SCALE-INVARIANT CORNER KEYPOINTS

Bo Li

Digital Media Lab
Dept. of Applied Physics and Electronics
Umeå University, Sweden

Haibo Li

School of Computer Science and Communication
KTH Royal Institute of Technology, Sweden

Ulrik Söderström

Digital Media Lab
Dept. of Applied Physics and Electronics
Umeå University, Sweden

ABSTRACT

Effective and efficient generation of keypoints from images is the first step of many computer vision applications, such as object matching. The last decade presented us with an arms race toward faster and more robust keypoint detection, feature description and matching. This resulted in several new algorithms, for example Scale Invariant Features Transform (SIFT), Speed-up Robust Feature (SURF), Oriented FAST and Rotated BRIEF (ORB) and Binary Robust Invariant Scalable Keypoints (BRISK). The keypoint detection has been improved using various techniques in most of these algorithms. However, in the search for faster computing, the accuracy of the algorithms is decreasing. In this paper, we present SICK (Scale-Invariant Corner Keypoints), which is a novel method for fast keypoint detection. Our experiment results show that SICK is faster to compute and more robust than recent state-of-the-art methods.

Index Terms— Keypoint detection, image matching, edge detection, corner detection, scale-space

1. INTRODUCTION

Keypoint detection and feature description are at the core of many computer vision applications, such as object matching, 3D reconstruction, image retrieval, and camera localization. These applications may have to handle a large amount of data or be able to run on mobile devices with limited computational power. Therefore, there is a growing need for fast keypoint detection, fast feature description, and fast matching. The last decade featured an arms race toward faster feature description and matching algorithms. Some of them are Scale Invariant Features Transform (SIFT) [1], Speed-up Robust Feature (SURF) [2], and more recently Oriented FAST and Rotated BRIEF (ORB) [3] and Binary Robust Invariant Scalable Keypoints (BRISK) [4]. The actual keypoint detection has been improved using various techniques for the different algorithms.

An ideal keypoint detector finds salient image locations that can be detected repeatedly despite all possible image transformations, such as a change of zoom or rotation. SIFT is the most well-known method from the early work of keypoint detection and description. In order to achieve a faster performance, SURF aimed at increasing the speed of keypoint detection and later research have aimed for extreme performance in speed, with one result being the FAST corner [5] detector. However, FAST corner detector is not scale invariant, a potential problem which was addressed with pyramid schemes in ORB and BRISK.

This paper proposes a novel method for detecting keypoints from images. Our method strictly follows the idea of generating local extremes. Non-maximum suppression is calculated both in 2-dimensional image space and in scale-space. First an edge chain is extracted by applying non-maximum suppression in both the 2D image space and scale space [26]. Secondly, edge based corner detection is implemented by detecting edge direction change.

Our algorithm has two advantages. Firstly, an edge based corner measure is more efficient than a corner measure using a Harris or Hessian matrix. Secondly, feature descriptors usually need an orientation assignment step to generate keypoint orientation. Since we use the gradient direction as keypoint orientation, the orientation can be assigned directly without additional cost.

