

# A Harris-like Scale Invariant Feature Detector

Yinan Yu\*, Kaiqi Huang, and Tieniu Tan

National Laboratory of Pattern Recognition  
Institute of Automation, Chinese Academy of Sciences  
`{ynyu, kqhuang, tnt}@nlpr.ia.ac.cn`

**Abstract.** Image feature detection is a fundamental issue in computer vision. SIFT[1] and SURF[2] are very effective in scale-space feature detection, but their stabilities are not good enough because unstable features such as edges are often detected even if they use edge suppression as a post-treatment. Inspired by Harris function[3], we extend Harris to scale-space and propose a novel method - Harris-like Scale Invariant Feature Detector (HLSIFD). Different to Harris-Laplace which is a hybrid method of Harris and Laplace, HLSIFD uses Hessian Matrix which is proved to be more stable in scale-space than Harris matrix. Unlike other methods suppressing edges in a sudden way(SIFT) or ignoring it(SURF), HLSIFD suppresses edges smoothly and uniformly, so fewer fake points are detected by HLSIFD. The approach is evaluated on public databases and in real scenes. Compared to the state of arts feature detectors: SIFT and SURF, HLSIFD shows high performance of HLSIFD.

**Key words:** Feature detector, image matching, scale invariant, harris

## 1 Introduction

Feature detection is a hot topic in computer vision which is widely used in many areas, such as tracking[4], image stitching[5], 3D reconstruction[6, 7], camera calibration[8], SLAM system[9], object classification and recognition[10]. In recent years, a lot of work has been done on effective feature detection[1–4, 11–17]. CSS[11], proposed by Mokhtarian, considers edge with high curvature as corner; Susan[12] and Fast[13] compare the intensity of each pixel with others in its neighborhood to find the corner-like points. Harris[3] constructs a corner model and proposes the Harris Cornerness Function. Similar to Harris, a method proposed by Shi & Tomasis[4] considers the minimum eigenvalue of the Harris Matrix as the cornerness. Apart from the corner detectors mentioned above, SIFT[1], as a region detector, is an approximation of Laplace of Gaussian(LoG), which is proved to be stable and effective in scale-space[14]. From another point

---

\* This work is supported by National Basic Research Program of China (Grant No. 2004CB318100), National Natural Science Foundation of China (Grant No. 60736018, 60723005), NLPR 2008NLPRZY-2, National Hi-Tech Research and Development Program of China (2009AA01Z318), National Science Founding (60605014, 60875021).

of view, LoG is the sum of the eigenvalues of the image second derivation matrix(Hessian Matrix). Another region detector which is also based on Hessian Matrix is Determinant of Hessian(DoH). This method calculates the product of the two eigenvalues of the Hessian Matrix, and also performs well in scale feature detection. SURF[2], which is proposed by Herbert Bay et al. and proven to be efficient, is an approximation of DoH.

It can be seen from above that SIFT(DoG/LoG) and SURF(DoH) are both trying to describe the Hessian Matrix with its eigenvalues. These two methods are good at representation of Hessian Matrix, but neither of them describes the matrix very well, for both of them lose much important information about the ratio of the eigenvalues. This value reflects the edge-likeness of areas. Experiments show that edge areas are unstable in localization and not discriminative in feature description. Edges increase false matches and reduce the accuracy. To cope with this problem, SIFT uses an edge suppression step with a threshold to get rid of edge-like features. The sudden cut process degrades the stability of performance and can not drive out edge-like areas uniformly. SURF, which is calculated very fast, detects a lot of key points full of edges and meaningless features, for it ignores this problem. In order to solve this problem, we propose a new algorithm for feature detection(HLSIFD). To prove the effectiveness of the proposed method, we compare it to the state of arts feature detectors: SIFT and SURF. Experimental results show that HLSIFD outperforms these two methods.

The rest of this paper is organized as follows. Section 2 presents a detailed analysis of feature model and eigenvalue description for feature detection. We then show the proposed detector in Section 3. Finally, experimental results are shown in Section 4 and Section 5 concludes this paper.

