Peer-reviewed version available at *Information* **2019**, *10*, 376; doi:10.3390/info10120376

Article

A Global Extraction Method of High Repeatability on Discretized Scale-space Representations

Qingming Zhang[1,2,*] and Buhai Shi[1,*]

[1] School of Automation Science and Engineering, South China University of Technology, Guangzhou, Guangdong, China, 510641

[2] School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, Guizhou, China, 550025

* Correspondence: autox2015@mail.scut.edu.cn; bhshi@scut.edu.cn

Abstract: This paper presents a novel method to extract local features, which instead of calculating local extrema computes global maxima in a discretized scale-space representation. To avoid obtaining precise scales by interpolation and to achieve perfect rotation invariance, two essential techniques, increasing the width of kernels in pixel and utilizing disk-shaped convolution template are adopted in this method. Since the size of a convolution template is finite and finite templates can introduce computational error into convolution, we sufficiently discuss this problem and work out an upper bound of the computational error. The upper bound is utilized in the method to ensure that all features obtained are computed under a given tolerance. Besides, the technique of relative threshold to determine features is adopted to reinforce the robustness for the scene of changing illumination. Simulations show that this new method attains high performance of repeatability in various situations including scale change, rotation, blur, JPEG compression, illumination change and even viewpoint change.

Keywords: local feature extraction; scale-space representation; Laplacian of Gaussian; convolution template

1. Introduction

Local feature extraction is a fundamental technique for solving problems of computer vision, such as matching, tracking and recognition. A local feature is a structure around a point in an image, and its size, which relates to the scale, is usually unknown before it is extracted. Traditional Harris corner detector [1] does not consider the variance of scale, which accounts for a drawback that it cannot be applied to matching features with different scales. For detecting corners with different resolutions, Dufournaud [2] discusses a scale-invariant approach based on the Harris detector, which adopts the Gaussian kernel with width σ and uses a variable s as the scale factor. Therefore $s\sigma$ represents an arbitrary scale, by which corner features with different scales are detected by traditional Harris corner detector. Scale-invariant properties are systematically studied by Lindeberg. Introducing a normalized derivative operator [3] into the scale-space theory [4], Lindeberg presents a framework for automatic scale selection, pointing out that a local maximum of some combination of normalized derivatives over scale reflects a characteristic length of a corresponding structure, and has a nice behaviour under rescaling of the intensity pattern [3], which has been a principle for solving problems of feature extraction. In SIFT [5,6], Lowe presents the Difference-of-Gaussian (DoG) method on image pyramids, which is an inchoate type of multi-scale representation, to approximate Laplacian of Gaussian (LoG). Mikolajczyk presents a Harris-Laplacian method [7], which uses Harris functions of images in scale-space representation to extract interesting points and then invokes Laplacian to

Peer-reviewed version available at Information 2019, 10, 376; doi:10.3390/info10120376

select feature points as well as their precise scale parameters. This method afterwards is extended to an affine-adapted approach [8]. Aiming at reducing computation time, Bay introduces integral images and box filters and works out SURF [9,10]. Because of the techniques of integral images and box filters, Hessian feature detector in SURF is revised into Fast-Hessian detector which can be computed more quickly than the former. Recently, Lomeli-R and Nixon present a feature detector, the Locally Contrasting Keypoints detector (LOCKY) [11,12], which extracts blob keypoints directly from the Brightness Clustering Transform (BCT) of an image. The BCT also exploits the technique of integral images, and performs a fast search through different scale spaces by the strategy of coarse-to-fine.

In the extractors mentioned above, features are extracted through comparison amid its immediate neighbours, which are in the image and two other adjacent images. We here name this methodology of these extractors as Local-Prior Extraction (LPE). Due to the exponential growth of scale parameters usually adopted by LPE (which are too coarse to locate features precisely at the scale axis) the LPE needs interpolation or other refining procedures to obtain precise scales. A great advantage of LPE is the relatively low cost of computation, which enables LPE to be broadly applied to numerous extractors. However, the repeatability of features obtained by LPE is yet to be improved. We alternatively study a novel method which instead of LPE, extracts features in a discretized scale-space representation that has been constructed in advance, and name this new method as Global-Prior Extraction (GPE). The rest of this paper is organized as follows. In section 2, we give a brief introduction for GPE. In section 3, we present an approach to compute feature responses in a discretized scale-space representation and to represent these responses by a 3-dimensional array. In section 4, we carry out a method for finding local features in the array to achieve the extraction of features. In section 5, we test the algorithm of GPE and compare results with some classical extractors. We conclude our work in section 6.

2. Sketch of GPE

Suppose $f(u,v)$ $(u,v \in \mathbb{R})$ to be a 2-dimensional signal and $K(\cdot;t)$ $(t \in \mathbb{R}_+)$ to be a kernel with width $\sqrt{t}$. Then the scale-space representation of $f(u,v)$ is (cf. [4])

$$\begin{cases} L(u,v;0) = f(u,v), \\ L(\cdot,\cdot;t) = K(\cdot,\cdot;t) * f. \end{cases} \tag{1}$$

Using the scale-normalized derivative $\mathcal{D}$, the scale space (1) can be transformed to (cf. [3])

$$\mathcal{D}L(\cdot,\cdot;t) = \mathcal{D}K(\cdot,\cdot;t) * f. \tag{2}$$

Assume that $\mathcal{K}(\cdot\,;\cdot)$ is an appropriate kernel that sensitively responds to a certain class of features, which is applied as a detector for some kinds of features. Then in any bounded open set $\Omega \subset \mathbb{R}^2 \times \mathbb{R}_+$ the following expression

$$(x,y;\tau) \in \underset{(u,v;t)\in\Omega}{\operatorname{argmax}}\, \mathcal{D}L^2(u,v;t), \tag{3}$$

represents a maximal responding position of both spatial space and scale space and therefore is an extremum of $L(u,v;t)$. This extremal point is a feature point in the signal $f(x,y)$, and has the scale-invariant property.