2. RELATED WORK

The early adopted interest point detector is the Harris corner detector [6], proposed in 1988, which uses eigenvalues of the second-moment matrix for cornerness measure. However, the Harris corner detector is not scale-invariant. Lindeberg developed a scale-invariant feature detection method which uses Hessian matrix measure to detect blob-like features [7]. Mikolajczyk and Schmid refined this method and named their solutions Harris-Laplace and Hessian-Laplace [8]. Hessian matrix is still used in both SIFT and SURF to eliminate edge responses. In order to get scale-space extremes [26], SIFT calculates the extreme of Difference of Gaussian (DoG) and SURF calculates the extreme of Hessian measure. Extremes of DoG have large edge responses and therefore a Hessian matrix is employed to measure cornerness. The use of Hessian matrix is time-consuming due to the convolution of Gaussian second order derivatives with the image in scale-space. By using integral images, the Hessian matrix can be calculated rapidly, a solution which is used in SURF. Thus SURF detects local extremes directly based on Hessian measures. Michael et al. improved SIFT in the sense of speed [9] by implementing DoG from approximate Gaussians. They achieved this by using box filters which can be calculated rapidly from integral images. Their experiments show that with a slight decrease in matching and repeatability performance, their Fast approximated SIFT can speed up the procedure by a factor of eight. Later research was eager to achieve extreme speed performances, as a result the FAST corner detector was adopted. FAST takes the intensity threshold between the center pixel and those in a circular ring around the center as the only parameter, making it very fast. FAST, however, is not scale invariant, an issue which was addressed with the pyramid schemes developed in ORB. FAST does not produce an accurate measure of cornerness so the Harris corner measure needs to be used. ORB employed Harris corner measure to arrange the FAST keypoints in order. For a target number of N keypoints, ORB first sets the threshold low enough to get more than N keypoints, orders them according to the Harris measure, and then picks the top N keypoints. ORB's implementation of multi-scale keypoint detection is the fastest today. It avoids using non-maximum suppression (NMS) which is time consuming to some extent. BRISK also uses FAST but takes it further by using NMS in 2D image space and in scale-space. It directly uses the FAST score instead of using further Harris measure check. BRISK's keypoint detector is a little slower than ORB's since it uses NMS.

Is there an alternative way to measure cornerness both regarding accuracy and speed? An early survey in corner detection has showed many examples [11]. Edge-related corner detection considers the corner point as the junction of two or more edge lines. Therefore, corners can be considered as additional features on edge points. Intuitively, one approach would be to produce curve representations from edges. Several such implementations have been developed. For example, Cooper et al. estimated the curvature using pixel coordinates of edge chains [12], Farzin and Riku defined corners as points where image edges have their maxima of absolute curvature [15].

We propose to measure the change of the gradient direction along an edge contour. This can be calculated rapidly through a one-dimensional integral image (also known as summed area table) [16, 17]. First, edges are generated by finding local extremes of gradients both in image space and scale-space. Secondly, the corner score is measured by a simple geometric vector subtraction which can be calculated rapidly. The vector subtraction takes the difference between summed vectors of 'left' edges and 'right' edges of a target location (See Fig. 2 for an explanation of 'left' and 'right' edges). The summing window size depends on the scale parameter of the edge in the target location. Therefore, the detected corners are scale invariant, namely Scale-Invariant Corner Keypoints (SICK).

Many keypoint detectors need a separate orientation operator. SIFT and SURF use histograms of gradients to find its main orientation and these methods are computationally demanding. In ORB, an intensity centroid is used to measure the corner orientation. This idea is very similar to the idea we adopted in our edge detection step using center of mass (CoM) [18]. By using this technique, corner orientation can be calculated directly from the gradient direction.

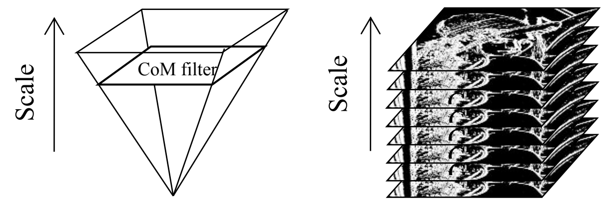

Fig. 1. The gradient on each scale (right) is generated by gradually increasing the CoM filter size (left).

3. SICK: THE METHOD

3.1. Edge in scale-space

Conventional edge detection methods use various gradient filters (i.e. kernels) to generate gradients (e.g. Laplace) or directional gradients (e.g. Sobel) [19, 20]. Inspired by previous work on multi-scale edge detection [21, 22], we generate edges on a continuous scale-space by finding local extremes of gradients both in image and scale-space.