## 2 Feature model and eigenvalue description

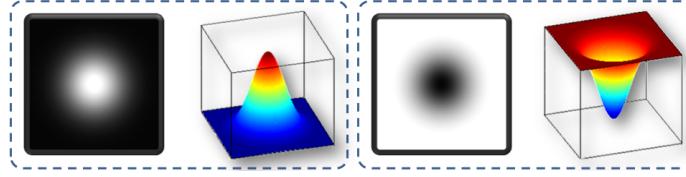
Generally, image smoothed by Gaussian filter can be modeled as a Multi-Gaussian-Mix matrix. So feature can also be modeled as Local Gaussian ellipse hill, and the key point is the hilltop, as shown in Fig 1 for an intuitive illustration. The function of this model can be described as follows:

$$M = \kappa \exp \left( -\frac{1}{2} (x, y) \Sigma^{-1} (x, y)^T \right) + T \quad (1)$$

where  $\kappa, T$  are parameters and  $\Sigma$  is the Gaussian Covariance Matrix. Note that the second derivative matrix(Hessian Matrix) of the model at point  $(0, 0)$  is:

$$H = \begin{bmatrix} M_{xx}(0, 0) & M_{xy}(0, 0) \\ M_{xy}(0, 0) & M_{yy}(0, 0) \end{bmatrix} = -\kappa \Sigma^{-1} \quad (2)$$

The Hessian Matrix is composed of the amplitude and the Covariance Matrix which contain most of the model information. The ratio of the eigenvalues of the Hessian Matrix indicates the eccentricity of the Gaussian model, and the eigenvectors depict the orientation. It is important to point out that, the eigenvalues



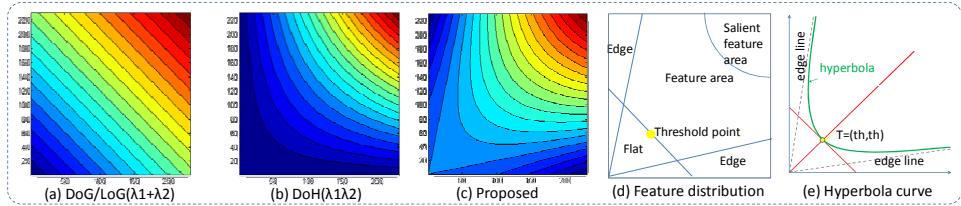
**Fig. 1.** An intuitive illustration for Gaussian Feature Model in 2D and 3D views. Models with positive  $\kappa$  are shown in the left block, while models with negative  $\kappa$  are shown in the right block.

of the Hessian Matrix also have high response in corner areas and edge areas. So one of the advantages of Gaussian model based method is that it can detect both Gaussian-like areas and corner-like areas in scale-space. This property enriches the feature abundance of Hessian based methods. The Hessian Matrix of smoothed image  $L = g * I$  is:

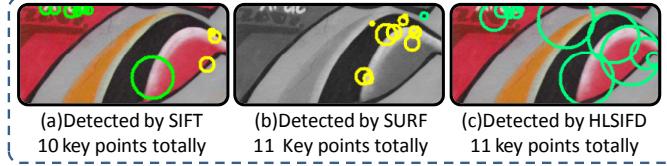
$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \Rightarrow \begin{cases} \det(H) = \lambda_1\lambda_2 = L_{xx}L_{yy} - L_{xy}^2 = DoH(I) \\ tr(H) = \lambda_1 + \lambda_2 = L_{xx} + L_{yy} = LoG(I) \approx DoG(I) \end{cases} \quad (3)$$

where  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of Matrix  $H$ . In SIFT, the response of the function(DoG) increases linearly according to the eigenvalues, as shown in Fig 2-(a). We can find that the function has high response even in the edge area where  $\max(\lambda_1/\lambda_2, \lambda_2/\lambda_1)$  is large. So in SIFT, an edge suppression[1] is added to get rid of edges, leading to unstableness of features on the boundary of the edges area.

SURF uses the multiplication of the eigenvalues, as shown in Fig 2-(b). The response is similar with SIFT(LoG/DoG) in the Salient Feature Area where both  $\lambda_1$  and  $\lambda_2$  are high. The problem of SURF is almost the same with SIFT. The product will be large when either of the  $\lambda_1$  or  $\lambda_2$  is high. However, SURF does not tackle this problem, resulting in a lot of edges detected, even with a high threshold. This property reduces directly the Repeatability Score. Experiments show that using a step like edge suppression can not tackle this problem well, which has been explained in the last paragraph. Some examples are shown in Fig 3-(a,b).