An iterative procedure can be proposed to work out scale-invariant features in $f(x,y)$. Let $F = \phi$ be the initial set of feature points, and $(x_i,y_i;\tau_i)$ be the i-th feature points calculated through (3). Denote by U_i a neighbourhood of the point $(x_i,y_i;\tau_i)$. Put $\Omega_i = \Omega \setminus \cup_{k=1}^{i-1} U_k$ (Obviously, $\Omega_i = \Omega_{i-1} \setminus U_{i-1}$). The $(i+1)$-th feature point can be computed through steps as follows.

- Compute the point of maximal response in Ω_i through (3);
- Update the set $\Omega_{i+1} = \Omega_{i-1} \setminus U_{i-1}$.

Peer-reviewed version available at Information 2019, 10, 376; doi:10.3390/info10120376

Repeatedly executing the two steps above, we can obtain a set $\{(x_i, y_i; \tau_i)\}_{i=1}^{N}$ for choosing features, where N is the times of iterations, and $(x_1, y_1; \tau_1)$ is obtained from $\Omega_1 := \Omega$.

In contrast to LPE, GPE does not detect features during the procedure of generating scaled images, but instead detects features in a discretized scale-space representation constructed beforehand. Therefore there are two stages essential in GPE: (1) constructing a discrete scale-space representation and transforming it properly; (2) obtaining maxima iteratively in this transformed discrete scale-space representation.

3. Discretization and transformation of scale-space representations

The natural structure imposed on a scale-space representation is a semi-group, and namely, the kernels should satisfy $K(\cdot; t_1) * K(\cdot; t_2) = K(\cdot; t_1 + t_2)$ [4]. For retaining the semi-group structure within some range in scale when discretizing a scale-space representation, one can sample scales equidistantly from the scale space. However, a computer image $f(x, y)$ $(1 \le x \le c, 1 \le y \le d; x, y, c, d \in \mathbb{Z}_+)$ can be regarded as a sample drawn equidistantly from a given 2-dimensional signal $f(u, v)$ $(u, v \in \mathbb{R})$. The domain of $f(x, y)$ therefore consists of finitely many pixels. Considering the computation of discrete convolution and its cost, we alternatively employ pixel as the unit for the width of kernels, and then determine sampling intervals on the scale space by these widths. We here call the kernel width in pixel as the *pixel scale*. When increasing the width of kernels by a single pixel each time, a sequence of samples with pixel scale $1, 4, 9, \cdots, N^2$ (where N is the maximal width of kernels used in computation), can be drawn from a scale-space representation. In contrast to multiplying the original scale, the preference of increasing scale by adding pixels rids GPE of interpolating scale values as many LPE extractors do.

3.1. Choice of an appropriate kernel

To choose a suitable kernel for our method, we consider the normalized LoG

$$\nabla^2_{norm} G = G^{norm}_{xx} + G^{norm}_{yy}, \tag{4}$$

where

$$\begin{aligned} G^{norm}_{xx}(x, y) &:= \sigma^2 G_{xx}(x, y) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\left(\frac{x^2}{\sigma^2} - 1\right) e^{-\frac{x^2+y^2}{2\sigma^2}}, \\ G^{norm}_{yy}(x, y) &:= \sigma^2 G_{yy}(x, y) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\left(\frac{y^2}{\sigma^2} - 1\right) e^{-\frac{x^2+y^2}{2\sigma^2}}, \end{aligned}$$

and $G(\cdot; \cdot)$ is the Gaussian kernel.

The LoG is preferable due to its excellent performance on scale-space feature detecting. Mikolajczyk pointed out that the LoG is the most efficient one to draw interesting points over a scale space in contrast to operators such as DoG, Gradient and Harris [7]. Moreover, the LoG operator (4) is a strict rotation-invariant integral kernel when the integral region is a finite disk. The rotation invariance will be justified by the following reasoning.

Suppose that $\boldsymbol{A}$ is a 2-by-2 orthogonal matrix and $\boldsymbol{\xi}$ is a vector in $\mathbb{R}^2$. It is obvious that

$$\nabla^2_{norm} G(\boldsymbol{\xi}) = \nabla^2_{norm} G(\boldsymbol{A}\boldsymbol{\xi}).$$

Peer-reviewed version available at Information 2019, 10, 376; doi:10.3390/info10120376

