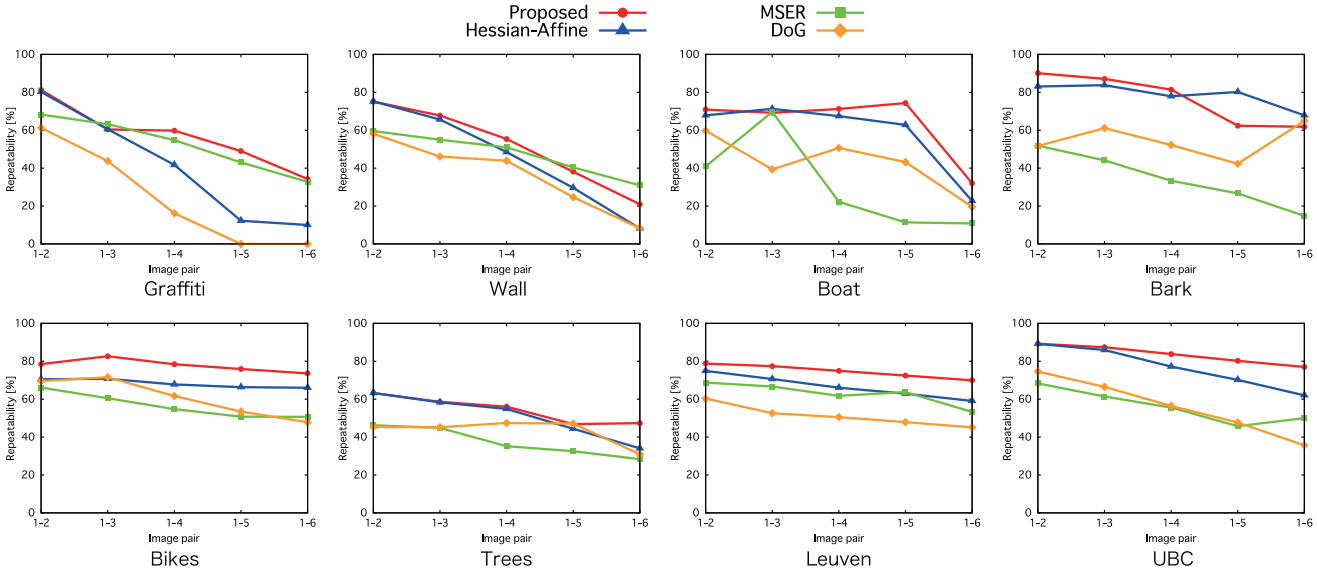

Figure 11. Repeatability of the Proposed, Hessian-Affine, MSER, and DoG methods operating on the Affine Covariant Regions Datasets.

