World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:11, No:8, 2017

K-Means Based Matching Algorithm for Multi-Resolution Feature Descriptors

Shao-Tzu Huang, Chen-Chien Hsu, Wei-Yen Wang

***Abstract*—**Matching high dimensional features between images is computationally expensive for exhaustive search approaches in computer vision. Although the dimension of the feature can be degraded by simplifying the prior knowledge of homography, matching accuracy may degrade as a tradeoff. In this paper, we present a feature matching method based on k-means algorithm that reduces the matching cost and matches the features between images instead of using a simplified geometric assumption. Experimental results show that the proposed method outperforms the previous linear exhaustive search approaches in terms of the inlier ratio of matched pairs.

***Keywords*—**Feature matching, k-means clustering, scale invariant feature transform, linear exhaustive search.

I. Introduction

FEATURE matching is widely used in computer vision application such as object recognition, image registration, and virtual reality. In general, the task of matching feature points between two images contains three essential steps: interesting point detection, feature description, and feature matching. First, interesting points are selected at the specific location in the image. Next, each of the interesting points is represented by a feature vector, namely, feature descriptor, which describes the geometric properties of the point with strong resistance to the transformation of homography matrix. Finally, features between two images are matched by comparing the differences between each of the feature vectors. There are several well-known features detection methods [1], such as Scale Invariant Feature Transform (SIFT) [2], Speeded-Up Robust Feature (SURF) [3], and Principal Components Analysis (PCA-SIFT) [4]. In practice, the dimension of feature vector depends on the amount of information that it used to describe the interest points distinctively. In the other words, feature vectors have high dimensionality for the descriptor design purpose of accuracy and robustness. Hence, the task of feature matching is to identify the most similar matches between different high-dimensional vectors which are the most computationally expensive part.

Given that exhaustive search is an intuitive and well-known approach for feature matching, the main challenge is that it requires a great amount of computation cost for high dimension feature matching. For example, if we have to match k dimension feature vectors of SIFT descriptors between two images, the computation cost increases as the order of $O(kMN)$, where M and N are the number of the features on each image. Moreover, the overfitting problem may seriously cause the matching inaccuracy that requires an iterative filtering process to handle. Therefore, it is not applicable to real-time tasks which are essential for modern computer vision application.

The present feature matching methods can be separated into two categories: distance metric and space partition. The former estimates the distance in feature vector space, and then, each of the two features in the different images which contain the shortest distance is formed as a feature pair. The exhaustive search method is an example and is improved by a considerable amount of literature for decades. Song et al. [5], [6] speed up the searching process by using a norm-sorted database. However, in real-time applications, the input data are non-sorted that need an additional computation cost of the sorting process before the use. Tsai et al. [7] apply Multi-resolution Candidate Elimination (MRCE) technique to deal with such problem by simply the prior knowledge of geometric transform between the two images, and hence degrade the matching accuracy as a tradeoff. The space partition method creates a tree structure to efficiently speed up the search of the nearest neighbors. Using KD-tree for matching [8], is a typical example of the space partition method. Silpa-Anan and Hartley [9] use multiple KD-trees from the same data set to improve the search performance. Muja and Lowe [10] present the priority search k-means tree algorithm to approximate nearest neighbors. However, all of the space partition approaches require additional memory for storing the tree structures. Furthermore, the size of memory will set limit on the matching performance which is not desirable.

According to the pros and cons of the feature matching approaches, we are motivated to develop a feature matching method that decreases the amount of redundant matching of exhaustive search approaches. Inspired by [7], our design reduces a great amount of redundant matching and hence overall degrades the computation cost of linear exhaustive search (LES) approaches. Furthermore, our matching method is designed without using a simplifying the prior knowledge of homography between the different images. Therefore, we leverage the applicability and the overall matching quality of LES approaches.

Shao-Tzu Huang is with the National Taiwan Normal University, Graduated Institution of Electrical Engineering Taipei, Taiwan (e-mail: glnhwng@hotmail.com).

Chen-Chien Hsu and Wei-Yen Wang are with the National Taiwan Normal University, Department of Electrical Engineering Taipei, Taiwan (e-mail: jhsu@ntnu.edu.tw, wywang@ntnu.edu.tw).

