

Using Edge SIFT Points for Simple Shape Object Matching

Han Fang¹ , Yule Yuan² , Ling Shen³ , Yong Zhao⁴

Abstract. SIFT the well-known image recognition algorithm performs reliable matching between different views of an object even it is rotated, in different scales, or has different brightness than of the original image. But in our button match works, the SIFT points detected in simple object may be few, and the wrong match between objects that has same textures but different shapes may lead wrong matching results. To address this problem, we only retain the detected SIFT points that on the edges and also compared the edge SIFT points with the classic SIFT points, then combined them to perform a better result. Experiments demonstrate that morphological operation tried to decrease the light problems and get a substantial improvement.

Keywords: objects matching; SIFT; edge; morphological operation

1. Introduction

¹Han Fang
Key Lab. of Integrated Microsystem Sci. & Eng. Applications, Peking University Shenzhen Graduate School, China
e-mail: fanghan@sz.pku.edu.cn

² Yule Yuan
Key Lab. of Integrated Microsystem Sci. & Eng. Applications, Peking University Shenzhen Graduate School, China
e-mail: yuanyl@pkusz.edu.cn

³ Ling Shen
Key Lab. of Integrated Microsystem Sci. & Eng. Applications, Peking University Shenzhen Graduate School, China
e-mail: forrest.lingshen@gmail.com

⁴ Yong Zhao (✉)
Key Lab. of Integrated Microsystem Sci. & Eng. Applications, Peking University Shenzhen Graduate School, China
e-mail: yongzhao@pkusz.edu.cn

In the original SIFT method [3], we detect the extremal points in the DoG(Difference of Gaussians) ,then find the local minima/maxima points of the DoG images across scales , This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint. And low contrast candidate points and edge response points along an edge are discarded, because these points may not detected under different situation and unstable to small noise, that may be a problem with a complex background and the object itself is not simple, but consider in our works, matches for the different shape buttons, all the buttons have simple shapes and background, the edge SIFT points may no longer be useless so we keep the edge points, and the SIFT points that not locate in edge are discarded. We found that the edge points also can lead correct matching, so if we matching the objects that have same textures or materials, combine the edge SIFT points and the accurate of the match improved. Consider the shadow and the effect of illumination, some morphological operation can be useful. All the detect points we compute with SIFT descriptors, which are robust to local affine distortion are then obtained by considering pixels around a radius of the key location, blurring and resampling of local image orientation planes.

Given a set of image descriptors computed from two different images, these image descriptors can be mutually matched by for each point finding the point in the other image domain that minimizes the Euclidean distance between the descriptors represented as 128-dimensional vectors. To suppress matches that could be regarded as possibly ambiguous, the SIFT author Lowe propose then the matches for which the ratio between the distances to the nearest and the next nearest points is less than 0.49 are accepted, but as in our works the detected SIFT points are not many, so we use 0.59 instead.

We build the Kd-tree and use the approximate BBF (Best-Bin-First) method used for finding the nearest neighbor.

2. Related Work

2.1 Detect SIFT points on edges

In Lowe's paper, the points located along the edge are regard poorly determined and therefore unstable to small amounts of noise. A 2×2 Hessian matrix can be used to compute the principal curvatures which will be large across the edge. Hessian matrix, H , computed at the location and scale of the keypoint:

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}$$

The derivatives are estimated by taking differences of neighboring sample points. And to avoid explicitly computing the eigenvalues, let α be the eigenvalue with the largest magnitude and β be the small one, let r be the ratio between the largest magnitude eigenvalue and the smaller one, so that $\alpha=r\beta$. Then,

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

Therefore, to check that the ratio of principal curvatures is below some threshold, r , we only need to check

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$

The keypoints that have a ratio between the principal curvatures greater than $r=10$ are eliminated.

So the work that we detect the edge SIFT points just do the opposite things, we keep the keypoints that have the ratio greater than $r=10$, and eliminated the other points.

Table 2.1 The detect points numbers between classic points and edge points in same size images.

