INTEGRATING SIFT AND HARRIS CORNER DETECTOR FOR DEFINITE KEYPOINT EXTRACTION FOR IMAGE MATCHING

Cheng-Yi, Chen[1] and Shih-Hong, Chio[2]

[1]Graduate student, Dept. of Land Economics, National Chengchi University,
NO.64, Sec.2, ZhiNan Rd., Wenshan District, Taipei City 11605, Taiwan; Tel: + 886-2-29393091#50621;
E-mail: 99257023@nccu.edu.tw

[2]Associate Professor, Dept. of Land Economics, National Chengchi University,
NO.64, Sec.2, ZhiNan Rd., Wenshan District, Taipei City 11605, Taiwan; Tel: + 886-2-29393091#51657;
E-mail: chio0119@nccu.edu.tw

KEY WORDS: Feature Extraction, SIFT, Harris Corner Detector, Image Matching

ABSTRACT: Feature point extraction automatically instead of manually is important and can improve the efficiency for photogrammetric tasks. Scale invariant feature transform (SIFT) is an important algorithm developed by Lowe (2004) to extract keypoints with invariance on scale and rotation for image matching and registration. Not all these kinds of keypoints do correspond to the definite position of object points in real world, therefore not all of them are suitable for precise localization in photogrammetric task. The definite keypoints can be used for image matching and precise location of object points. For photogrammetric applications, Harris corner detector is often employed to detect definite and precise keypoints, but those kinds of keypoints do not have the characteristics of invariance on scale and rotation. Therefore, this study will integrate SIFT and Harris corner detector to extract the precise and definite keypoints for image matching in order to determine the precise location of object points. Meanwhile, the efficiency of image matching from the extracted keypoints will be investigated in this study.

1. INTRODUCTION

Finding keypoints in image is an important aspect in the field of computer vision and photogrammetry. There are two types of keypoint: Point-like keypoints and Blob-like keypoints. Point-like keypoints stand for specific point, like a junction or a cross point. Blob-like keypoints refer to small area or regions(Förstner, 2009). Therefore keypoints have different types that can be extracted by suitable keypoint extraction methods. Some of them can only extract point-like keypoints, others can extract blob-like keypoints, and the others can extract both. In computer vision field, these two kinds of keypoints are both important, but in photogrammetric task only needs point-like keypoints. Because photogrammetric task needs precise points extracted at an exact location on the object surface, surveyors can acquire the coordinates of the extracted points by photogrammetric principles.

Moravec corner detector (Moravec, 1980) is one of the earliest corner detector, and it also defined what the corner is. It makes the computer can automatically extracted image features, and lets photogrammetric task can be more efficient. Then, Harris and Stephens improved Moravec corner detector (Harris, 1988) makes the computer can more accurately identify the feature points. This method always finds the keypoints on the corner that is useful for photogrammetric task so it is one of the methods currently used. But it only provides the image coordinates of the point during the image matching, the imaging condition of the images must be quite close to each other. In other words, the images' scale, brightness, rotation, etc. must be very similar, or matching will be fail. This is not a big problem in aerial photography, but in close-range photogrammetry this is a serious problem. Because the orientation and scale of images in close-range photogrammetry are quite different, only the image coordinates of keypoints is not sufficient to get more correct image matching result possible in the case of image matching operation .

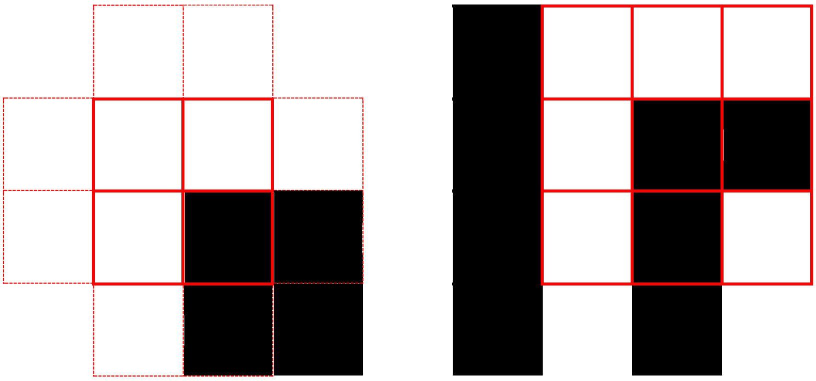
Scale invariant feature transform (SIFT) is an important algorithm developed by Lowe (2004) to extract keypoints with invariance on scale and rotation for image matching and registration. It can overcome the different orientation and scale in close-range photogrammetry, and it provide every keypoint a unique descriptor, these descriptors just like the fingerprint of keypoint, every single keypoint in different images should have the same descriptor in theory. So it is useful in image matching, these descriptors can be provided for efficient image matching. Even thought SIFT can find large numbers of keypoints in single image, these keypoints are not only point-like keypoints but also blob-like keypoints. It can be said that SIFT finds numerical keypoints instead of geometric keypoints , that is why SIFT can find both point-like keypoints and blob-like keypoints. As mentioned earlier, blob-like keypoints are less suitable for photogrammetry, therefore highly accurate photogrammetric operations must exclude blob-like keypoints.In addition, because keypoints with descriptors is more suitable to match corrreponding points for close-range photogrammetry, Contrast context histogram (CCH) descriptor (Huang, 2008) use Harris corner detector to extract keypoints and give every keypoints a descriptor. This approach can ensure that every keypoints are all point-like keypoints and these keypoints with descriptor are more suitable for image matching.