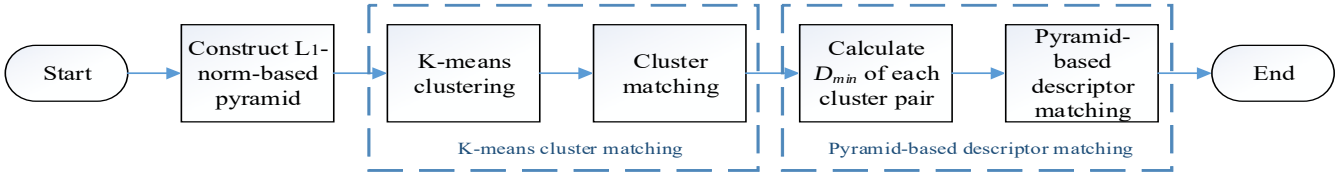

Fig. 1 The flow chart of proposed method

II. Method

The proposed method reduces the computation cost of feature matching by using *k*-means clustering. It mainly contains three parts: 1) construction of L_1 norm based pyramid, 2) k-means cluster matching, and 3) pyramid-based descriptor matching. The details of each step are demonstrated in Fig. 1 and explored in this section.

A. Construction of L_1- Norm-Based Pyramid

As described in the previous section, we target on reducing the computational cost of feature matching. Under the same scene, two images I_1 and I_2 are captured at different camera positions. We utilize an existing algorithm to detect interest points in the image and represent each of the points with 2^l dimension vector. Inspired by [5]-[7], a pyramid structure is constructed on the feature vector domain for each interest point in the two images. Each layer of the pyramid contains one dimension information. From bottom to top of the pyramid, the element of layer *i* is computed by

$$u_k^{(i)} = u_{2k-1}^{(i+1)} + u_{2k}^{(i+1)},\ 1 \le k \le 2^i \tag{1}$$

where $u_k^{(i)}$ is the *k*-th element of layer *i*. In the other words, there is only one element in the top layer of the pyramid. The whole pyramid is constructed without sorting process because of time-consuming concern.

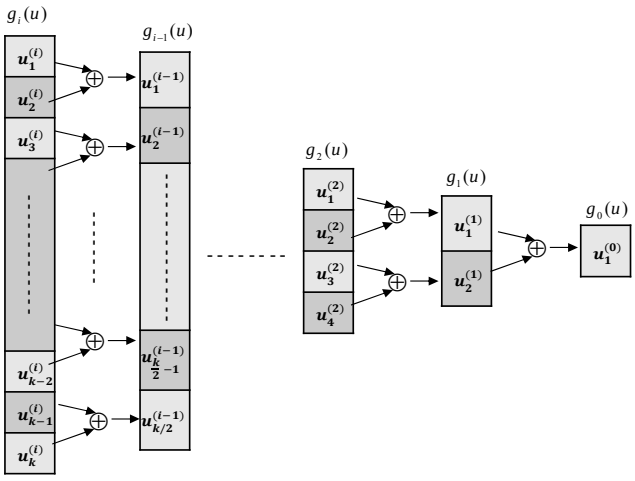

Fig. 2 L_1-norm-based pyramid structure

B. k-Means Cluster Matching

To reduce the computation cost of feature matching, the feature points in the image are clustered by using *k*-means algorithm [11]-[13]. That is, given the position vector *x* of each feature point, we have to solve the objective function

$$\arg\min_{S} \sum_{j=1}^{k} \sum_{x \in S_j} \left\| x - c_j \right\|^2 \tag{2}$$

where c_j is the position vector of the center point of *j*-th cluster S_j. The center point c_i is stored in the cluster matching process between the two images. We compute the Euclidian distance, and hence each of the two closest clusters at different image forms a cluster pair. The number of cluster pairs depends on the parameter *k*.

C. Pyramid-Based Descriptor Matching

The final step of the proposed method is a matching process between each two feature-based pyramid in different images. As mentioned in the previous section, our method seeks to reduce the amount of redundant feature matching. In practice, only the features in the same cluster pair are used for the match. For each cluster pair S_{j1} and S_{j2} on image I_1 and I_2 respectively, we estimate the average Manhattan distance of the cluster pairs

$$D_{\min} = \frac{1}{MN} \sum_{\forall g_0(u) \in S_{j1}, \forall \hat{g}_0(u) \in S_{j2}} \left| g_0(u) - \hat{g}_0(u) \right| \tag{3}$$

where *M* and *N* are the number of feature points in cluster S_{j1} and S_{j2}, respectively. After that, the matching process begins from the top layers of each two pyramids.