Detect points	1	2	3	4	5	6	7
Classic points($r \leq 10$)	28	94	58	125	117	230	305
Edge points ($r > 10$)	44	130	51	81	122	231	285

As show in the table 2.1, in our works edge SIFT points has the same amounts as the classic SIFT points, so disregard them especially that detected points are not many is not a good way. In some papers show that Harris interest points compute with the SIFT descriptors perform well [1]. The edge SIFT points are also a kind of interest points.



Fig. 2.1 the left image is the classic SIFT points ($r \leq 10$) detected, the right image only keep the SIFT points ($r > 10$) that on the edge.

From Figure Fig. 2.1 we can find out in the same button image, some classic SIFT points distribute by the light spot side, the edge SIFT points are better. But most of the edge SIFT points are in small scales, as the length of the arrow is short which means the points detect in the low scale of the image pyramid.

2.2 Compare two kinds of points

We compare two kinds of points under the same matching images. As shown in figure 2, classic SIFT points have few matches, and if the objects are simple and have same texture, wrong matching can lead wrong result.

Unlike classic SIFT points, the edges SIFT points have more matches, but wrong matches remain a problem.

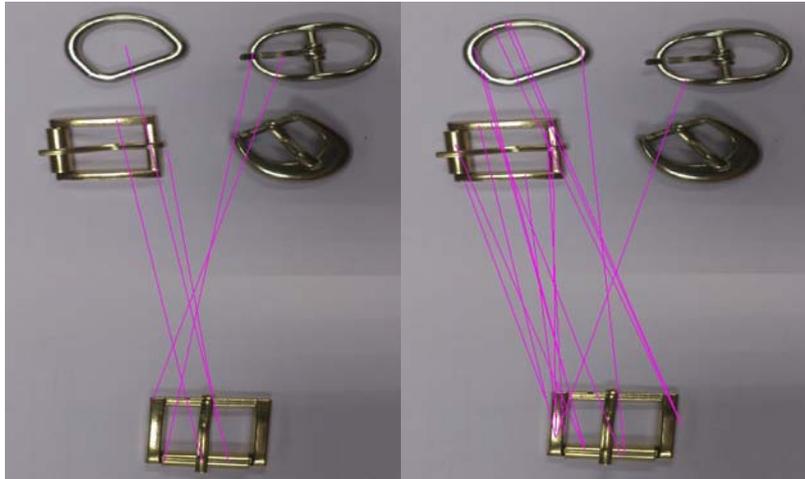


Fig. 2.2 the left image is the classic SIFT points matching result, the right image is the edge SIFT points matching result.

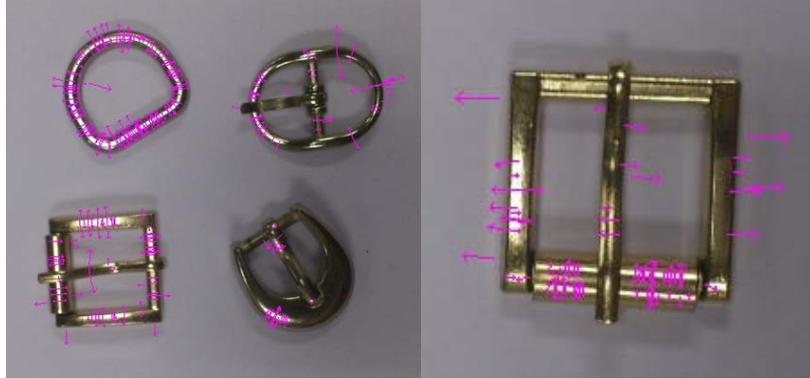


Fig. 2.3 the edge SIFT points detected between two matching images

Figure 2.3 show that the edge SIFT points basically have small scales in the scale-space, which means also the SIFT descriptors only compute the feature that is near the edge SIFT points. While the classic SIFT points detected in large scale will be more robust. Lack of the global information of the objects is not the only problem of edge SIFT descriptors, also because of different illumination, the shadows and the reflected lights bring noise which problem can be more serious when the scale of the descriptor is small. Consider these problems, so we need some pre-processing, for our matching objects are simple, some morphological operations proven to be effective.

2.3 Morphological operations

The other problem we need to solve is the reflected lights leaving large and small spots on the surface.

The opening operations perform well, it can erase small spot, and also the black top-hat transform is useful, which is defined dually as the difference between the closing and the input image.