Consider two signals f and f' related by $f(\boldsymbol{\xi}) = f'(\boldsymbol{A}\boldsymbol{\xi})$. Then on a disk $\mathcal{D}$ with center $\boldsymbol{c}$, we have

$$\begin{aligned}
&\int_{\mathcal{D}} \nabla^2_{norm} G(\boldsymbol{\xi}-\boldsymbol{c}) f(\boldsymbol{\xi}) \sqrt{\mathrm{d}\boldsymbol{\xi}^T \mathrm{d}\boldsymbol{\xi}} \\
&= \int_{\mathcal{D}} \nabla^2_{norm} G(\boldsymbol{A}(\boldsymbol{\xi}-\boldsymbol{c})) f'(\boldsymbol{A}\boldsymbol{\xi}) \sqrt{\mathrm{d}(\boldsymbol{A}\boldsymbol{\xi})^T \mathrm{d}(\boldsymbol{A}\boldsymbol{\xi})} \\
&= \int_{\mathcal{D}'} \nabla^2_{norm} G(\boldsymbol{\eta}-\boldsymbol{A}\boldsymbol{c}) f'(\boldsymbol{\eta}) \sqrt{\mathrm{d}\boldsymbol{\eta}^T \mathrm{d}\boldsymbol{\eta}},
\end{aligned} \tag{5}$$

where $\mathcal{D}'$ is a disk centred at $\boldsymbol{A}c$ with radius identical to that of the disk $\mathcal{D}$.

Under the scale-invariant framework, many feature detectors can be modified into scale-invariant detectors. However some of them, such as the determinant of the Hessian, the Gradient, and the Harris are not rotation-invariant on such a disk region because $G_{xy}(\boldsymbol{\xi}) \neq G_{xy}(\boldsymbol{A}\boldsymbol{\xi})$, $G_x(\boldsymbol{\xi}) \neq G_x(\boldsymbol{A}\boldsymbol{\xi})$ and $G_y(\boldsymbol{\xi}) \neq G_y(\boldsymbol{A}\boldsymbol{\xi})$.

3.2. Size of convolution templates

To compute the convolution of a kernel with a computer image, it should be discretized into a bounded template. By our foregoing results, the templates for LoG in GPE should be disks with certain radii. Denote by r_T the radius of a LoG template utilized in GPE. We discuss how to determine the radius r_T.

Suppose the current scale to be σ^2. For a given signal $f(u,v)$, it follows that

$$\begin{aligned}
L(u,v;\sigma^2) =& \frac{1}{\sqrt{2\pi}\sigma} \iint_{\mathbb{R}^2} \left(\frac{x^2+y^2}{\sigma^2} - 2\right) e^{-\frac{x^2+y^2}{\sigma^2}} f(x-u, y-v)\mathrm{d}x\mathrm{d}y \\
=& \frac{1}{\sqrt{2\pi}\sigma} \iint_{\mathbb{R}^2} \left(\frac{r^2}{\sigma^2} - 2\right) e^{-\frac{r^2}{\sigma^2}} f(r\cos\theta - u, r\sin\theta - v) r\mathrm{d}r\mathrm{d}\theta \\
=& \frac{1}{\sqrt{2\pi}\sigma} \int_0^{\infty} r\left(\frac{r^2}{\sigma^2} - 2\right) e^{-\frac{r^2}{\sigma^2}} \int_0^{2\pi} f(r\cos\theta - u, r\sin\theta - v)\mathrm{d}\theta\mathrm{d}r.
\end{aligned}$$

When $r > 4\sigma$, the function $g(r) = r(\frac{r^2}{\sigma^2} - 2)e^{-\frac{r^2}{\sigma^2}}$ is monotonically decreasing. It is easy to know that

$$0 < g(r) \cdot I_{\{r>4\sigma\}} < 56\sigma e^{-\frac{r^2}{\sigma^2}}.$$

Let

$$e(4\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \int_{4\sigma}^{\infty} r\left(\frac{r^2}{\sigma^2} - 2\right) e^{-\frac{r^2}{\sigma^2}} \int_0^{2\pi} f(r\cos\theta - u, r\sin\theta - v)\mathrm{d}\theta\mathrm{d}r.$$

Then we have

$$e(4\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \int_{4\sigma}^{\infty} h(r)g(r)\mathrm{d}r < \frac{1}{\sqrt{2\pi}\sigma} \int_{4\sigma}^{\infty} 56\sigma e^{-\frac{r^2}{\sigma^2}} h(r)\mathrm{d}r,$$

where $h(r) = \int_0^{2\pi} f(r\cos\theta - u, r\sin\theta - v)\mathrm{d}\theta$. Considering that the maximal gray level is 256, we further have

$$e(4\sigma) < 56 \times 256\sqrt{2\pi} \int_{4\sigma}^{\infty} e^{-\frac{r^2}{\sigma^2}} \mathrm{d}r < 3584\sigma\pi\sqrt{2\pi}e^{-16}. \tag{6}$$

Peer-reviewed version available at *Information* **2019**, *10*, 376; doi:10.3390/info10120376

Then we set

$$v(4\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\int_0^{4\sigma} r(\frac{r^2}{\sigma^2}-2)e^{-\frac{r^2}{\sigma^2}}\int_0^{2\pi} f(r\cos\theta-u, r\sin\theta-v)\mathrm{d}\theta\mathrm{d}r,$$

and estimate $\mathcal{D}L(u,v,;\sigma)$ by $v(4\sigma)$. Inequality (6) gives an upper bound of the error in this estimation. Utilizing this upper bound, we can preclude points not satisfying the tolerance of computation error from feature candidates. Therefore, we introduce a relative error threshold α, and construct a threshold for feature response:

$$\beta = \frac{3584\tilde{\sigma}\pi\sqrt{2\pi}e^{-16}}{\alpha}, \tag{7}$$

where $\tilde{\sigma}$ is the maximal width of kernels used in the computation of drawing features. Hence the function to determine features is:

$$\rho(u,v;\sigma) = 1\cdot I_{\{\mathcal{D}L^2(u,v,;\sigma)\geqslant\beta^2\}} + 0\cdot I_{\{\mathcal{D}L^2(u,v,;\sigma)<\beta^2\}},$$

and the maximal point $(u,v;\sigma)$ is a feature point if and only if $\rho(u,v;\sigma)=1$.