1) Fast gradient detection: Multi-scale edge detection needs to calculate gradients by gradually increasing the filter size when the scale increases. Time consumption for a conventional gradient filter increases quadratically when the scale increases. With the fast CoM edge detector [18], gradients can be calculated with a constant time consumption even when the scale increases. We use the CoM filter to calculate gradients in a continuous scale-space (e.g. filter size with 3×3, 5×5 7×7, 9×9, 11×11, 13×13, 15×15, 17×17, totally 8 layers) (Fig. 1).

2) Non-maximum suppression in image and scale-space: First we need to find all potential edges by using Canny's non-maximum suppression method [19], meaning that we compare gradients at target locations with their two neighbors along the gradient direction. Secondly, to generate a scale-space maximum, the gradient needs to be compared with its two neighbors from a higher and a lower scale. To prevent noise, we actually compare the gradient with neighborhoods, both along the gradient direction and in scale-space. Before the use of NMS, gradient images in each scale are multiplied with a noise reduction parameter C_S ($C_S<1$). This idea is borrowed from work in [22]. C_S decreases slightly when the scale increases.

3) Image pyramid: To boost the performance of our method, edge detection is applied in an image pyramid, which is similar to the framework used in SIFT.

3.2. Corner in scale-space

1) Preprocessing on edge chain: We need to organize and save the detected edges in the correct order in a one-dimensional sequence. This can easily be implemented with edge linking. However, detected edges still have a problem with redundancy. Double edge effect appears on the edges around the directions of *-135°*, *-45°*, *45°*, and *135°*. This redundancy can be deleted by applying a small condition called Restricted Hysteresis [23].

2) Fast gradient direction change measure: The vector subtraction between 'left' and 'right' gradient vectors of a target location can simply reflect the magnitude of the gradient direction change (Fig. 2). The norm of V_d is then used in the corner measure (or corner score):

$$\|V_d\| = \|Vr - V_l\| \tag{1}$$

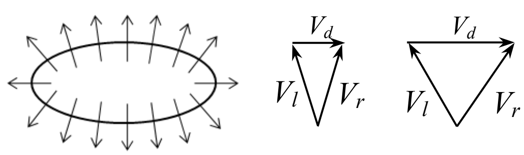

Fig. 2. The gradient direction is illustrated on the schematic edges with an ellipse shape (left). Substration of two gradient vectors can simply reflect gradient direction change (middle and right). V_l and V_r denote 'left' and 'right' gradient vectors of a target location.

The magnitude of V_d is also affected by magnitude of V_l and V_r. To normalize this effect, the corner score is defined as:

$$S_c = \frac{\|V_d\|}{\|V_l\|+\|V_r\|} \tag{2}$$

Simply using a single gradient vector on the 'left' and a single vector on 'right' side will affect the accuracy of the corner score because of noise. This noise is caused by the discrete property of digital images. Therefore, we need to consider more gradient vectors on the 'left' and 'right' side in a range w. The new corner score is calculated as:

$$S_c' = \frac{\|\sum_{t-w/2}^{t} V - \sum_{t}^{t+w/2} V\|}{\|\sum_{t-w/2}^{t} V\| + \|\sum_{t}^{t+w/2} V\|} \tag{3}$$

where t is target location on the edge. The sum function is a one dimensional summation on the edge chain. Furthermore, since we are considering scale presentation, the range w should correctly reflect the corner scale. The edges detected are both image space maxima and scale-space maxima. Therefore, we can use the detected edge scale on target location t as an indicator of the range of the sum function:

$$w = a\, s_t \tag{4}$$

We chose $a = 3$ in our experiment. The geometrical vector summation or subtraction is simply a summation or subtraction of its components. Therefore, the equation (3) can be represented as:

$$S_c' = \frac{\sqrt{\left(\sum_{t-w/2}^{t} G_x - \sum_{t}^{t+w/2} G_x\right)^2 + \left(\sum_{t-w/2}^{t} G_y - \sum_{t}^{t+w/2} G_y\right)^2}}{\sqrt{\left(\sum_{t-w/2}^{t} G_x\right)^2 + \left(\sum_{t-w/2}^{t} G_y\right)^2} + \sqrt{\left(\sum_{t}^{t+w/2} G_x\right)^2 + \left(\sum_{t}^{t+w/2} G_y\right)^2}} \tag{5}$$