**Fig. 2.** From left to right: (a)response of  $\lambda_1 + \lambda_2$ , (b)response of  $\lambda_1\lambda_2$ , (c)Proposed method, (d)Feature distribution in eigenvalues plane,(e)Hyperbola curve. The angle from edge line to coordinate is  $\alpha$  and from edge line to  $45^\circ$  line is  $\theta$ .  $T$  is the threshold point. The hyperbola is decided by  $\theta$ (or  $\alpha$ ) and  $T$ .



**Fig. 3.** Key points detected by SIFT, SURF and our method. Our method is proposed in Section 3. Green circles are good feature areas; Yellow circles are edge areas which are not stable.

Actually, two small eigenvalues represent the flat area in image, while a small value and a large one represent edges. The area is stable key point when the two eigenvalues are large, as shown in Fig 2-(d). Thus, using a function to describe the three point mentioned above is the standard mission of feature detectors. Supposing the feature-like point has high response, the detector function should have these four properties:

1. Low response in flat area.
2. Low response in edge-like area.
3. High response in feature-like areas.
4. Smooth function surface.

Our extensive experimental results show that the last property is very important for the stability of detector. In the following section, a novel algorithm for feature detection is described in detail based on the four properties.

### 3 Our method: HLSIFD

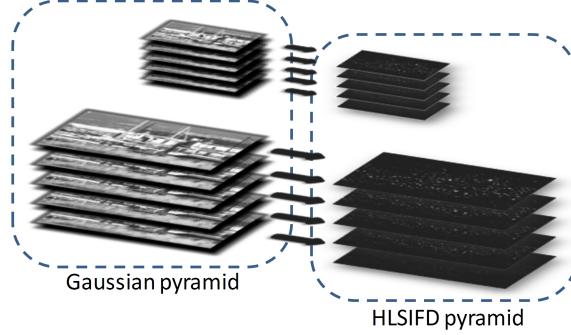
#### 3.1 Detector procedure

Denote an image as  $I$ . Since the image is usually noisy, a Gaussian filter is used with  $\sigma = \sigma_s$  to smooth the original image:  $I_{\sigma_s} = g(\sigma_s) * I$ . In order to detect points in scale-space, image pyramid is constructed by smoothing the image with a group of Gaussian Filters. To be more efficient, we down sample the pyramid every  $s$  layers, and form several octaves, which are shown in the left of Fig 4. The scale interval of two consecutive octaves is 2. Different from SIFT, we must construct  $s+2$  layers per octave and the Gaussian Filters are sampled uniformly in scale-space with a factor of  $k$ , where  $k = 2^{1/s}$ . For each octave, The Gaussian Filters are:  $g_i = g(\sigma_i)$ , and  $\sigma_i = k^i \sigma_0$ ,  $i \leq s$  and the Hessian Matrix of the image is:

$$H(\sigma_i) = \begin{bmatrix} L_{xx}(\sigma_i, \sigma_s) & L_{xy}(\sigma_i, \sigma_s) \\ L_{xy}(\sigma_i, \sigma_s) & L_{yy}(\sigma_i, \sigma_s) \end{bmatrix} \quad (4)$$

To satisfy the four properties mentioned in last section, we construct a hyperbola function with a rotation of  $45^\circ$  anticlockwise. The vertex point of the function stands on the threshold line to filter noises, and the asymptotes are set to filter edges uniformly, as shown in Fig 2-(e). Therefore, the cornerness function is:

$$\frac{(\lambda_1 + \lambda_2)^2}{2} - \frac{(\lambda_1 - \lambda_2)^2}{2t g^2(\theta)} - \frac{(2t h)^2}{2} \quad (5)$$



