Sensors & Transducers, Vol. 165, Issue 2, February 2014, pp. 137-141

www.sensorsportal.com

Sensors & Transducers

http://www.sensorsportal.com

A Novel Real-Time Feature Matching Scheme

Ying Liu, * Hongbo Bi, Mengmeng Wu, Wufeng Yue, Panpan Zhao
North East Petroleum University, Daqing, 163318, China
* Tel.: 086-0459-6503272, fax: 086-0459-6503160
* E-mail: bhbdq@126.com

Received: 14 November 2013 /Accepted: 28 January 2014 /Published: 28 February 2014

Abstract: Affine Scale Invariant Feature Transform (ASIFT) can obtain fully affine invariance, however, its time cost reaches about twice that in Scale Invariant Feature Transform (SIFT). We propose an improved ASIFT algorithm based on feature points in scale space for real-time application. In order to detect the affine invariant feature point, we establish a second-order difference of Gaussian (DOG) pyramid and replace the extreme detection in the DOG pyramid by zero detection in the proposed second-order DOG pyramid, which decreases the complexity of the scheme. Experimental results show that the proposed method has a big progress in the real-time performance compared to the traditional one, while preserving the fully affine invariance and precision. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Scale space, Affine invariance, Scale invariant feature transform, Difference of Gaussian.

1. Introduction

Feature extraction and description play a very important role in image and video analysis and interpretation. Intensive research results show that local features are more robust than the global ones. Local image detectors used for image processing can be typically classified by their incremental invariance properties. Local image features have been widely applied in image retrieval, image matching, image register, motion recognition, video monitoring, etc [1-2]. Almost all of the existed local image detectors are translation invariant. Furthermore, the Harris detector is also rotation invariant [3]. Harris-Laplace, Hessian-Laplace and Difference of Gaussian (DoG) are scale and rotation invariant [4]. The Hessian-Affine, Harris-Affine and Maximally Stable Extremal Region (MSER) are expected to be affine transform invariant [5-6]. The invariance is commonly achieved by normalizing the local regions, patches or level lines and so on, where the effect of the affine transform has been eliminated. However, none of these approaches are yet fully affine invariant when a strong change of scale or angle is present. Affine scale invariant feature transform (ASIFT) provided a potential solution [7]. It simulates all possible image views and thus achieves fully affine transform invariant. ASIFT can handle much higher transition tilts than SIFT, Harris-Affine, Hessian-Affine and MSER. But its complexity is about twice that of SIFT. So on account of it, ASIFT is not suitable for the real-time application. Inspired by ASIFT, the proposed method in this paper simulates possible affine distortions caused by the change of camera optical axis orientation from a frontal position. And SIFT algorithm is modified to speed up the process of feature extraction and further to meet the real-time requirements.

Specifically, the contributions of this paper are as follows.

1) The proposed method makes full use of the affine invariant advantage of ASIFT and the efficient merit of SIFT.

2) We establish the second-order DoG pyramid to simplify the detection of initial extreme feature points, which speeds up the feature extraction and description.

The rest of the paper is organized as follows. Section 2 reviews the ASIFT algorithm. In Section 3, we briefly describe the proposed real-time ASIFT in detail. Section 4 presents our experimental results comparing the novel method to the traditional ASIFT. Finally, the paper is concluded in Section 5.

2. Review of the ASIFT Algorithm

We first briefly review the fully affine invariant image matching method, ASIFT [6]. Unlike SIFT, which is invariant with respect to only four parameters namely zoom, rotation and translation, ASIFT deals with another two parameters: the angles defining the camera axis orientation. Against any prognosis, simulating all views depending on these two parameters is feasible. The method permits to reliably identify features that have undergone very large affine distortions measured by the transition tilt.

Image distortions arising from viewpoint changes can be modeled by affine transform, provided the image's boundaries are piecewise smooth [8]. As shown in Fig. 1, the gray image area is a flat physical object. The top right small rectangle represents a camera.