2. HARRIS-SIFT ALGORITHM

In this paper, Harris-SIFT algorithm will be developed to extract the keypoints by Harris corner detector, and to provide a descriptor for each keypoint by the concept of SIFT. The followings will describe it in more details.

2.1 Harris Corner Detector

When it comes to Harris corner detector, it should explain the Moravec detector firstly. Moravec detector uses a moving window to calculate the sum of the gradient's difference. If the difference is larger than a threshold, the point is called keypoint, likes Figure 1(a). When window size increase, it will fail at the area with alternating light and dark because the gradient of the positive and negative offset, likes Figure 1(b).



(a) (b)

Figure 1. (a) The red window is moving on the image, when at the center of the image it will have the largest sum of the gradient's difference. So there should have a corner point. (b) The red window cross the bright area and dark area, it will fail to find a corner because it makes the sum of the gradient's difference small.

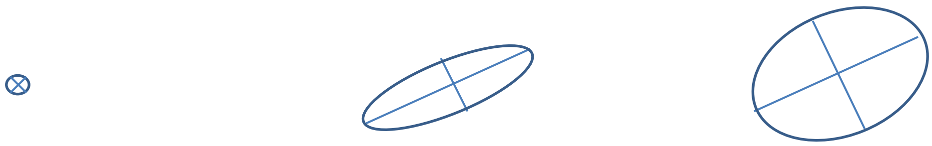
Harris used numerical methods to improved Moravec detector, using a matrix calculating the eigenvalues to find the corner or edge, the equation is below:

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} G_x^2 & G_x G_y \\ G_x G_y & G_y^2 \end{bmatrix} = \begin{bmatrix} \langle G_x^2 \rangle & \langle G_x G_y \rangle \\ \langle G_x G_y \rangle & \langle G_y^2 \rangle \end{bmatrix} \quad (1)$$

Where A is the structure tensor, w(u,v) is a window over the area (u,v), G_x is the gradient of x-direction, G_y is the gradient of y-direction, and angle brackets denote averaging. This matrix is a Harris Matrix, by finding two eigenvalues to determine the point is corner, edge or nothing. Let λ_1 is the bigger eigenvalue, and λ_2 is the smaller eigenvalue, the rule is shown below:

1. If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$ then this point (x,y) is not corner point or edge.
2. If λ_1 has some large positive value and $\lambda_2 \approx 0$ then this point (x,y) is on edge.
3. If λ_1 and λ_2 have some large positive values then this point (x,y) is on corner.

It can be said that the two eigenvectors of Harris matrix is the two directions of point, and the two eigenvalues is the strength of the direction, the three rules can be represented by Figure 2:



(a) (b) (c)

Figure 2. (a) Two eigenvalues are both very small, so it is not an obvious point. (b) Only one eigenvalue is very big, so it is on the edge. (c) Both two eigenvalues are very big, so it is a corner.

2.2 Scale Invariant Feature Transform (SIFT)

SIFT is an algorithm to extract and describe the keypoints in images. It can be applied in 3D modeling, object recognition, video tracking, etc. It can be used to match image features on different images even though these images have different scales and rotations. Because of the above-mentioned features, SIFT is useful in close-range photogrammetry.

In original paper, SIFT have four main steps (Lowe, 2004):

1. Scale-space extreme detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

But for illustration, the following six steps will explain the SIFT alogrithm:

1. Building a scale space:
First, SIFT creates different octaves and scales of original image, the number of octaves and scales depends on the original image. Secondly, SIFT uses original image to generate progressively blurred out images, resize the original image to half size and generate blurred out images again. And keep repeating. But original paper suggests that 4 octaves and 5 blur levels are ideal. When blurring images, it used Gaussian Smoothing.