**Fig. 4.** Gaussian pyramid is shown in left and HLSIFD pyramid is shown in right. Number of octave is decided by the image size, here we only draw up 2 bottom octaves and each octave 5 layers. For each layer, calculate the corner response using Equation (9) from Gaussian pyramid to HLSIFD pyramid

where  $\lambda_1$  and  $\lambda_2$  are the two eigenvalues of the Hessian Matrix. Note that  $\alpha + \theta = \pi/4$ , and the function can be rewritten as:

$$4 \left( \frac{1+tg(\alpha)}{1-tg(\alpha)} \right)^2 \left( \lambda_1 \lambda_2 - \frac{1}{4} \left( 1 - \left( \frac{1-tg(\alpha)}{1+tg(\alpha)} \right)^2 \right) (\lambda_1 + \lambda_2)^2 \right) - 2th^2 \quad (6)$$

Let  $\gamma$  represents  $4 \left( \frac{1+tg(\alpha)}{1-tg(\alpha)} \right)^2$ ,  $\kappa$  represents  $\frac{1}{4} \left( 1 - \left( \frac{1-tg(\alpha)}{1+tg(\alpha)} \right)^2 \right)$ , the function can be written briefly as:

$$\gamma \left( \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 \right) - 2th^2 \quad (7)$$

After normalizing the coefficient of  $\lambda_1 \lambda_2$  and letting  $th'$  equals  $\frac{2th^2}{\gamma}$ , the final cornerness is:

$$\lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 - th' \quad (8)$$

Function (8) has the same form as Harris Cornerness Function. The difference is Harris corner is based on Harris Matrix which represents image edge curvature, while Function (8) is based on Hessian Matrix which represents the Gaussian Ellipse Model. In order to detect feature in scale-space, we use the scale normalized Hessian Matrix to express cornerness:

$$D(x, y, \sigma) = \sigma^4 (\lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 - th') \quad (9)$$

where  $\sigma^4$  is a normalization coefficient in scale-space. We call this method the Harris-like Scale Invariant Feature Detector(HLSIFD). The Gaussian pyramid is filtered by Equation (9) to get the HLSIFD pyramid, as shown in the right part of Fig 4. Our HLSIFD is:

$$\begin{aligned} D(\sigma_l) &= \sigma_l^4 (\lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 - th') \\ &= \sigma_l^4 (\det(H(\sigma_l)) - \kappa \text{trace}^2(H(\sigma_l)) - th') \\ &= \sigma_l^4 (L_{xx}(\sigma_l, \sigma_i) L_{yy}(\sigma_l, \sigma_i) - L_{xy}(\sigma_l, \sigma_i)^2 \\ &\quad - \kappa (L_{xx}(\sigma_l, \sigma_i) + L_{yy}(\sigma_l, \sigma_i))^2 - th') \end{aligned} \quad (10)$$

$$\kappa = \frac{1}{4} \left( 1 - \left( \frac{1-tg(\alpha)}{1+tg(\alpha)} \right)^2 \right) \Leftrightarrow tg(\alpha) = \frac{1-\sqrt{1-4\kappa}}{1+\sqrt{1-4\kappa}} \quad (11)$$

where  $\kappa \in [0, 0.25]$ . On one hand Function (10) degenerates into the Determinant of Hessian(DoH) when  $\kappa = 0$ . On the other hand, when  $\kappa \rightarrow 0.25 \Rightarrow \alpha \rightarrow 45^0$ , only the areas with approximately equivalent Hessian eigenvalues would be selected. An un-max-suppression step is used to get the local peak which is considered as a key-point in  $3 \times 3 \times 3$  neighborhood in the HLSIFD pyramid. Experimentally, a larger neighborhood is not helpful for increasing the performance. Negative minimums should also be discarded, since they may be edges, noise or even worse saddle points with opposite eigenvalues.

### 3.2 Matching and description procedure

We use Repeatability Score(RS)[18] to evaluate the performance of detector. This score is the ratio of the number of correct matches and key points totally detected on reference image. In order to get this score, we detect key points in reference image and test image first. Then all key points are described by SIFT descriptor for easy comparison. For each key point  $A$  in the reference image, we calculate the feature distance(Euclid Distance) from every point detected in the test image to  $A$ . Next, the first and second nearest point  $B$  and  $C$  are found in the test image. Supposing the feature distance between  $A$  and  $B$  is  $d_1$ , and that between  $A$  and  $C$  is  $d_2$ , if  $d_1/d_2 < t$ ,  $A$  matchs  $B$ . Otherwise  $A$  does not match any point in the test image. Finally, RANSAC [19] algorithm is used to eliminate fake matchings and selects the correct matching pairs from all matching pairs.

