PROCEEDINGS OF SCIENCE

Region-based Pre-selection of Overlapping Area for the SIFT Algorithm

Yuliang Zhou
School of Computer Science and Engineering, South China University of Technology
Guangzhou, 510006, China
E-mail: `cszhouyl@mail.scut.edu.cn`

Haoxiang Wang
School of Computer Science and Engineering, South China University of Technology
Guangzhou, 510006, China
E-mail: `hxwang@scut.edu.cn`

Maoran Jiang
School of Computer Science and Engineering, South China University of Technology
Guangzhou, 510006, China
E-mail: `csmoranjim@mail.scut.edu.cn`

This paper presents a novel method of determining the overlapping area in advance to accelerate the SIFT (Scale Invariant Feature Transformation) algorithm. The SIFT algorithm can achieve high accuracy in image matching, however, it can be time-consuming when processing large images. The proposed method in this paper pre-selects the overlapping area of input images to speed up the feature extraction of SIFT. This improvement of SIFT can greatly reduce the processing time without affecting the accuracy.

CENet2017
22-23 July 2017
Shanghai, China

PoS(CENet2017)015

1. Introduction

The image matching is fundamental for the computer vision, medical imaging and virtual reality. The SIFT (Scale Invariant Feature Transform) algorithm is one of the most popular algorithms for image matching and has many advantages in comparison to other algorithms [1]. Many different improvements on the SIFT algorithm have been developed and proven to be successful. Ke and Sukthankar proposed a PCA-SIFT which improves local descriptors when using PCA [2]. A.E. Abdel-Hakim and A.A. Farag proposed a Colored SIFT (CSIFT) which builds a novel descriptor in a colorful invariant space [3]. Ma et al. proposed a MI-SIFT which is invariant to mirror images and grayscale-inverted images [4]. Peker proposed a fast image retrieval with the binary quantized SIFT features [5]. Dellinger et al. proposed a SAR-SIFT for the registration of SAR images [6].

At meanwhile, SIFT can be time-consuming for large image stitchings. In most cases, the overlapping area between two images is just 20%-30%, but the SIFT algorithm has to go through the entire image to extract features. Determining the overlapping area in advance can greatly reduce the time of image matching.

The rest of this paper is organized as follows. Section 2 introduces the major steps of the SIFT algorithm. In Section 3, we propose a methodology to roughly determine the overlapping area between two images. Section 4 shows our experiment results and performance evaluation between the original method and the proposed method. Finally, the conclusion is given in Section 5.

2.Background

The SIFT algorithm was first proposed by Lowe in 1999 [1] and then completed by Lowe in 2004 [7]. The SIFT algorithm is a local feature extraction algorithm which is invariant to scaling and orientation, and partially invariant to illumination changes and affine distortion. The four major steps of the SIFT algorithm are described as follows.

2.1 Scale-Space Local Extreme Detection

Keypoint candidates are located at local extrema of Difference of Gaussians (DoG) [1], which is generated due to the difference of the Gaussian pyramid. To build up the Gaussian pyramid, the input image is convolved with Gaussian filters on different scales:

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y); \tag{2.1}$$

where G is the Gaussian function with scale σ, I is the input image and * represents the convolution operation between G and I.

The last convolved image is down-sampled by the factor of 2 and the convolution is repeated on it. This process should be repeated until it cannot be down-sampled further. The convolved images of the same size are grouped by octaves. The DoG pyramid is generated according to the difference of the successive images belonging to the same octave:

$$DoG(x,y,\sigma)=L(x,y,k\sigma)-L(x,y,\sigma); \tag{2.2}$$

The local extrema are detected by comparing each pixel in the DoG pyramid with its 26 neighbours on the current and adjacent scales.

2.2 Keypoint Localization

To eliminate the influence of the edge response and increase the robustness to noise, a Taylor expansion is used for each keypoint candidates:

$$D(X)=D+\frac{\partial D^T}{\partial X^2}X+\frac{1}{2}X^T\frac{\partial^2 D^T}{\partial X^2}X\,; \tag{2.3}$$

where $X=(x,y,\sigma)^T$ represents the displacement from its real position. Set the derivative of $D(X)$ to 0 gives:

$$\hat{X}=-\left(\frac{\partial^2 D}{\partial X^2}\right)^{-1}\frac{\partial D}{\partial X}\,; \tag{2.4}$$

$$D(\hat{X})=D+\frac{1}{2}\frac{\partial D^T}{\partial X}\hat{X}\,; \tag{2.5}$$

The absolute value of $D(\hat{X})$ which is less than a certain threshold will be discarded.

2.3 Orientation Assignment

Each neighbouring pixel is weighted by the gradient magnitude and a Gaussian window with a σ that is 1.5 times of the scale of the keypoint is built for the orientation histogram [8]. The peak is selected as the main orientation and any other orientation greater than 80% of the maximum value is selected as the auxiliary orientation.