2. Laplacian of Gaussian approximation:
Laplacian of Gaussian blurs an image a little, then calculates second order derivatives on it. After doing this, edges and corners will locate on the image. But second order derivative is sensitive to noise. So here use another way to approximate to Laplacian of Gaussian. The images created in step 1 will be calculated the difference between two consecutive scales, like Figure 3.

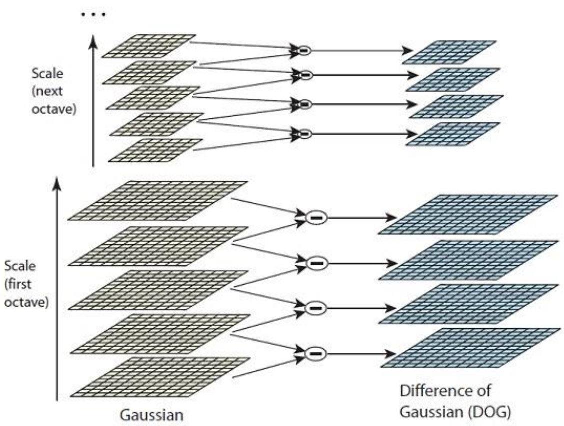


Figure 3. Different of Gaussian (DOG) (Lowe, 2004).

3. Finding Keypoints:
A lot of points be found in step 2, then SIFT will find appropriate point. For each point, take the eight points around it and nine points at the same location of its upper and lower image, total 26 neighbor points (shown in Figure 4). If the DOG value of this point is the maxima or minima by comparing to its 26 neighbor points, then keep the point as a keypoint.

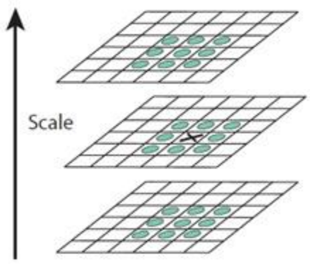


Figure 4. Use the 26 neighbor points to find the X mark whether is maxima or minima (Lowe, 2004).

4. Deleting edge and low contrast points:
SIFT needs to find specific keypoints, but the point found in step 3 may be on the edge or have low contrast.. For eliminating low contrast points, it can establish a threshold value to filter low-contrast points. For the point on the edge, use the Harris Matrix to find the edge point.

5. Determining the orientation of keypoint:
This step will give every keypoint an orientation. For each keypoint, it will be calculated its magnitude and orientation for all pixels around it, use the following equation:

$$\mathrm{m}(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right) \tag{3}$$

$\mathrm{m}(x,y)$ is magnitude of pixel (x,y) and $\theta(x,y)$ is orientation of pixel (x,y), $L(x,y)$ is the gray value of pixel (x,y), then a histogram is created for determining the key point orientation. 360 degrees will be divided into different regions according to demand. The sum of every magnitude in each regions will be caculated, and the maxima value will be found. The region with maxima is the keypoint orientation . In this study, it uses 4x4 window to calculate histogram.

6. Create SIFT features:
When creating a histogram in step 5, it is recommended that divide 360 degrees into 8 regions. That means 0-44 degree, 45-89 degree, and so on. Then create a 16x16 window (see Figure 5) with a keypoint at center, calculate a histogram every 4x4 region, after all 4x4 regions' histogram done, the descriptor of this keypoint is also done. The descriptor should like Figure 5:

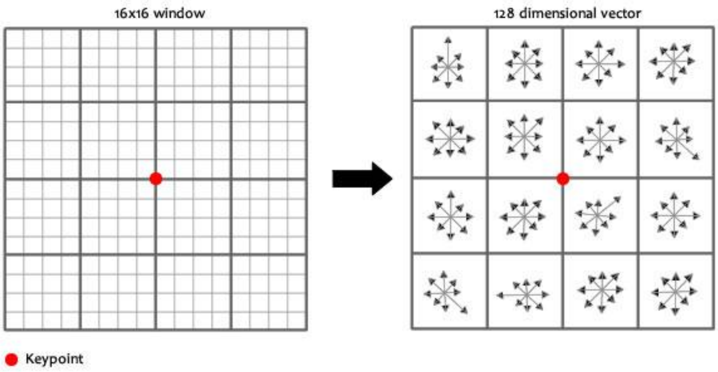


Figure 5. The illustration of keypioint descriptor (Lowe, 2004).