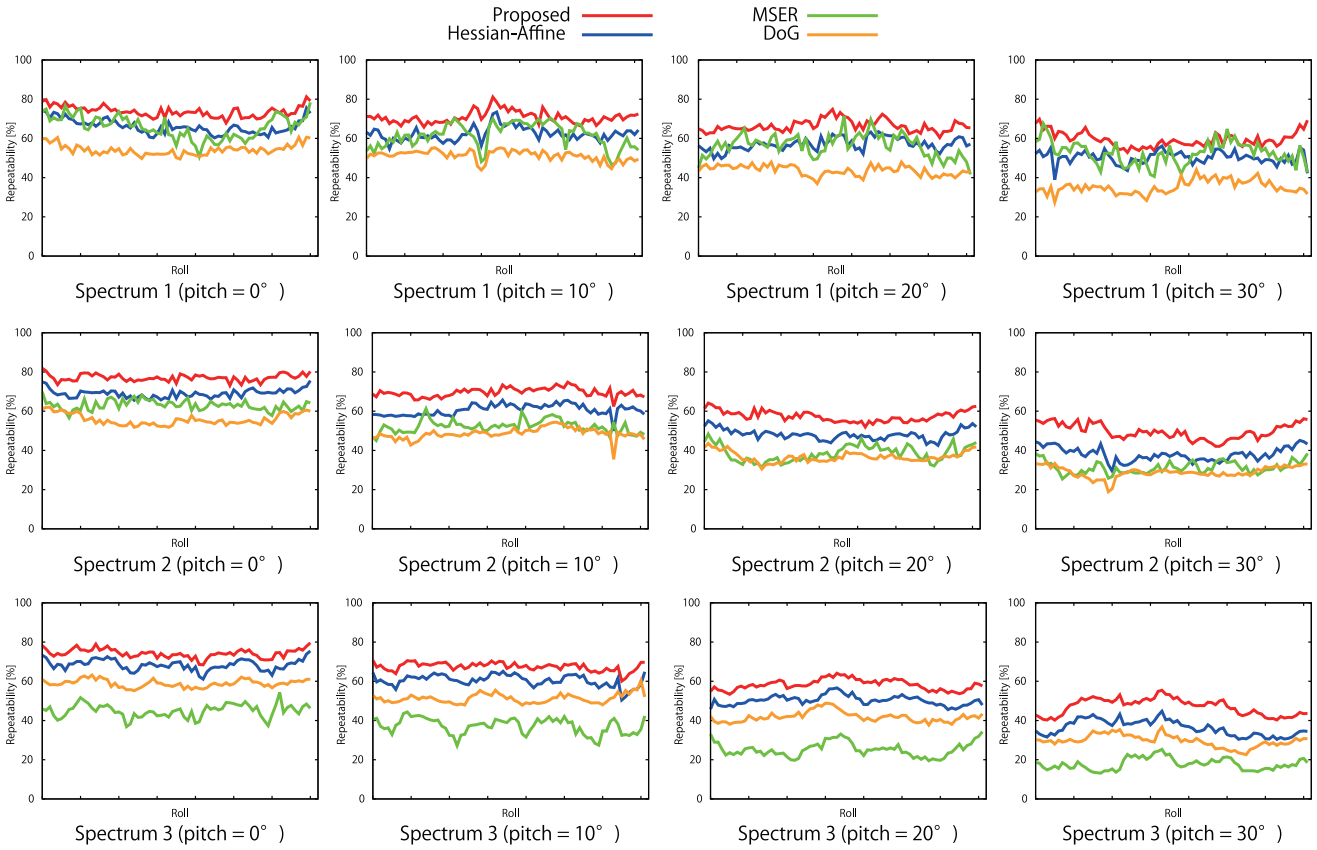

Figure 12. Repeatability of the Proposed, Hessian-Affine, MSER, and DoG methods operating on the Spectrum magazine datasets.

References

[1] M. Ambai and Y. Yoshida. Card: Compact and real-time descriptors. In *International Conference on Computer Vision*, pages 97–104, 2011.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded-Up Robust Features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[3] J. Crowley and A. Parker. A representation for shape based

on peaks and ridges in the difference of low pass transform. *Pattern Analysis and Machine Intelligence*, 6(2):156–170, 1984.

[4] W. Freeman and E. Adelson. The design and use of steerable filters. *Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[6] T. Hasegawa, Y. Yamauchi, M. Ambai, Y. Yoshida, and H. Fujiyoshi. Keypoint Detection by Cascaded FAST. In *International Conference on Image Processing*, pages 5611–5615, 2014.

[7] G. Koutaki and K. Uchimura. Scale-space Processing Using Polynomial Representations. In *Conference on Computer Vision and Pattern Recognition*, pages 2744–2751, 2014.

[8] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[9] S. Leutenegger, M. Chli, and R. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *International Conference on Computer Vision*, pages 2548–2555, 2011.

[10] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, pages 224–270, 1994.

[11] T. Lindeberg. Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30:79–116, 1998.

[12] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *British Machine Vision Conference*, pages 36.1–36.10, 2002.

[14] K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Points. In *International Conference on Computer Vision*, pages 525–531, 2001.

[15] K. Mikolajczyk and C. Schmid. Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[17] H. Moravec. Rover visual obstacle avoidance. In *International Joint Conference on Artificial Intelligence*, pages 785–790, 1981.

[18] J.-M. Morel and G. Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[19] S. Obdrzalek and J. Matas. Object Recognition Using Local Affine Frames on Maximally Stable Extremal Regions. In *Toward Category-Level Object Recognition*, pages 85–108, 2006.

[20] P. Perona. Deformable Kernels for Early Vision. *Pattern Analysis and Machine Intelligence*, 17:488–499, 1991.

[21] E. Rosten, R. Porter, and T. Drummond. FASTER and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32:105–119, 2010.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *International Conference on Computer Vision*, pages 2564–2571, 2011.

[23] J. Shi and C. Tomasi. Good Features to Track. In *Conference on Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.

[24] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua. Learning Separable Filters. *Pattern Analysis and Machine Intelligence*, pages 94–106, 2015.

[25] S. M. Smith and J. M. Brady. Susan & mdash; a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[26] T. Tuytelaars and L. V. Gool. Content-based Image Retrieval based on Local Affinely Invariant Regions. In *International Conference on Visual Information Systems*, pages 493–500, 1999.

[27] T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference*, pages 412–425, 2000.