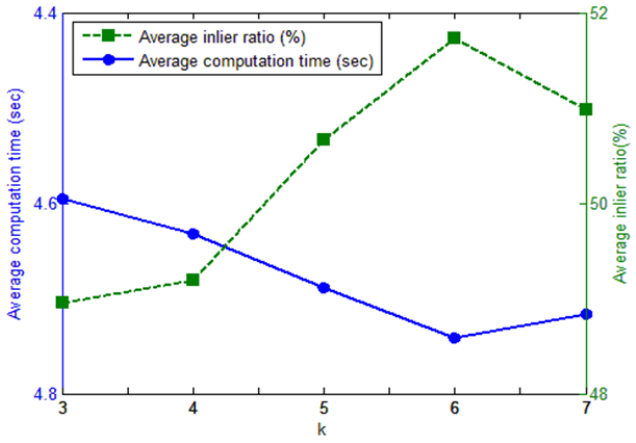

Fig. 3 The matching performance using various parameter k of k-means clustering. The blue line represents the average computation time. The green dashed line represents the average inlier ratio of matched pairs

If the Manhattan distance between the elements of the top layer is smaller than D_{min}, we will repeat the same matching

process in the next layer of two pyramids until reaching the bottom layer of the pyramid. Otherwise, we consider the two features as unrelated feature points. Therefore, any matched feature pairs have to satisfy the condition that each of the Manhattan distance between the elements in the same layer of pyramids is smaller than D_{min}.

III. Experiment

An experiment is conducted to evaluate our method. The input of our simulation tests is synthetic scenes from the benchmark of Nister and Stewenius [14] and Mikolajczyk amd Schmid [15]. We utilize SIFT algorithm [2], [8] to locate the feature points in different images and represent each of them with a 128-dimensional vector. Then, we apply our feature matching method to obtain matched feature pairs between different images. To evaluate the matching performance, the matched feature pairs are classified into inlier and outlier pairs according to the homography matrix [1]. In practice, we input the position data of matched feature pairs to the OpenCV *findhomography* function with Random Sample Consensus (RANSAC) method to handle the classification. Then, we compute the inlier ratio by

$$R = W_{inlier} / W_{all} \quad (4)$$

where W_{inlier} and W_{all} are the number of inliers and the total feature pairs, respectively. Due to the random property of *k*-means algorithm, we run the method 10 times and compute the average inlier ratio. The computation time of the matching process is also considered in our test.

TABLE I
Performance Comparison of 600 Scenes

Matching method					
LES		**MRCE [7]**		**Proposed method**	
Average Inlier ratio	**Average Time (sec.)**	**Average Inlier ratio**	**Average Time (sec.)**	**Average Inlier ratio**	**Average Time (sec.)**
0.499	59.311	0.486	3.430	0.509	3.549

TABLE II
Performance Comparison of Five Selected Scenes

Test scene	Matching method					
	LES		**MRCE [7]**		**Proposed method**	
	Inlier ratio	**Time (sec.)**	**Inlier ratio**	**Time (sec.)**	**Inlier ratio**	**Time (sec.)**
Fig. 4 (a)	0.848	53.568	0.900	3.910	**0.911**	**3.010**
Fig. 4 (b)	0.702	6.016	0.676	**1.271**	**0.715**	1.299
Fig. 4 (c)	0.556	45.773	0.778	**3.339**	**0.800**	3.344
Fig. 4 (d)	**0.611**	36.650	0.433	**2.636**	0.512	2.642
Fig. 4 (e)	**0.736**	98.594	0.614	4.766	0.624	**4.542**

To determine the parameter *k* of k-means clustering, we repeat the experiment with various *k* setting using 600 test scenes from the benchmark dataset. The experiment results are shown in Fig. 3. We can see that the computation time and the average inlier ratio of matched pairs increase as *k* increases from 3 to 6. To achieve the balance of joint matching performance, *k* should locate at the point before the crossing section of the two lines in Fig. 3. Therefore, we suggest setting *k* to be 4 for a desirable joint matching performance of both computation time and average inlier ratio of matched pairs.