2.3 Harris-SIFT

In this study, step 1 ~ 3 of SIFT's six steps are replaced by Harris corner detector, in order to find the keypoints' location. By using Harris corner detector can ensure that all keypoints are point-like, and using SIFT descriptor can have better matching correctness.

3. EXPERIMENTS

In the test, the matching correctness of Harris-SIFT algorithm developed in this study will be compared with general SIFT algorithm that extract point-like keypoints and blob-like keypoints.

3.1 Sample Images

Sample images is taken in National Chengchi University shown in Figure 6, there are 4 stereo image pairs and image size is 1280x1010, Harris-SIFT and SIFT will use these image to extract and match keypoints.

Figure 6. Sample image pairs.

3.2 Number of Point-like Keypoints and Blob-like Keypoints

Because blob-like keypoints is less suitable for close-range photogrammetry, it will be tested that Harris-SIFT finds less blob-like keypoints than SIFT do, the result is in Table 1. It uses Harris Matrix to test the keypoint is point-like keypoint or blob-like keypoint.

Table 1. Number of blob-like keypoints found in different method

Image	*SIFT (blob-like keypoints / total keypoints)*	*Harris-SIFT (blob-like keypoints / total keypoints)*
1	256/1938	0/266
2	253/1887	0/251
3	180/747	0/95
4	184/799	0/96
5	343/2734	0/391
6	312/2645	0/261
7	376/1360	0/165
8	363/1258	0/98

From the result, Harris-SIFT do find point-like keypoints, therefore Harris-SIFT can be more suitable for close-range photogrammetry. An example is shown in Figure 7.

(a) (b)

Figure 7. Sample of extract keypoints. (a) Using SIFT. (b) Using Harris-SIFT. Red circle stands for point-like keypoints, green circle stands for blob-like keypoints.

3.3 Matching Correctness

Keypoint matching will be based on Euclidean distance of their feature vectors. Two keypoints on different images with the smallest Euclidean distance will be considered as a successful matching . After matching, the correctness will be calculated. Matching correctness is calculated by follow equation, and result is in Table 2.

$$\text{Matching Accuracy} = \frac{\textit{Number of correct matching pairs}}{\textit{Number of total matching pairs}} \times 100\% \quad (4)$$

Table 2. Matching crroectness in different image pairs(unit: %)

Image pairs	*SIFT*	*Harris-SIFT*
1	77.90	50.20

2	91.04	87.23
3	84.51	84.16
4	87.29	84.21

From the result, Harris-SIFT has almost the same correctness than SIFT, but in image pair 1, it has low correctness because image pair 1 have repeated region, makes the descriptors very close to each other. A matching example is shown in Figure 8.

Figure 8. illustrations of matching using Harris-SIFT, correct point pairs are linked by blue lines, wrong point pairs are linked by red lines.

4. CONCLUSION

According to the experimental results, Harris-SIFT performed better than SIFT in keypoints extraction, and matching is almost the same. Especially in keypoints extraction, Harris-SIFT finds no blob-like keypoints. It can be said that Harris-SIFT is better than SIFT for applications in close-range photogrammetry, the points that Harris-SIFT extracts can correspond to the definite position in real world, so surveyor can determinie these points' precise coordinates for subsequent application, such as 3D building model reconstruction automatically. It can enhance operational efficiency of close-range photogrammetry, and make the 3D model more accurate.

In future study, because Harris's algorithm can find edge points (Harris, 1988), it will be tried to matching edge points as 3D line features by improving Harris-SIFT for providing more information to build 3D object model automatically. Furthermore, this study doesn't test images that have different scale and rotation, it will also have tested in future.

REFERENCES

Förstner, W; Dickscheid, T; Schindler, F, 2009. Detecting Interpretable and Accurate Scale-Invariant Keypoints In: 12th IEEE International Conference on Computer Vision (ICCV'09). Kyoto, Japan 2009, pp. 2256-2263.

Hans P. Moravec, 1980, Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University & doctoral dissertation, Stanford University.

Harris, C. G. and Stephens, M. J., 1988. A combined corner and edge detector, Proceedings of the Fourth Alvey Vision Conference, Manchester, United Kingdom, pp. 147-151.

Huang, C.R., Chen, C.S., Chung, P.C., 2008, Contrast context histogram–an efficient discriminating local descriptor for object recognition and image matching. Pattern Recognition (PR 2008) 41(10), 3071–3077.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110.