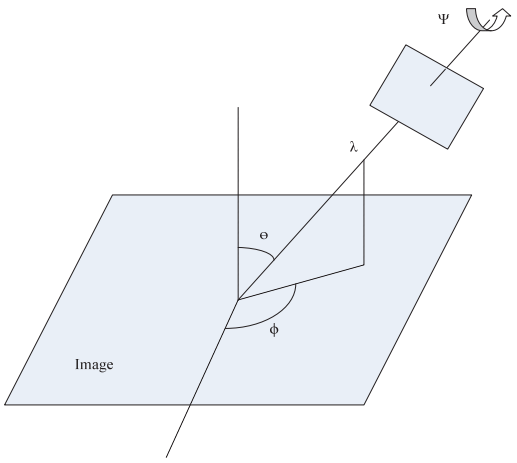


Fig. 1. The Affine Camera Model

Suppose an image $I(x,y)$ is affine transformed to $I'(x',y')$:

$$\begin{aligned} I'(x',y') &= I(x,y) * A + E \\ &= \begin{bmatrix} x \\ y \end{bmatrix} * \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \\ &= I(ax+by+e, cx+dy+f) \end{aligned} \tag{1}$$

So we obtain the affine decomposition matrix:

$$A = \lambda \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{2}$$

The ϕ and θ are the longitude and latitude angles of the camera optical axis, respectively. The ψ angle is the camera spin, and λ represents the zoom parameter.

In ASIFT [8], the longitude angle ϕ is changed with the latitude angle θ, with step $\Delta\phi = 72^\circ / t$, $t = 1/\cos\theta$, $\phi \in [0, \pi)$, and $\theta \in [-\pi/2, \pi/2]$.

Generally speaking, ASIFT proceeds as follows:

1) Each image is transformed by simulating all possible linear distortions caused by the change of orientation of the camera axis. These distortions depend upon two parameters: the longitude ϕ and the latitude θ.

2) The rotations and tilts are performed for a finite and small number of latitude and longitude angles. Then the images undergo rotations with angle followed by tilts.

3) All simulated images are compared to each other by some scale invariant, rotation invariant, and translation invariant algorithm. And all simulated images are computed by the scheme, while original SIFT is used in ASIFT for feature extraction.

After all, the main contributions of ASIFT are:

1) It simulates all six parameters in affine model.

2) It makes full use of the merits of SIFT.

3. Speed Up ASIFT

SIFT is invariant to image scaling, blur, and illumination, and partially invariant to rotation and viewpoint changes. However, SIFT is not affine invariant. ASIFT can obtain fully affine invariance, while its time computation cost is rather huge, about twice that of SIFT because of the use of SIFT.

In this paper, we propose an effective method to overcome the above drawback of ASIFT. Unlike MSER, Harris-affine, and Hessian-affine, which normalize all six affine parameters, The improved ASIFT simulates the two camera axis parameters, which is same as ASIFT, and then applies the proposed fast version of SIFT to extract features from the generated images. In order to further speed up feature extraction, the number of latitudes and longitudes are also carefully selected.

By analyzing the process of ASIFT, it can be divided into two parts mainly, namely, the simulation of the view angle and SIFT, where SIFT occupies most of the computation. To meet the real-time application, a straightforward solution is to speed up SIFT. SIFT proceeds by the following steps: the detection of the extreme in scale space, the extraction

of stable key points, assignment of direction for each key point, and generation of feature descriptor. To speed up the performance of SIFT, many improved versions have been reported, such as PCA-SIFT, GLOH and SURF [9-11]. However, most of them are concentrated on the simplification of descriptors, which can enhance the calculation speed, while decreasing the robustness and accuracy.