We compare the matching performance of our method with LES and MRCE [7] with another 600 test scenes. The comparison is shown in Table I. We can see that our method outperforms LES and MRCE in the comparison of the average inlier ratio of matched pairs. Although the computation time of this method is slower than MRCE, the average ratio of inlier feature pairs is higher than that of MRCE. This is because our k-means clustering process suppresses the impact of D_{min} inaccurate estimate.

As described in the previous section, the computation of D_{min} is required for the matching process. The physical meaning of D_{min} is a generalized distance of the geometric transform between different images. If wrong matching feature pairs exist in the computation, which are extreme values of the input, estimation error of D_{min} occurs due to the averaging operator. MRCE does not contain clustering process before the D_{min} computation. Hence, it simplifies the prior knowledge of D_{min} as a fixed distance of the geometric transform for every matched pair in that MRCE suffers from the inaccurate D_{min} problem. This causes a serious impact when the two matching images are captured in the same scene but with different illuminations or blurriness. To elaborate, we empirically select five typical scenes from the 600 test scenes as shown in Fig. 4. The first two image pairs are images captured in different camera positions, and the third pair is in different zoom settings. The fourth and fifth image pairs are the same images with different blurriness and illumination, respectively. The experimental results of the five scenes are shown in Table II. In Table II, we can see that the inlier ratio of MRCE is the lowest in the comparison of test scene Figs. 4 (d) and (e) due to inaccurate D_{min} problem.

On the contrary, the *k*-means clustering process of our method reduces some sort of wrong feature matches from the computation and hence suppresses the impact of wrong matching pairs. In this circumstance, the accuracy of the D_{min} estimate will be improved. As a result, the introduced method will effectively reduce the redundant matching and it improves the matching accuracy.