2.4 Keypoint Descriptor

To determine the orientation of the keypoints, the gradient magnitude and orientation of each pixel in the neighbourhood are calculated as:

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}\,; \tag{2.6}$$

$$\theta(x,y)=\tan^{-1}((L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2)\,; \tag{2.7}$$

The area around the keypoint is divided into 4*4 grids and within each grid the orientation is computed with an appropriate Gaussian window to establish an 8-bin histogram. Altogether there are 4*4*8=128 elements forming the keypoint descriptor.

3. Proposed Methodology

In the present work, we propose a methodology that roughly determines the overlapping area between two images as the initial step to reduce the searching area for the SIFT algorithm. Not all the features extracted by the SIFT algorithm are effective and those in the non-overlapping area can reduce the accuracy of image matching and increase the processing time.

Pre-selecting the overlapping area can exclude those useless features, thus greatly improving the efficiency and accuracy of the basic SIFT algorithm.

The proposed algorithm works as follows. First, the two input images A and B are divided into girds. For each grid, we can calculate the average greyscale level. Two windows of the same size should be used to select an area in both image A and B, represented by the grey parts in Fig. 1. Then the difference D of the two selected windows is defined as:

$$D = \sum_{x} \sum_{y} \left(M(x,y) - N(x,y) \right)^2 ; \tag{3.1}$$

Where M and N refer to the matrices of greyscale levels of the selected windows from A and B. The exhausted search is used in A and B to find the minimum difference D, and the corresponding windows are used for feature extraction.

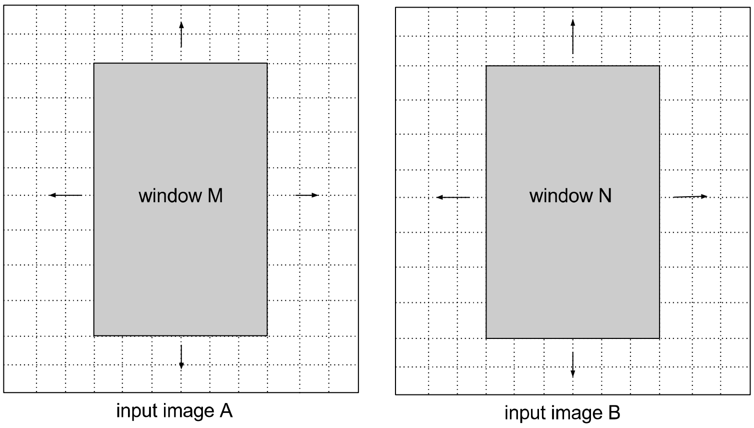

Figure 1: Illustration of the Proposed Method

The elapsed time of this algorithm is relevant to the size of two initial pictures, grid size and window size. The grid size influences the pre-selecting time and the window size affacts the running time of the SIFT algorithm. If the gird size is too small, it will take too much time to find the least difference and large grid size may result in low accuracy. Small window size can reduce the time of SIFT processing, however, it may also lead to insufficient feature points. The performance evaluation of the proposed algorithm is discussed in the next section with experiment results.

4. Results

Our experiment is based on the implementation of SIFT in Matlab where two satellite photos are used as test images shown in Fig. 2. Both images are 600*600 pixels. The original SIFT took 13.342s and found 663 matches.

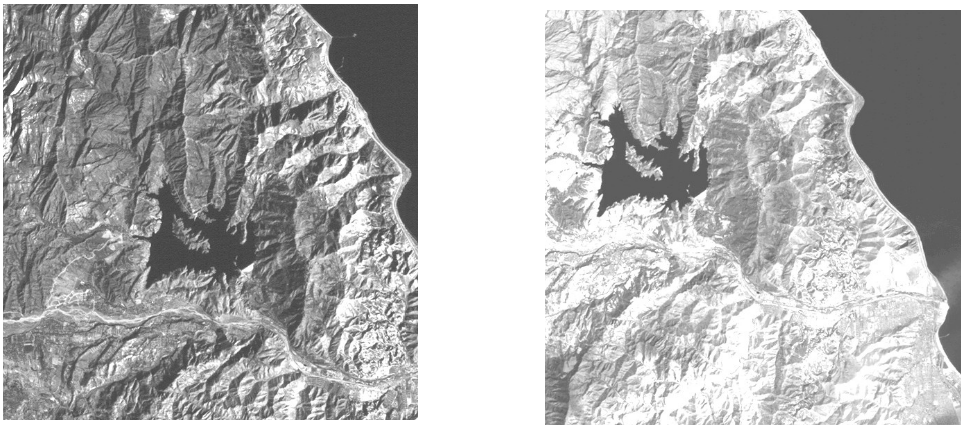

Figure 2 : Two Images Used in the Tests

Using the proposed method with different grid sizes and window sizes, the elapsed time and number of matches are demonstrated in Table 1. The elapsed time includes the pre-processing time and the feature extraction and matching time.