We can draw conclusion by analyzing the above steps: the detection of scale invariant feature points is the basis of the whole SIFT scheme, which influences the accuracy greatly. Furthermore, the establishment of LOG pyramid and DOG pyramid occupies about 80 % of the computation [12]. As a result, simplifying the establishment of the pyramid can decrease the time cost efficiently. We propose to exploit the second-order DOG to detect the initial extreme feature points.

3.1. Theoretical Basis

In SIFT, the detection of extreme is obtained by DoG, where each pixel will compare with 26 pixels in the neighboring layers and octaves, which needs much calculation time. On the other hand, inspired by plane geometry, the local extreme of the primitive function corresponds to the zero point of the first derivative of the function. Furthermore, the zero point is more apparent than the extreme and easy to detect. So we can replace the detection of local extreme by the detection of zero point of the first derivative of the scale normalized primitive function. As a result, the local extreme in DoG corresponds the zero point of the derivative of DoG, that is, the second-order DoG.

Suppose $D(\cdot)$ denotes the DoG, so

$$D(x,y,k\sigma)-D(x,y,\sigma)=D^2(x,y,\sigma), \quad (3)$$

where $D^2(x,y,\sigma)$ represents second-order DoG. Since

$$\frac{\partial D}{\partial \sigma}\approx\frac{D(x,y,k\sigma)-D(x,y,\sigma)}{k\sigma-\sigma}=\frac{D^2(x,y,\sigma)}{k\sigma-\sigma} \quad (4)$$

$$D^2(x,y,\sigma)=D(x,y,k\sigma)-D(x,y,\sigma)\approx\frac{\partial D}{\partial \sigma}(k\sigma-\sigma) \quad (5)$$

And $k\sigma-\sigma\neq 0$, so

$$D^2((x,y,\sigma)=0\Leftrightarrow\frac{\partial D}{\partial \sigma}=0 \quad (6)$$

3.2. Detection of Second-order DoG

3.2.1. Establishment of Second-order DoG

The second-order DoG is built based on DoG, as illustrated in Fig. 2.

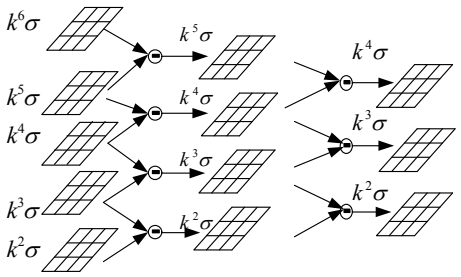


Fig. 2. Establishment of the second-order DoG (Left is Gaussian pyramid, middle is DoG, right is the second-order DoG).

Take the subtraction between the neighboring 2 layers in the same order in DoG, and the result is regarded as one layer in the second-order DoG, whose scale is the scale of the subtrahend in DoG. For example, the scale image in the first layer second-order in second-order DoG results from the subtraction the first layer $D(x,y,k^2\sigma)$ from the second layer $D(x,y,k^3\sigma)$ in DoG, and its scale is $k^2\sigma$. As a result, the number of order in second-order DoG is the same as that in DoG, while the number of layer in each order in second-order DoG is less 1 than that in DoG.

3.2.2. Detection of Zero Point

In the real application, the zero point in the second-order DoG does not equal to zero accurately, so we set a predefined threshold *T*, and compare the threshold with the absolute value of the pixel in the second-order DoG to detect the zero point. If the result is less than *T*, the pixel is considered as one feature point, and its location and scale (x_i,y_i,σ_i) is recorded. In the proposed scheme, the selection of *T* is rather important. When *T* is selected to be bigger, the number of the feature points becomes bigger, which may detect partial false feature points, and leads to increase the false matching probability, and vice versa. We have carried out a large number of experiments to certain the relationship between *T* and the performance. Experimental results show that with the increase of *T*, the number of the feature points increases, and the number of the correct matching pairs increases. Furthermore, the correct matching probability increases first and then decreases, and the time consumption will increase with the increase of *T*. So we suggest the range $T\in[0.0000012,0.0000035]$ be a tradeoff between the precision and the computation.