In summary, we set the radius of the convolution template in GPE as $r_T = 4\sigma$, and introduce a relative error threshold α to ensure that all features obtained are computed under a given tolerance.

3.3. Algorithm for discretizing and transforming scale-space representations

The general idea of discrtetizing and transforming a scale-space representation is to produce a sequence of smoothed images, which are obtained through convolution between the original image and a series of LoG templates with increasing widths. The criterion to stop the process is the maximal width of kernels, which should be set in advance. A pseudo-code for this algorithm is shown in Table 1.

Table 1. Algorithm for Sampling a scale-space representation

Algorithm 1: Sampling a scale-space representation
Input: (i) image to be processed, $f(x,y)$; (ii) the maximal pixel scale, N.
for $\sigma = 1:N$
(a) Calculate the radius of LoG template, $r = 4\sigma$;
(b) If $2r$ exceeds the size of the image, then break;
(c) Construct the normalized LoG template $T_\sigma(x,y)$ with radius r;
(d) Compute the convolution $\mathcal{D}L^2_\sigma = (f * T_\sigma)^2$;
end for
Build a 3-dimensional array $\mathcal{A}(:,:,:)$ by $\mathcal{A}(x,y,\sigma) = \mathcal{D}L^2_\sigma(x,y)$ $(\sigma = 1,2,\cdots,N)$;
Output: the array $\mathcal{A}(:,:,:)$, a discretized sample of $(\nabla^2_{norm}G * f(u,v))^2$.

In this algorithm, the responses of LoG on discretized scale-space representation are described by a 3-dimensional array, where the 1st and the 2nd dimensions represent x-axis and y-axis respectively for the image, and the 3rd dimension is the scale axis.

4. Extracting features from discretized scale-space representations

It is easy to find the maximal entry in the 3-dimensional array $\mathcal{A}$ in Table 1, and therefore through recording extrema and then excluding their neighbourhoods iteratively, a series of candidates of local features can be extracted. A crucial problem is how many candidates should be chosen as true local features. Since the scheme of global comparison is adopted in GPE, besides the threshold β mentioned in the previous section, a parameter λ can also be introduced to calculate a relative threshold to be applied to determinate a position in the series of candidates, before which all candidates are considered as local features. The parameter λ works with the maximal entry in the array $\mathcal{A}$ (denoted by M

Peer-reviewed version available at Information 2019, 10, 376; doi:10.3390/info10120376

here). When the response of a candidate times λ is less than M, the candidate is not a local feature. Otherwise, it is a local feature. Because of the scheme of local comparison, LPE employs an absolute threshold to determinate whether a candidate is a local feature. In contrast to the relative threshold in GPE, it lowers adaptiveness in the scene of illumination change. From (2), it is easy to know that for two images with the same content but different illumination, LPE and the GPE that only applies β as the threshold can compute different sets of extremal points under variant illumination, whereas using the relative threshold, GPE computes the same set under variant illumination, and therefore achieves adaptiveness to illumination change. Here we call the adaptiveness to illumination change as illumination invariance. Table 2 shows the algorithm for extracting features in GPE.

Table 2. Algorithm for extracting local features

Algorithm 2: Extracting extrema in discretized scale-space representations

Input: (i) the sample $\mathcal{A}(:,:,:)$ (an $n_1 \times n_2 \times n_3$ array) obtained by Algorithm 1; (ii) relative error threshold α; (iii) a positive real number λ.
Calculate the threshold β for the error tolerance by (7)
for $i = 1 : n_1 n_2 n_3$

(a) Find the maximum m in $\mathcal{A}(:,:,:)$;
(b) If m is the first maximum found, set $M = m$;
(c) If the product $\lambda \cdot m$ is smaller than M or $m < \beta^2$, then break;
(d) If the coordinate (x, y, σ) of m in the image has not been registered and $1 < \sigma < n_3$ then record the coordinate (x, y) in the image and the pixel scale σ in the scale space of this m as a vector $(x, y, \sigma)^T$;
(e) Register the coordinate (x, y);
(f) Annihilate all entries of the square submatrices of $\mathcal{A}(:,:,\sigma-1)$, $\mathcal{A}(:,:,\sigma)$ and $\mathcal{A}(:,:,\sigma+1)$ centered at (x, y) of order $6(\sigma-1)+1$, $6\sigma+1$ and $6(\sigma+1)+1$ respectively;

end for
Build a matrix $\boldsymbol{M}(:,:)$ by all records (column vectors) from step (d);
Output: the matrix $\boldsymbol{M}(:,:)$ consisting of extracted local features.