## 4 Experimental Results

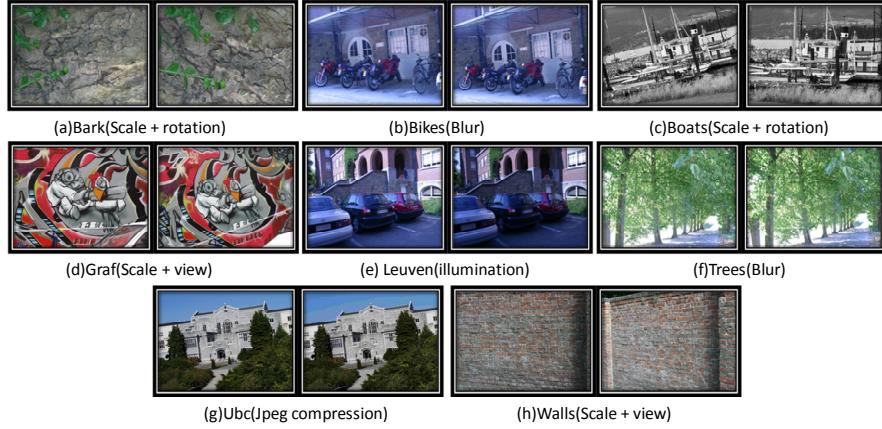
To evaluate the performance of the proposed detector, we do experiments on the database provided by Mikolajczyk<sup>1</sup> in comparison to the state of arts: SIFT<sup>2</sup> and SURF<sup>3</sup>. This database contains 8 groups images with challenging transformations. Parts of them are shown in Fig 5.

We test our method with the first two lightly transformed images at each group first. The total number of features detected, true positive(Repeatability/correct match) and precision(ratio between number of correct match and total match) were calculated for comparison, as shown in Table 1. Our method is better in the true positive, not that our method increases the total matching rate, but our method increases the precision. The precision of our method is always higher than others. Experimentally, detector with a higher precision denotes that the key points detected are more accurate and stable, and there are fewer fake, invalid or meaningless key points.  $t$  and  $\kappa$  are set to 0.95 and 0.1 in this experiment. For more experiments about our method, we tested HLSIFD in all the images of

<sup>1</sup> <http://www.robots.ox.ac.uk/~vgg/research/affine/index.html>

<sup>2</sup> Provided by Rob Hess:<http://web.engr.oregonstate.edu/~hess/index.html>

<sup>3</sup> Provided by OpenCV 1.1:<http://sourceforge.net/projects/opencvlibrary/>



**Fig. 5.** Database with 8 groups images provided by Mikolajczyk. Each group contains one or two transformations with 6 images and parts of them are shown.

**Table 1.** Experimental result on the low transformed images

Detector	SIFT			SURF			Ours(HLSIFD)		
	total	tp <sup>a</sup> (%)	p <sup>b</sup> (%)	total	tp(%)	p(%)	total	tp(%)	p(%)
Bark(s <sup>c</sup> +r)	4162	14.4	20.5	3481	6.29	11.8	3588	24.6	31.0
Bikes(b)	3202	23.7	30.3	4019	33.5	44.0	4363	43.8	50.9
Boats(s+r)	7986	20.0	27.1	5056	12.2	18.1	4677	31.5	38.2
Graf(s+a)	2837	33.1	41.3	3342	14.8	22.1	2493	35.4	40.9
Leuven(i)	2131	40.5	49.5	3245	39.4	49.0	2841	55.0	61.8
Trees(b)	11279	9.19	14.0	7684	11.1	19.4	7442	17.2	22.5
Ubc(j)	4511	47.3	58.3	4286	68.8	77.3	4025	74.8	80.3
Walls(s+a)	8218	36.2	47.5	6792	8.84	15.7	8107	42.9	51.2