where G_x, G_y are directional gradients. To simplify the equation, we get the new corner score as:

$$S_c'' = \frac{\left(\sum_{t-w/2}^{t} G_x - \sum_{t}^{t+w/2} G_x\right)^2 + \left(\sum_{t-w/2}^{t} G_y - \sum_{t}^{t+w/2} G_y\right)^2}{\left(\sum_{t-w/2}^{t} G_x\right)^2 + \left(\sum_{t-w/2}^{t} G_y\right)^2 + \left(\sum_{t}^{t+w/2} G_x\right)^2 + \left(\sum_{t}^{t+w/2} G_y\right)^2} \tag{6}$$

By borrowing the idea of integral images [16], the one-dimensional summation along an edge chain can be calculated even faster. The summation within a range can be calculated with a constant number of operations regardless of the range.

3) Non-maximum suppression: The last step of corner detection is simply to detect one-dimensional local maxima of the corner score along edge chains using NMS. Endpoints on edge chains are automatically considered as corners.

4) Eliminating weak keypoints: Eliminating weak keypoints is a thresholding procedure on the keypoint strength function. The strength of keypoints (S_k) is measured both by corner score and gradient strength:

$$S_k = S_c''G \tag{7}$$

3.3. Orientation assignment

We use the directional gradient calculated by the CoM filter in our edge detection step to estimate the corner orientation. The orientation of a corner can be calculated as:

$$\theta = atan2(G_y, G_x) \tag{8}$$

4. PERFORMANCE EVALUATION

Most of the methods mentioned above have been implemented in the OpenCV Library [24]. This library has been widely used in various computer vision applications and the source code is gradually optimized to achieve real-time performance. Therefore, with the aim of providing a constructive reference for later research and practical applications, all the compared and evaluated algorithms are from OpenCV implementation.

The evaluation of keypoint detectors has been exhaustively studied in the literature. The testing environment used in this paper is the well-known dataset introduced by Mikolajczyk and Schmid [25]. We present the repeatability score which represents the correspondence

Fig. 3. Detected keypoints on the *graf* image using SICK3.

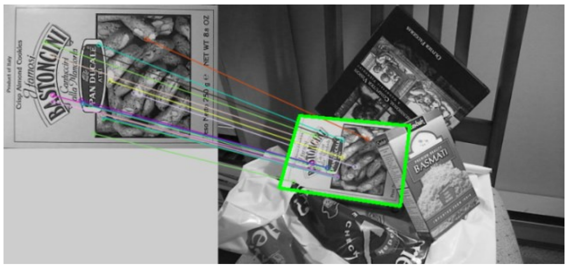

Fig. 4. Matching test on planar images using SICK3.

Detector	SIFT	SURF	ORB	BRISK	SICK8	SICK3
Time(ms)	5024	807	74	180	510	147

Table 1. Time consumption of keypoint detectors. Results are averaged on all images in the Mikolajczyk dataset.

quality of keypoint pairs over two images.

We implemented SICK using 8 filter layers (SICK8) and SICK using 3 filter layers (SICK3). SICK8 is implemented in order to achieve the best performance and SICK3 is implemented to test its speed and performance limitations. Fig. 5 illustrates the evaluation results of various detectors. SICK outperforms other methods under scale changes, viewpoint changes and JPEG compression, especially under scale changes. The performance of FAST, ORB, and BRISK decrease dramatically when the scale increases. SICK's performance under blur change is between SIFT's and SURF's performance. SICK's performance under light change is slightly weaker than other methods but still within an acceptable range, similar to SIFT. This is affected by the edge threshold set in our algorithm and can be improved in the future using dynamic thresholding.