In this algorithm, local features in the image $f(x, y)$ are extracted to construct a matrix $\boldsymbol{M}$ whose columns are vectors $(x_i, y_i, \sigma_i)^T$, $i = 1, \cdots, N$ (for some positive integer N), which means that there are N local features located at (x_i, y_i) in the image with pixel scale σ_i.

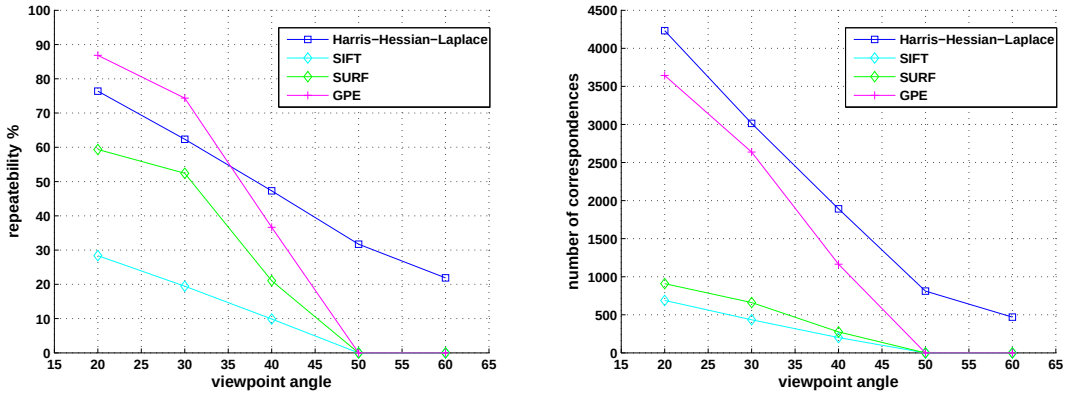
5. Simulations

Adjoining Algorithm 1 and Algorithm 2, we arrive at a complete algorithm for GPE, and we utilize repeatability to test the extracting performance of GPE. The score of repeatability is a ratio between the number of true matches and the number of matches. In general, an extractor attaining higher score of repeatability and larger number of true matches is a better extractor [13]. Test data, criteria and codes for the test of repeatability can be found at Mikolajczyk [13].[1] In all the following tests, set the parameter N in Table 1 to be 16, and the parameter α and λ in Table 2 to be 10^{-4} and 2000 respectively. We test the repeatability of GPE, Harris-Hessian-Laplace, SIFT and SURF on Mikolajczyk's test data. The executable file of Harris-Hessian-Laplace is from VGG [2]. The executable file of SIFT detector is from David Lowe [3]. The codes of SURF detector are OpenSURF, which are developed by Chris Evans [4]. The test results are shown from Figure 1 to Figure 8.

In the aspect of repeatability, GPE shows promising results. In comparison with SIFT and SURF, except the score being close to SIFT under the scene of JPEG compression (cf. Figure 7(a)), GPE acquires prominent advantage in all other cases. In contrast to Harris-Hessian-Laplace detector, except for some

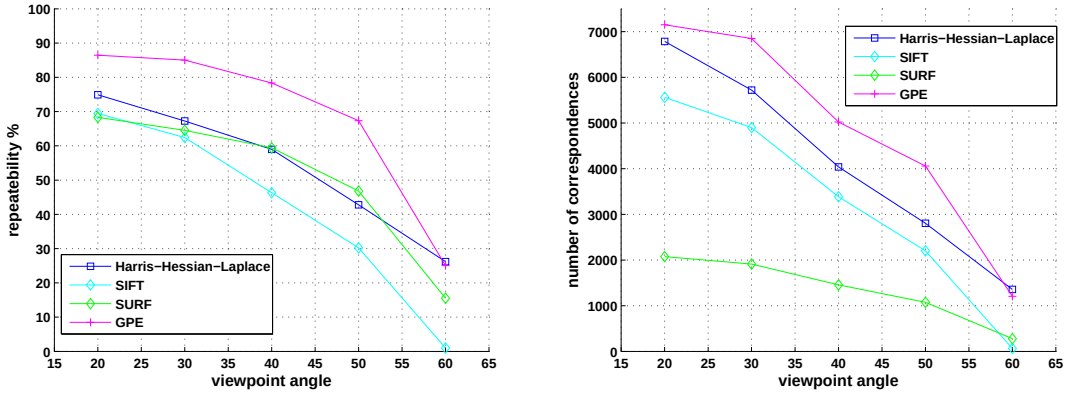
[1] The image sequences and the test software are from the website http://www.robots.ox.ac.uk/~vgg/research/affine/
[2] This executable file for windows is from the website http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html
[3] This executable file for windows is from the website http://www.cs.ubc.ca/lowe/keypoints/
[4] The codes of OpenSURF is from the website http://github.com/gussmith23/opensurf

Peer-reviewed version available at Information 2019, 10, 376; doi:10.3390/info10120376