The black top-hat transform of \mathcal{F} is given by:

$$Tb(\mathcal{F}) = \mathcal{F} \bullet b - \mathcal{F}$$

And the $b(\chi)$ is a grayscale structuring element, we use the common 3×3 square one. The transform can be iterating several times to get a better result.



Fig. 2.4 left one is the original image and the right one is the result of black-hat transform iterate that 5 times.

We are glad to find out that the shadow problem solved too, but when we used the black top-hat transformed image matching, the result got worse, so we used binary images, and doing closing operation, the result improved.

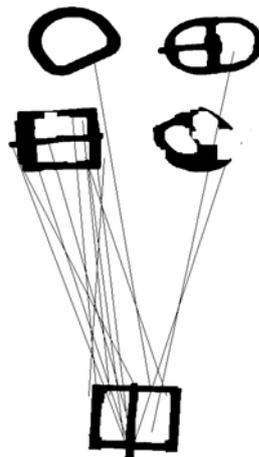


Fig. 2.5 wrong matches decrease when take the binary images and closing operation.

Light condition still remains a problem, as shown in the Figure 2.5 after the morphologic operations the last one of the four buttons appears a broken shape. In order to solve this problem we just guarantee the images in our matching library bright enough and also the image use to match.

3. Experimental Results

In our works we get 50 different shape buttons and take 100 images of these buttons each button have two images under different orientations. All the images have the size of 800×600, and make sure the light is enough.



Fig. 3.1 part of the button images dataset.

We test for three kinds of pre-processing of the images:

1. Grayscale.
2. Grayscale→ Opening.
3. Grayscale→ Black top-hat→ Otsu binarization→ Closing(B→O→C)

And we detect classic SIFT and edges SIFT points, then use SIFT descriptor to represent the feature. For the 100 images we build a Kd-tree, and put every image to match, if the same shape button image is in the top 5 matches, that's a correct match. Also we combine all the two kinds of SIFT points. As shown in the table 3.1 though edge SIFT may not perform better than classic SIFT points, but combine them can improve the matches.

Table 3.1 The match result.

Match result	Classic SIFT	Edge SIFT	Combine two
1.Grayscale	55	62	67
2.Opening	71	59	79
3.B→O→C	85	81	94

4. Conclusion

We have presented an experiment result which demonstrate the eliminated edge SIFT points also can achieved effectively for object matching, especially for simple shape objects. The result shows that morphological operations alleviate the problems of illumination which lead the most wrong matches. Combine the morphological operation and the edge SIFT points get a substantial improvement.

5. References

1. Azad P., Asfour T., Dillmann R.: Combining Harris interest points and the SIFT descriptor for fast scale-invariant object recognition. Intelligent Robots and Systems, IROS 2009 IEEE/RSJ International Conference on (2009)
2. Belongie. S., Malik J., Puzicha J.: Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol. 24 (2002) p. 509-522
3. Lowe, D. G.: *Object recognition from local scale-invariant features*, Computer Vision. The Proceedings of the Seventh IEEE International Conference on (1999)
4. Lindeberg T., Feature detection with automatic scale selection. International Journal of Computer Vision, 1998. 30.
5. Mikolajczyk K., Schmid C. Scale & affine invariant interest point detectors. International Journal on Computer Vision, 2004, 60
6. Nibouche O., Bouridane A., Crookes D., Gueham M., Laadjel M.: Retrieval of shoemarks using Harris points and SIFT descriptor. Image Processing (ICIP), 2009 16th IEEE International Conference on (2009)
7. Shao Wei, Li Na, Sun Lijuan, Sui Shulin, Liu Xiangpeng: Two improved methods of SIFT algorithm combined with harris. Control and Decision Conference (CCDC), 2012 24th Chinese (2012)
8. Schaffalitzky F., Zisserman A. Multi-View Matching for Unordered Image Sets, Proc. Seventh European Conf. Computer Vision, 2002.
9. Tuytelaars T., Mikolajczyk K. Local Invariant Feature Detectors: Foundations and Trends in Computer Graphics and Vision, v3, 2008.
10. Vedaaldi A., An implementation of SIFT detector and descriptor.