Table 1 shows the time consumption of keypoint detectors. All algorithms are running on an Intel Duo core (4 threads) 2.7 GHZ CPU using a single core single thread. SICK3 is 34 times faster than SIFT and 5.5 time faster than SURF and slightly slower than ORB. Fig. 3 illustrates part of the detected keypoints on the *graf* image, where the size of circles indicate scale and lines indicate direction. A matching test on planar images using SICK3 detector and SURF descriptor is shown in Fig. 4. SICK3 is able to maintain a high performance except for a little quality loss under scale change.

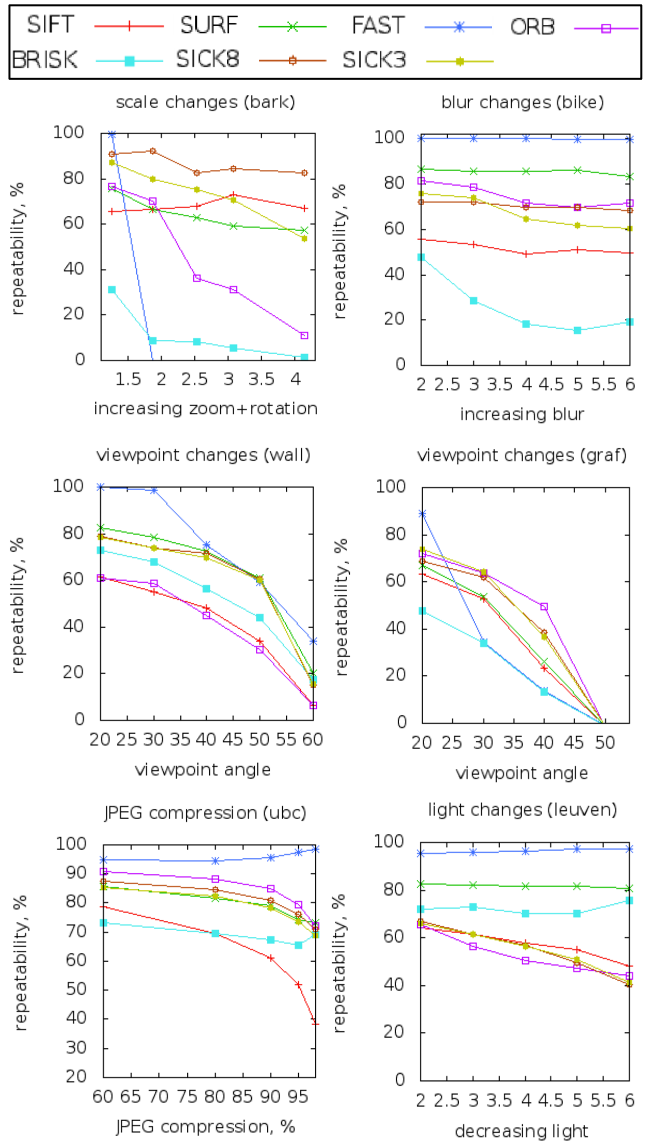

Fig. 5. Detector performance evaluation.

5. CONCLUSION

We have presented a scale-invariant corner keypoint detector that enhances current keypoint detector performance. It outperforms recent state-of-the-art keypoint detectors while it maintains a high computational speed. One advantage is that edge based corner measure is more efficient than corner measure using Harris or Hessian matrix because it is calculated in one dimension. Another advantage is that we use the gradient direction as keypoint orientation and the orientation can be assigned directly without additional cost. Most feature descriptors need a separate orientation assignment step to generate keypoint orientation. SICK overcomes the performance loss that other methods show when the computational speed is increased.

6. REFERENCES

[1] Lowe, David G. "Distinctive image features from scale-invariant keypoints", International journal of computer vision 60.2 (2004): 91-110.

[2] Bay Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006. 404-417.

[3] Rublee, Ethan, et al. "ORB: an efficient alternative to SIFT or SURF", Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[4] Leutenegger, Stefan, Margarita Chli, and Roland Y. Siegwart. "BRISK: Binary robust invariant scalable keypoints", Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[5] Rosten Edward, and Tom Drummond. "Machine learning for high-speed corner detection", Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006. 430-443.