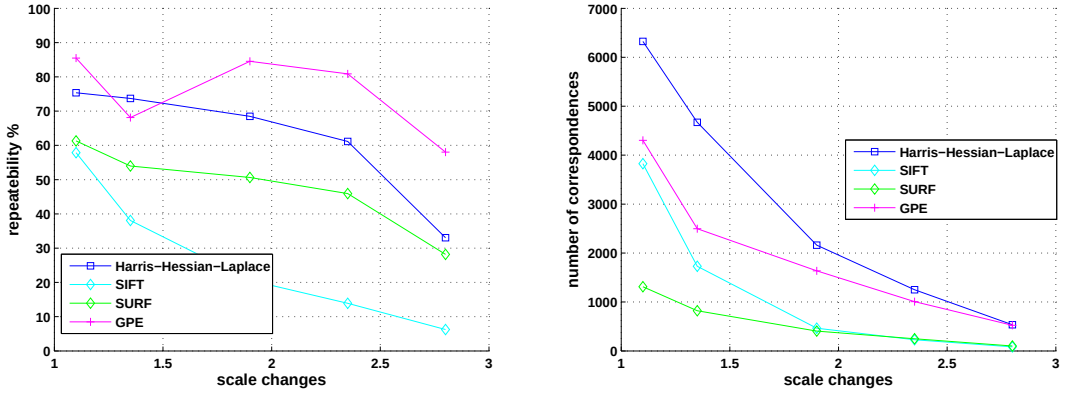
(a) (b)

Figure 1. Viewpoint change for the structured scene by the Graffiti sequence.



(a) (b)

Figure 2. Viewpoint change for the textured scene by the Wall sequence.

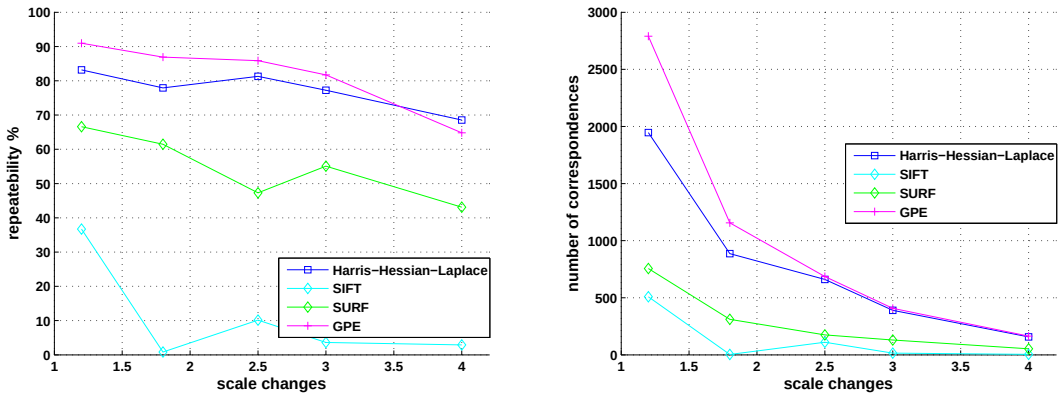


(a) (b)

Figure 3. Scale change for the structured scene by the Boat sequence.

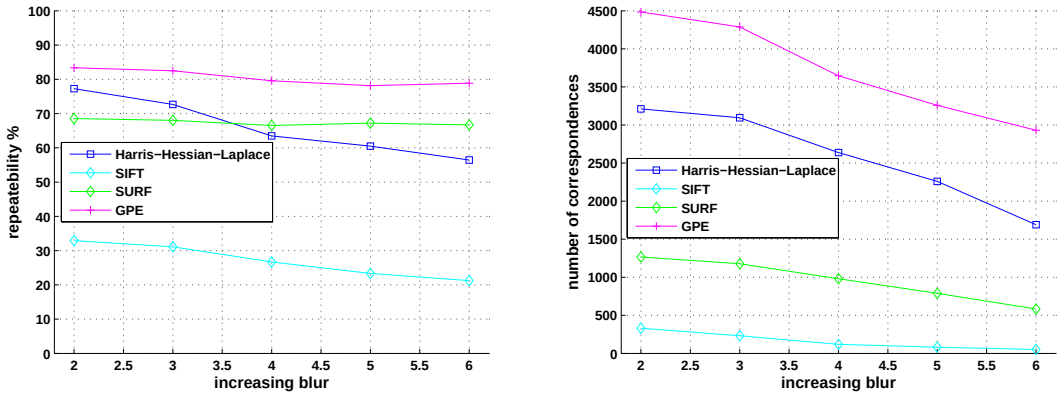
special situations, i.e., viewpoint change with more than 40 degrees for the structured scene in Figure 1(a), the viewpoint change with degrees greater than 60 for the textured scene in Figure 2, one slight change of scale change for the structured scene in Figure 3(a), the largest scale change for the textured scene in Figure 4(a), and the first four cases of JPEG compression in Figure 7(a), GPE obtains higher

Peer-reviewed version available at Information 2019, 10, 376; doi:10.3390/info10120376



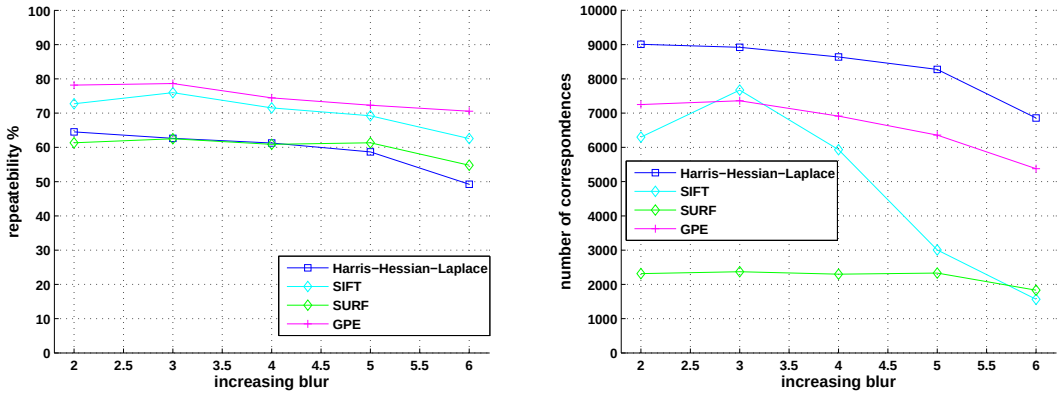
(a) (b)

Figure 4. Scale change for the textured scene by the Bark sequence.