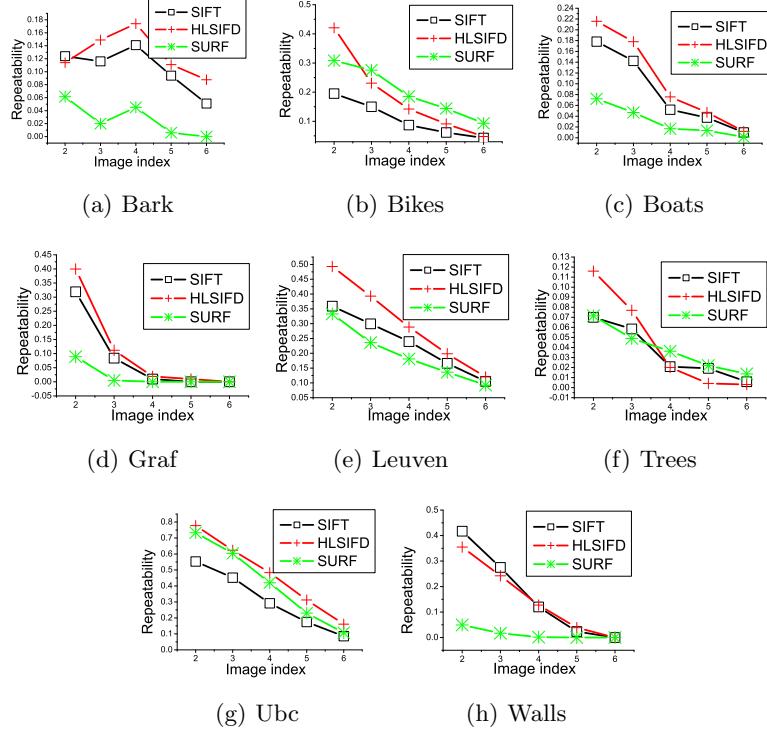
<sup>a</sup> "tp" denotes true positive rate

<sup>b</sup> "p" denotes precision rate

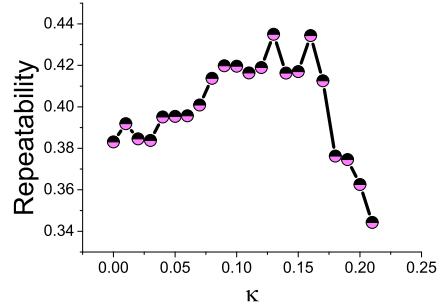
<sup>c</sup> "s", "r", "b", "i", "a", "j" respectively denote scale, rotation, blur, illumination change, affine transformation, and Jepg compression

the database we mentioned above. The results are shown in Fig 6 (a)-(h). The RS of HLSIFD outperforms others in scale, rotation, and affine transformation, since Harris-like function is smooth in eigenvalues space and intensities of edges can be restricted by  $\kappa$ . HLSIFD is better than SIFT and SURF in most groups, but a little weaker than SURF in seriously blurred image, shown in Fig 6 (b). The computation of our method is nearly same with SIFT, because the most time consuming step is pyramid constructing.

$\kappa$  is an important parameter in our method. When  $\kappa = 0$ , it degenerates into Determinant of Hessian(DoH) [17]. We have done many experiments to test the influence of  $\kappa$  to the performance, one of them is shown in Fig 7. We choose  $\kappa$  between 0.04 and 0.15 experimentally and the performance of repeatability would be increased by 10% approximately.

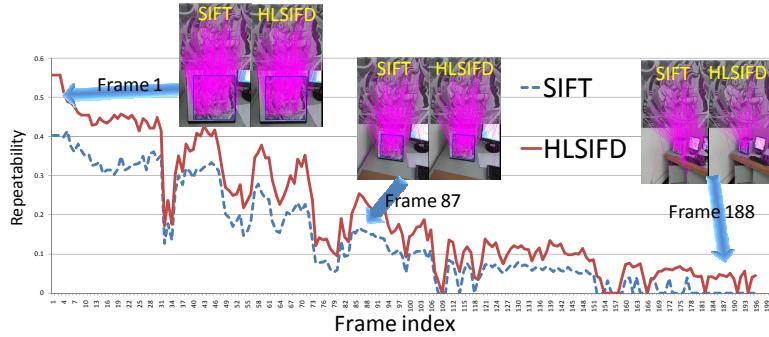


**Fig. 6.** 8 groups experimental results are shown. Use the first image of each group as reference image and others as test image. From left to right, top to down:(a) Bark: scale and rotation change. (b)Bikes: blur. (c)Boats: scale and rotation change. (d)Graf: scale and view change image. (e)Leuven: illumination change. (f)Trees: blur. (g)Ubc: Jpeg compression. (h)Walls: scale and view change.