[6] Harris Chris, and Mike Stephens. "A combined corner and edge detector", Alvey vision conference. Vol. 15. 1988.

[7] Lindeberg, Tony. "Feature detection with automatic scale selection."International journal of computer vision 30.2 (1998): 79-116.

[8] Mikolajczyk, Krystian, and Cordelia Schmid. "Indexing based on scale invariant interest points." Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Vol. 1. IEEE, 2001.

[9] Grabner, Michael, Helmut Grabner, and Horst Bischof. "Fast approximated SIFT", Computer Vision–ACCV 2006. Springer Berlin Heidelberg, 2006. 918-927.

[10] Calonder, Michael, et al. "Brief: Binary robust independent elementary features", Computer Vision–ECCV 2010. Springer Berlin Heidelberg, 2010. 778-792.

[11] Zheng, Zhiqiang, Han Wang, and Eam Khwang Teoh. "Analysis of gray level corner detection." Pattern Recognition Letters 20.2 (1999): 149-162.

[12] Cooper, James, Svetha Venkatesh, and Leslie Kitchen. "Early jump-out corner detectors." Pattern Analysis and Machine Intelligence, IEEE Transactions on15.8 (1993): 823-828.

[13] Sheu, Hsin-Teng, and Wu-Chih Hu. "A rotationally invariant two-phase scheme for corner detection". Pattern Recognition 29.5 (1996): 819-828.

[14] Noble, J. Alison. "Finding corners." Image and Vision Computing 6.2 (1988): 121-128.

[15] Mokhtarian, Farzin, and Riku Suomela. "Robust image corner detection through curvature scale space." Pattern Analysis and Machine Intelligence, IEEE Transactions on 20.12 (1998): 1376-1381.

[16] Konstantinos G. Derpanis, “Integral image-based representations”, Department of Computer Science and Engineering York University Paper, 2007, Vol. 1 No. 2 pp. 1-6

[17] Franklin C. Crow, "Summed-area tables for texture mapping", ACM SIGGRAPH Computer Graphics. Vol. 18, No. 3, 1984

[18] Bo Li, Aleksandar Jevtic, Ulrik Söderström, Shafiq Ur Réhman, Haibo Li, “Fast Edge Detection by Center of Mass”, in Proceedings of the 1st IEEE/IIAE International Conference on Intelligent Systems and Image Processing (ICISIP 2013), 26-27September 2013, Kitakyushu, Japan.

[19] Canny, John. "A computational approach to edge detection", Pattern Analysis and Machine Intelligence, IEEE Transactions on 6 (1986): 679-698.

[20] Ziou, Djemel, and Salvatore Tabbone. "Edge detection techniques-an overview", Pattern Recognition and Image Analysis C/C Of Raspoznavaniye Obrazov I Analiz Izobrazhenii 8 (1998): 537-559.

[21] Lindeberg, Tony. "Edge detection and ridge detection with automatic scale selection", International Journal of Computer Vision 30.2 (1998): 117-156.

[22] Bo Li, Shafiq Ur Réhman, Ulrik Söderström, “Independent Thresholds on Multi-scale Gradient Images”, in Proceedings of the 1st IEEE/IIAE International Conference on Intelligent Systems and Image Processing (ICISIP 2013), 26-27September 2013, Kitakyushu, Japan.

[23] Bo Li, Ulrik Söderström, Shafiq Ur Réhman, Haibo Li, “Restricted Hysteresis Reduce Redundancy in Edge Detection”, in Journal of Signal and Information Processing, August 2013.

[24] The OpenCV library, http://opencv.org, acessed 01.24.2014.

[25] Mikolajczyk, Krystian, et al. "A comparison of affine region detectors", International journal of computer vision 65.1-2 (2005): 43-72.

[26] Lindeberg, Tony. Scale-space theory in computer vision. Springer, 1993.