(a) (b)

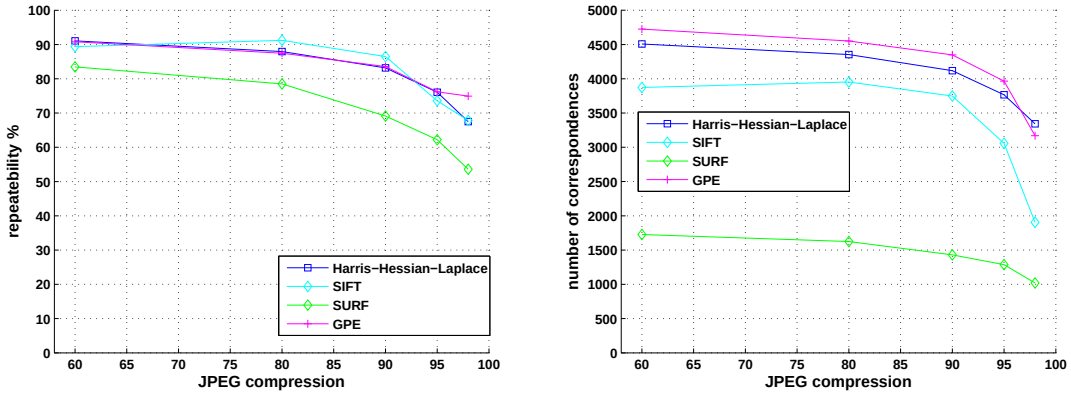
Figure 5. Blur for the structured scene by the Bikes sequence.



(a) (b)

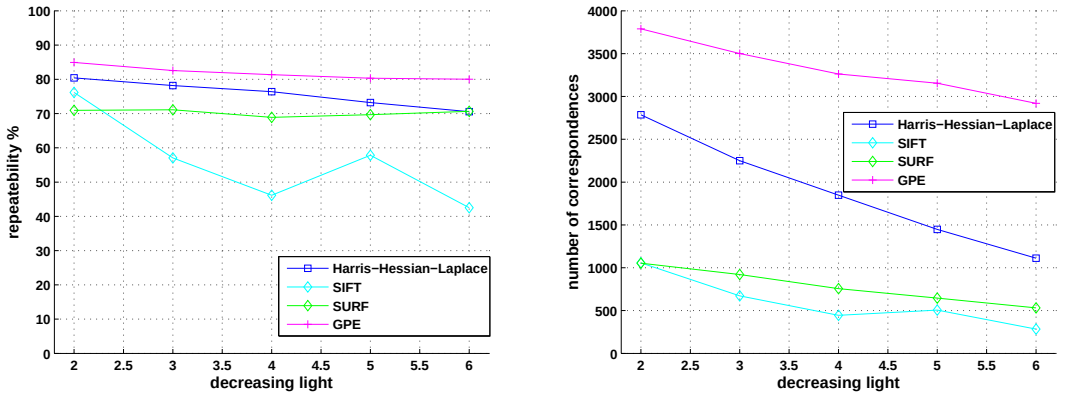
Figure 6. Blur for the textured scene by the Trees sequence.

scores. In the aspect of true recalls, GPE also shows fairly better performance under the situations of viewpoint change for the textured scene (cf. 2(b)), scale change for the textured scene (cf. 4(b)), blur for the structured scene (cf. 5(b)), JPEG compression (cf. 7(b)), and illumination change (cf. 8(b)). In



(a) (b)

Figure 7. JEPG compression by the UBC sequence.



(a) (b)

Figure 8. Illumination change by the Leuven sequence.

addition to the above comparisons, we use the results directly from related works to compare with GPE, and discuss them in following subsections.

5.1. Comparison with results of affine detectors from [13]

In the work [13], there are eight sets of test results for six affine region detectors, namely Harris-Affine [8,14,15], Hessian-Affine [8,14], MSER [16], IBR [15,17], EBR [17,18] and Salient [19].