**Fig. 7.** This figure is one test of influence from  $\kappa$  to the repeatability. The repeatability is increasing with  $\kappa$  from 0 to 0.16, and suddenly drops after 0.16

Using feature detection and image matching in video processing is an important application. We compare our method with SIFT in video sequences matching. We study the reference image and matched all the video frames with it. Then the Affine Transform Matrix is calculated by matching points. The video we used contains a large scale change and a gradual view change from  $0^0$  to  $45^0$ . These transformations are common and have certain representativeness in real scene. Experimental results are shown in Fig 8. The Repeatability Scores of SIFT and HLSIFD reduce continuously with scale and view change. Some motion blurs are presented because of the shaking of our hand-hold camera, and the performance suddenly drops in these frames. The Repeatability Score of HLSIFD is better in most of the time, since it is more stable and detects less fake points by the smooth edge suppression of HLSIFD function.



**Fig. 8.** Experiments on video with scale and view transformation. The reference image was captured from a camera. The results of frame 1, 87 and 188 are shown with SIFT in the left and HLSIFD in the right.

## 5 Conclusions

In this paper, we propose a novel scale invariant feature detector: Harris-like Scale Invariant Feature Detector(HLSIFD). The advantage of this detector is the high precision, since fewer fake points could be detected by the proposed method. Unlike SIFT, our method does not need a post-treatment step to cut edge-like points suddenly which would affect the stability. The proposed method can suppress the unstable fake feature points in a uniform way and increase the feature repeatability. Thus, with fewer meaningless key points, features are more significant. Experimental results show the effectiveness of our method.

In the future, we will use our method in real-time image stitching and SLAM system, so a fast algorithm of the proposed detector will be investigated.

## References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* **60**(2) (2004) 91–110

2. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). *Comput. Vis. Image Underst.* **110**(3) (2008) 346–359
3. Harris, C., Stephens, M.: A combined corner and edge detection. (1988) 147–151
4. Shi, J., Tomasi, C.: Good features to track. Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on (Jun 1994) 593–600
5. Li, Y., Wang, Y., Huang, W., Zhang, Z.: Automatic image stitching using sift. Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on (July 2008) 568–571
6. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. *International Journal of Computer Vision* **59**(3) (2004) 207–232
7. Brown, M., Lowe, D.: Unsupervised 3d object recognition and reconstruction in unordered datasets. 3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on (June 2005) 56–63
8. Telle, B., Aldon, M.J., Ramdani, N.: Camera calibration and 3d reconstruction using interval analysis. Image Analysis and Processing, 2003. Proceedings. 12th International Conference on (2003) 374–379
9. Davison, A., Mayol, W., Murray, D.: Real-time localization and mapping with wearable active vision. Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on (Oct. 2003) 18–27
10. Lisin, D., Mattar, M., Blaschko, M., Learned-Miller, E., Benfield, M.: Combining local and global image features for object class recognition. Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on (June 2005) 47–47
11. Mokhtarian, F., Suomela, R.: Robust image corner detection through curvature scale space. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(12) (1998) 1376–1381
12. Smith, S.M., Brady, J.M.: Susan - a new approach to low level image processing. *International Journal of Computer Vision* **23** (1997) 45–78
13. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. **1** (May 2006) 430–443
14. Lindeberg, T.: Scale-space theory in computer vision. (1994)
15. Lowe, D.G.: Object recognition from local scale-invariant features. Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on **2** (1999) 1150–1157 vol.2
16. Grabner, M., Grabner, H., Bischof, H.: Fast approximated sift. ACCV 2006 (2006) 918–927
17. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. *Int. J. Comput. Vision* **60**(1) (2004) 63–86
18. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. *Int. J. Comput. Vision* **65**(1-2) (2005) 43–72
19. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6) (June 1981) 381–395