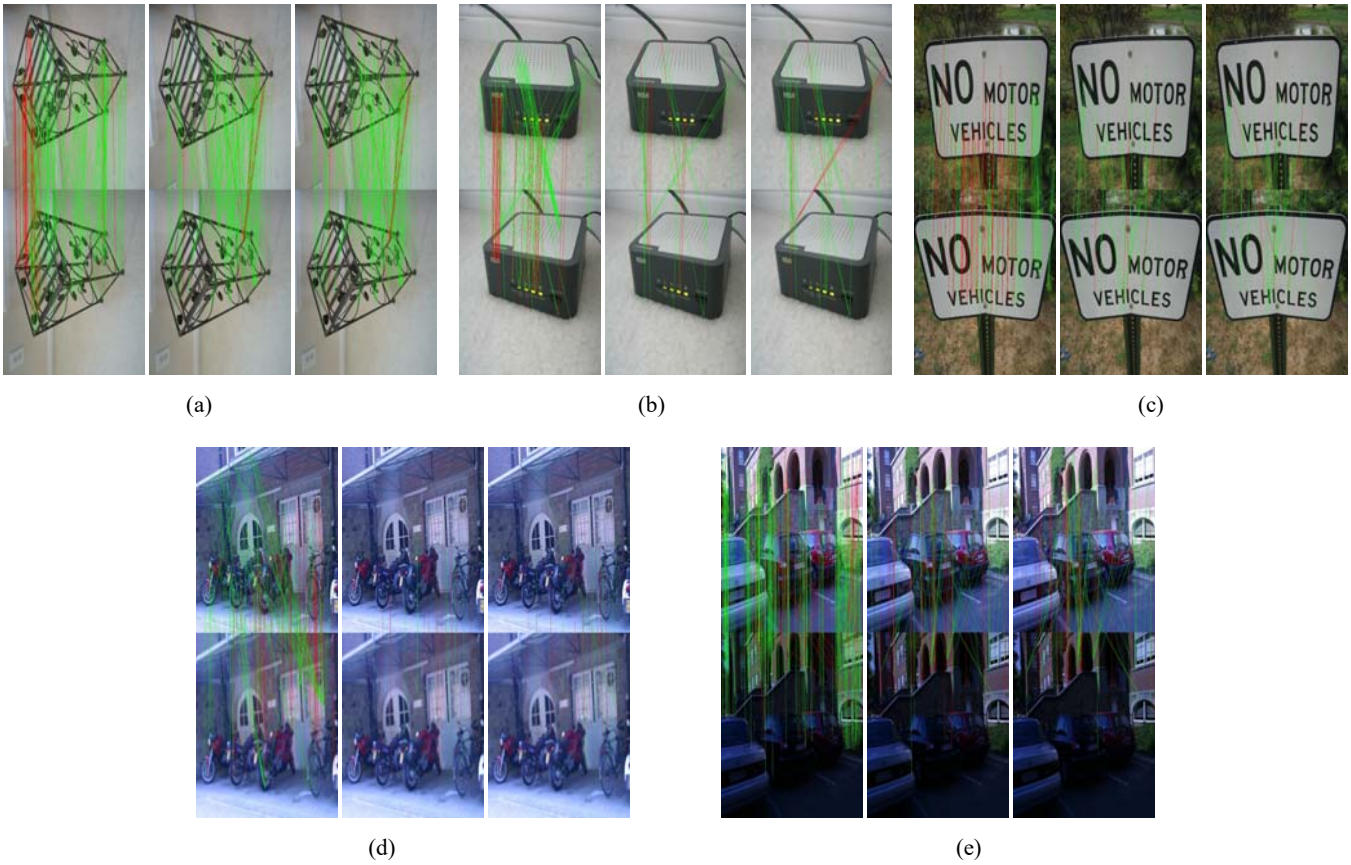

Fig. 4 The comparison of matched feature pairs between LES (left), and MRCE [7] (middle), and the proposed method (right). The green lines and red line represents inlier and outlier feature pairs, respectively. (a) and (b) are the affine transformed images, (c) are the scale changed images, (d) are the blurred images, and (e) are the illumination changed images

IV. Conclusion

We have described a feature matching method based on combining L_1-norm based pyramid and *k*-means clustering. Our method prevails over the conventional brute force method and previous linear exhaustive approaches in the average ratio of inlier matched feature pairs. The method lays the foundation of the robust feature matching without using a simplified prior knowledge of homography and additional memory needs.

Acknowledgment

This research is partially supported by the "Center of Learning Technology for Chinese" and "Aim for the Top University Project" of National Taiwan Normal University (NTNU), sponsored by the Ministry of Education, Taiwan, R.O.C. and Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 105-2221-E-003 -010.

References

[1] L. Juan and O. Gwun. "A Comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing*, Vol. 65, pp. 143-152, 2009.

[2] D. G. Lowe. "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, No. 2, pp. 91-110, 2004.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, No. 3, pp. 346-359, 2008.

[4] Y. Ke and R. Sukthankar. "PCA-SIFT a more distinctive representation for local image descriptors," *Proc. International Conference on Computer Vision and Pattern Recognition*, 2004, Vol. 2, pp. 506-513.

[5] B. C. Song and J. B. Ra, "Multiresolution descriptor matching algorithm for fast exhaustive search in norm-sorted databases," *Journal of Electronic Imaging*, vol. *14, No.*4, pp. 043019-043019, 2005.

[6] B. C. Song, M. J. Kim, and J. B. Ra." A fast multiresolution feature matching algorithm for exhaustive search in large image databases," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. *11*, No. 5, pp. 673-678, 2001.

[7] C.-Y. Tsai, A.-H. Tsao, and C.-W. Wang. "Real-time feature descriptor matching via a multi-resolution exhaustive search method," *Journal of Software,* vol. 8, no. 9, pp. 2197-2201, 2013.

[8] D. G. Lowe. "Object recognition from local scale-invariant features," *In Computer vision, The proceedings of the seventh IEEE international conference on IEEE*, 1999, Vol. 2, pp. 1150-1157.

[9] C. Silpa-Anan and R. Hartley. "Optimised KD-trees for fast image descriptor matching," In *Proc. Computer Vision and Pattern Recognition, CVPR 2008. IEEE Conference on IEEE*, June, 2008, pp. 1-8.

[10] M. Muja and D. G. Lowe. "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, No.11, pp. 2227-2240, 2014.

[11] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm." *Journal of the Royal Statistical Society. Series C (Applied Statistics),* Vol. 28, No. 1, pp. 100-108, 1979.

[12] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. "Constrained k-means clustering with background knowledge," *In ICML*, June, 2001, Vol. 1, pp. 577-584.

[13] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, and P. Fränti. "Improving k-means by outlier removal," *In Scandinavian Conference on Image Analysis*. Springer Berlin Heidelberg, 2005, pp.

978-987.
[14] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” *Proc. International Conference on Computer Vision and Pattern Recognition*, 2006, Vol. 2, pp.2161-2168.
[15] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol27, No.10, pp. 1615-1630, 2005.