Since GPE is not intended for the situation of viewpoint change, when the viewpoint angle is greater than 30 degrees, the repeatability score for structured scene is less than all those affine detectors. But from 20 to 30 degrees, GPE drastically overcomes any other detectors (cf. 1(a) and Figure 13(a) in [13]). In the situation of the images containing repeated texture motifs, from Fig. 2(a) (in comparison with Figure 14(a) in [13]) it can be seen that except viewpoint change of 60 degrees, GPE reaches higher repeatability score than all affine detectors, which means that as long as the viewpoint angle is less than 50 degrees, GPE has strong capacity of extracting affine features. In the tests for scale change and rotation, GPE shows its obvious advantages in both structured scene and texture scene except at scale 4 in the textured scene, where Hessian-Affine attains the repeatability of 70% (cf. Figure 4(a), 3(a) and Figure 15(a), Figure 16(a) in [13]). In the results shown in Figure 5 (in contrast to Figure 17(a) in [13]) and Figure 6 (in contrast to Figure 18(a) in [13]), GPE shows excellent capacity to cope with the situation of blur in both structured scene and texture scene. In those comparisons, none of other detectors achieves higher repeatability score than GPE at any test point. In the test of JPEG compression, GPE

Peer-reviewed version available at Information 2019, 10, 376; doi:10.3390/info10120376

has similar performance compared with Harris-Affine and Hessian-Affine, but obviously outperforms other LPE detectors (cf. Figure 7(a) and Figure 19(a) in [13]). The Hessian-Affine detector shows its slight advantage for JPEG compression change. Figure 8(a) and Figure 20(a) in [13] shows that GPE has good robustness to illumination change and overall higher repeatability score than other extractors.

5.2. Comparison with results from [10]

In the work [10], five detectors, FH-15, FH-9, DoG [6], Harris-Laplace and Hessian-Laplace [14] have been tested for repeatability performance of viewpoint change for structured scene, viewpoint change for textured scene, scale change for structured scene, and blur for structured scene. Therefore there are four results can be adapted directly, which are shown in the Fig. 16 and Fig. 17 in [10]. Comparing these results respectively with the (a) in Fig. 1, 2, 3 and 5, it can be seen that except one point (which is the scale change about 1.3 for FH-15 and DoG), GPE apparently overcomes FH-15, FH-9, DoG, Harris-Laplace and Hessian-Laplace in all these tests.

5.3. Comparison with results from [11,12]

Figure 5 in [11] and Figure 7 in [12] show results in the tests of LOCKY, where the sub-figures, (a), (b), (c), (d), (e), (f), (g), (h) correspond to Figure 4(a), 5(a), 3(a), 1(a), 8(a), 6(a), 7, and 2(a) respectively in our work. Since LOCKY mainly aims to achieve faster computation than most of the currently used feature detectors, except the cases that viewpoints are greater than 40 in the Graffiti sequence, GPE shows apparently higher repeatability score than LOCKY.

6. Conclusion

We present a new method (GPE) for local feature extracting with high repeatability, which transforms a discretized scale-space presentation through LoG and extracts local features by the scheme of global comparison. Because of the use of convolution templates of disk shape, GPE is rotation-invariant. Discussion for the radii of convolution templates and the error caused by finite radii is an important merit in our work. We first decompose the LoG transformation of a discretized scale-space presentation into two parts, the approximation and the error. Then an upper bound of the error under a given radius is worked out and we utilize this upper bound to determinate a threshold, below which the candidates are no longer regarded as features since the computational error can influence the precision of the approximation (cf. (6) and (7)). Because of the global comparison, the relative threshold can be employed to choose local features from candidates, and hence these chosen features are illumination-invariant. Since the kernel width increases only one pixel a time, GPE obtains more precise scales for extracted local features without interpolation than LPE does, and therefore the step of interpolation for precisely locating the scale of a feature point in LPE is elided in GPE. Simulations show that GPE reaches high performance for repeatability and true recalls in various situations, including scale change, rotation, blur, JPEG compression, illumination change and even viewpoint change of a textured scene.

Author Contributions: Methodology, original draft and writing, Q.Z.; Supervision, B.S.

Funding: This research is supported by Guangdong Project of Science and Technology Development (2014B09091042) and Guangzhou Sci & Tech Innovation Committee (201707010068).

Acknowledgments: The authors appreciate Krystian Mikolajczyk for his test data set, criteria and executable files. The authors are also thankful to David Lowe for his executable file of SIFT, and to Chris Evan for his code of OpenSURF.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. C Harris and M. Stephens. A combined corner and edge detector. *Proc Alvey Vision Conf*, 1988(3):147–151, 1988.

2. Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 1, pages 612–618 vol.1, 2000.
3. Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, Nov 1998.
4. Tony Lindeberg. Scale-space theory: a basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1-2):225–270, 1994.
5. D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.
6. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
7. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 525–531 vol.1, July 2001.
8. Krystian Mikolajczyk and Cordelia Schmid. *An Affine Invariant Interest Point Detector*, pages 128–142. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
9. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. *SURF: Speeded Up Robust Features*, pages 404–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
10. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.
11. J. Lomeli-R and Mark S. Nixon. The brightness clustering transform and locally contrasting keypoints. In George Azzopardi and Nicolai Petkov, editors, *Computer Analysis of Images and Patterns*, pages 362–373, Cham, 2015. Springer International Publishing.
12. Jaime Lomeli-R. and Mark S. Nixon. An extension to the brightness clustering transform and locally contrasting keypoints. *Machine Vision and Applications*, 27(8):1187–1196, Nov 2016.
13. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, Nov 2005.
14. Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, Oct 2004.
15. F. Schaffalitzky and A. Zisserman. *Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?"*, pages 414–431. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
16. Jiri Matas, Ondrej Chum, Martin Urban, and Tomas Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, volume 22(10), pages 384–393, Cardiff, UK, 01 2002.
17. Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *In Proc. BMVC*, pages 412–425, 2000.
18. Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, Aug 2004.
19. Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *the 8th European Conference on Computer Vision*, pages 228–241, Prague, Czech Republic, 05 2004.