3.2.3. Discussion about Location and Description of Feature Points

For the feature points detected in the second-order DoG, it is necessary to locate their and position in the original image. We need to map the detected feature points in the second-order DoG to the DoG space, and the principle is the same order, the same scale and the same position, that is, feature point (x_m, y_m, σ_m) in the second DoG corresponds to pixel (x_m, y_m, σ_m) in DoG. So we can adopt the same way as in SIFT to locate the position of each feature point precisely. Furthermore, the rest after detection of extreme is also the same as in SIFT, including the removal of low contrast and unstable edge points, generation of feature descriptor, etc.

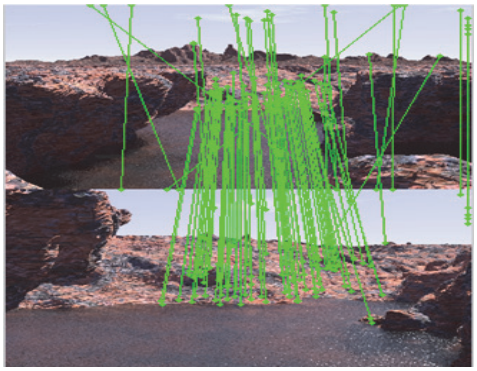
4. Experimental Results

For assessing the performance of the proposed algorithm, a variety of experiments were carried out. The experimental environment are: AMD Athlon ™ X2 DualCore CPU 2.0 GHz, memory 2.0 GHz, Matlab2011b. We analyze the results from the angle of total time, corresponding number and correct rate. Partial simulation results are depicted in Fig. 3, Fig. 4 and Fig. 5.

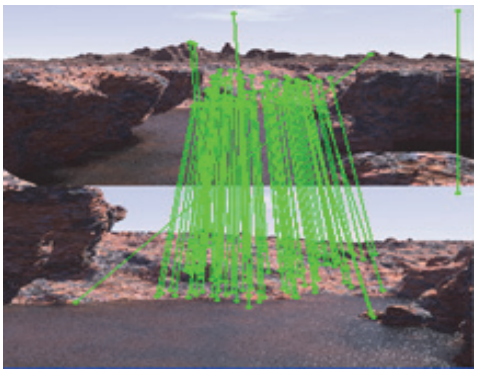
(a) Feature Points by ASIFT.

(b) Feature Points by Improved ASIFT.

Fig. 3. Comparison between ASIFT and Improved ASIFT (Lena).

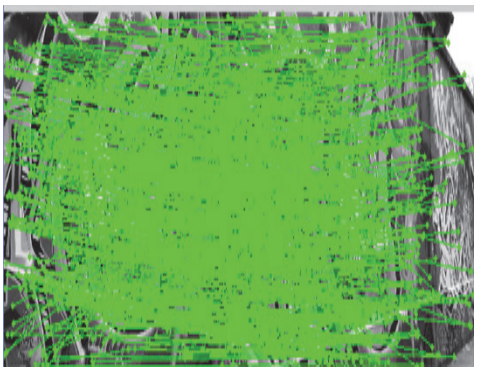


(a) Matching by ASIFT.

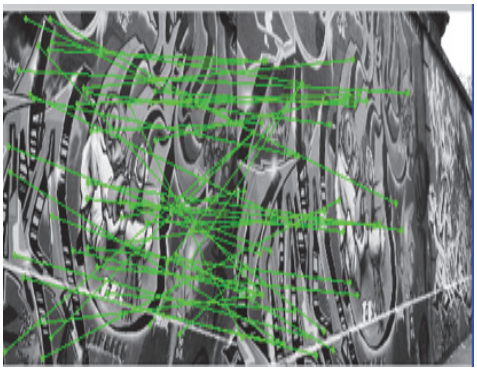


(b) Matching by Improved ASIFT.

Fig. 4. Comparison between ASIFT and Improved ASIFT (landscape).