Grid width [pixel]	Window Width [grid]					
	10	12	14	16	18	20
10	25.014s /2	22.060s / 0	19.934s / 3	18.387s / 103	17.796s / 141	14.799s / 187
15	3.336s / 5	2.953s / 56	2.689s / 79	2.567s / 89	2.609s / 283	2.552s / 356
20	1.503s / 92	1.620s / 98	2.005s / 263	2.258s / 387	3.561s / 490	5.253s / 515
25	1.610s / 238	1.949s / 337	3.172s / 480	4.281s / 521	6.042s / 616	8.146s / 636
30	2.045s / 335	3.502s / 490	5.412s / 551	8.124s / 625	13.519s / 573	14.301s / 663

Table 1 : Elapsed Time and Number of Matches Using the Proposed Method

The performance index is defined as the ratio of number of matches to the elapsed time:

$$Performance = \frac{Number\ of\ matches}{Elapsed\ time} \tag{4.1}$$

Using this criterion, the index of the original SIFT method is 49.69 and the best result of the proposed method is 172.94 when the grid width is 25 pixels and each window takes up 12*12 grids. The performance surface is shown in Fig. 3.

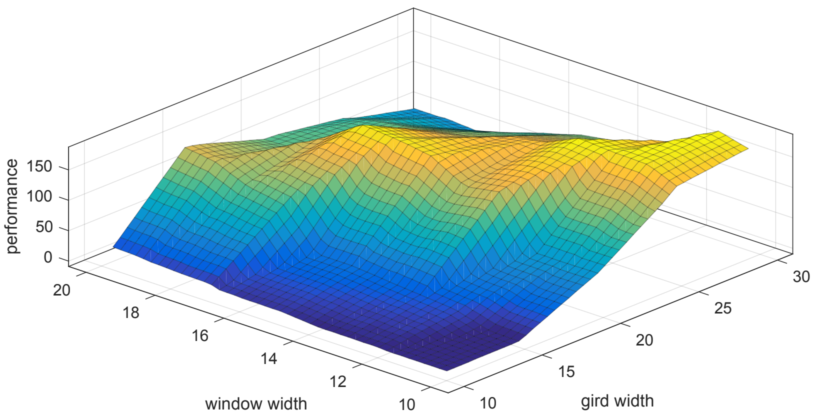

Figure 3: Performance of the Proposed Method

Comparing these two algorithms, we reach the conclusion that the proposed method performs better than the original method when choosing the appropriate gird size and window size. When the window size is the same as the input image size, the performances of the proposed method and the original SIFT method are almost the same. Generally speaking, the grid width of 20-30 pixels and the window taking up 20%-30% of the original image area are recommended for both efficiency and accuracy of image matching.

5. Conclusion

In view of the long processing time of SIFT, this paper proposed a fast image matching algorithm, which pre-selects the overlapping area to speed up the process of the SIFT algorithm. On the occasion that requires fast image stitching while photos are of the same scale and direction, the algorithm proposed in this paper can perform better than the SIFT algorithm, increasing the efficiency without affecting accuracy.

References

[1]D.G. Lowe. *Object recognition from local scale-invariant features*[C]. Proceedings of the IEEE International Conference on Computer Vision. IEEE, New Jersey, United States. Vol. 2, pp. 1150-1157(1999)

[2]Y. Ke and R. Sukthankar. *PCA-SIFT: A more distinctive representation for local image descriptors*[C] Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference. IEEE Computer Society, New Jersey, United States. Vol. 2, pp. II-II(2004)

[3]A.E. Abdel-Hakim and A.A. Farag. *CSIFT: A SIFT Descriptor with Color Invariant Characteristics*[c] Computer Vision and Pattern Recognition, 2006. CVPR 2006. Proceedings of the 2006 IEEE Computer Society Conference. IEEE Computer Society, New Jersey, United States. Vol. 2, pp. 1978-1983(2006)

[4]R. Ma, J. Chen and Z. Su. *MI-SIFT: mirror and inversion invariant generalization for SIFT descriptor*[C]. Proceedings of the ACM International Conference on Image and Video Retrieval. Association for Computing Machinery, New York, United States. pp. 228-235(2010)

[5]K.A. Peker. *Binary sift: Fast image retrieval using binary quantized sift features*[C]. Content-Based Multimedia Indexing (CBMI). IEEE Computer Society, New Jersey, United States. pp. 217-222(2011)

[6]F. Dellinger, J. Delon, Y. Gousseau, J. Michel and F. Tupin. *SAR-SIFT: a SIFT-like algorithm for SAR images*[J]. IEEE Transactions on Geoscience and Remote Sensing, 53(1): 453-466(2015)

[7]D.G. Lowe. *Distinctive image features from scale-invariant keypoints*[J]. International Journal of Computer Vision. 60(2): 91-110(2004)

[8]W.T. William and M. Roth. *Orientation histograms for hand gesture recognition*[C]. International Workshop on Automatic Face and Gesture Recognition. IEEE Computer Society, New Jersey, United States. Vol. 12, pp. 296-301(1995)