(a) Matching by ASIFT.



(b) Matching by Improved ASIFT.

Fig. 5. Comparison between ASIFT and Improved ASIFT (img).

Other results are listed in Table 1. Where parameters of SIFT scheme are same as recommended by Lowe, and the parameter $T = 0.0000012$ in improved ASIFT.

Table 1. Comparison between ASIFT and proposed ASIFT.

	Total Time (s)	Corresponding Number	Correct Rate (%)
ASIFT	79.756	103	89.56
Proposed ASIFT	39.949	34	89.07

From Table 1, we can see that regardless of under geometric distortion condition, under the illumination variation condition, the proposed scheme is better than the traditional one. The total time is about 1/2 that of ASIFT, indicating the better real-time performance. Furthermore, the correct rate is equivalent to that in ASIFT, indicating good robustness.

5. Conclusions

In this paper, a novel speed up affine SIFT scheme is presented. ASIFT is fully affine invariant with respect to 6 parameters, including zoom, rotation, translation and angles, etc, which favors the image matching with large transition tilts. However, the computation of ASIFT is about twice that of SIFT, which is not suitable for real-time application. Analyzing the realization steps, we propose to simplify the detection of extreme in SIFT, and build the second-order DoG, exploiting detection of zero point to replace the comparison between pixels. Experimental results show the proposed scheme decreases the running time while preserving the precision. On the other hand, the corresponding number of the proposed scheme is rather lower than that of the traditional one, which may lose some distinctive points, especially on the edge or highly texture parts, leading to lower the precision of matching. Further work of increasing the corresponding number to enhance the precision in our approach is in progress.

Acknowledgements

This work is supported by the Science & Technology Project of Heilongjiang province under Grant No. 12521056.

References

[1]. Q. Hu, X. Chen, L. Zhang, *et al.*, Feature evaluation and selection based on neighborhood soft margin, *Neurocomputing*, 73, 10–12, 2010, pp. 2114-2124.

[2]. Yanwei Pang, Wei Li, Yuan Yuan, *et al.*, Fully affine invariant SURF for image matching, *Neurocomputing*, 85, 2012, pp. 6-10.

[3]. C. Harris, M. Stephens, A combined corner and edge detector, in *Proceedings of the Alvey Vision Conference*, Manchester, 1988, pp. 147-152.

[4]. David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision,* 60, 2, 2004, pp. 91-110.

[56]. K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in *Proceedings of the Seventh European Conference on Computer Vision*, Copenhagen, Denmark, 2002, pp. 128-142.

[6]. J. Matas, O. Chum, M. Urban, *et al*., Robust wide-baseline stereo from maximally stable extremal regions, *Image Vision Comput.,* 22, 10, 2004, pp. 761-767.

[7]. J. M. Morel, G. Yu, ASIFT: a new framework for fully affine invariant image comparison, *SIAM J. Imaging Sci.*, 2, 2, 2009, pp. 438–469.

[8]. K. Mikolajczyk, T. Tuytelaars, C. Schmid, et al., A Comparison of Affine Region Detectors, *IJCV*, 65, 1, 2005, pp. 43-72.

[9]. Rahul Sukthankar, Yan Ke, PCA-SIFT: A More Distinctive Representation for Local Image Descriptors, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, 27 June-2 July 2004, pp. II-506 - II-513.

[10]. K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, *IEEE Trans. Pattern Anal. Mach. Intell.*, 27, 10, 2005, pp. 1615-1630.

[11]. H. Bay, A. Ess, T. Tuytelaars, *et al.*, SURF: speeded-up robust features, *Comp. Vision Image Understanding*, 110, 3, 2008, pp. 346-359.

[12]. Cao Juan, Li Xing-Wei, Lin Wei-Ting, *et al.*, Study on Improved SIFT Feature Matching Algorithm, *Journal of System Simulation*, 22, 11, 2010, pp. 2760.