

A novel method for SIFT features matching based on feature dimension matching degree

Yao Yang^{1,*}, Jinkang Wei¹, Ximing Zhan¹, and Xikui Miao²

¹School of Astronautics Northwestern Polytechnical University Xi'an ,shannxi, China

²Key Laboratory of Electro-Optical Countermeasures Test & Evaluation Technology, LuoYang, China

Abstract. We proposes a method for fast matching SIFT feature points based on SIFT feature descriptor vector element matching. First, we discretize each dimensional feature element into an array address based on a fixed threshold value and store the corresponding feature point labels in an address. If the same dimensional feature element of the descriptor vector has the same discrete value, their feature point labels may fall into the same address. Secondly, we search the mapping address of the feature descriptor vector element to obtain the matching state of the corresponding dimensions of the feature descriptor vector, thus obtaining the number of dimensions matching between feature points and feature dimension matching degree. Then we use the feature dimension matching degree to obtain the suspect matching feature points. Finally we use the Euclidean distance to eliminate the mismatching feature points to obtain accurate matching feature point pairs. The method is essentially a high-dimensional feature vector matching method based on local feature vector element matching. Experimental results show that the new algorithm can guarantee the number of matching SIFT feature points and their matching accuracy and that its running time is similar to that of HKMT, RKDT and LSH algorithms

1 Introduction

Image feature matching is essential and crucial for computer vision, and can be employed in image retrieval[1], image stitching[2], visual reality[3], and scene reconstruction[4-6]. Firstly, local features are extracted from each image in these applications, and then the local descriptor with invariance is computed. By applying the nearest neighbor search of features, the most similar feature can be found to achieve the image matching. SIFT[7] is known as the most representative feature extraction and matching method currently, and it had been improved by many researchers[8-10]. The feature-points extracted from SIFT descriptors are extremely robust, it is adapt to illumination change and deformation. Thus, it is still the most popular local descriptors to solve the image matching problems. However, the SIFT descriptors is high-dimension, namely 128. It is difficult to find a way to deal with the computational complexity during the matching process. The proposed method is a high-dimensional feature vector matching method based on the partial matching of feature vector.

* Corresponding author: yangyao61@163.com

The experimental results show that the performance of our proposed method is similar to that of HKMT, RKDT and LSH.

2 Related work

SIFT operators are considered to be highly robust, but their descriptors have 128 dimensions. The existing algorithms use the exhaustive method as the search matching algorithm. The exhaustive search algorithm can ensure that the correct matching point pairs are searched out, however, its time cost is high; therefore many approximately nearest neighbor search algorithm comes into being. These algorithms can not only ensure the search precision to a certain degree but also save much time compared with the exhaustive search algorithm. Generally speaking, these algorithms can be divided into two types: those based on tree indexing method and those based on hashing indexing method. The KD tree search method[11] is a typical search matching algorithm based on trees but may fail when its number of dimensions is more than 15[12] or may even be inferior to the linear exhaustive search algorithm. The best-bin-first (BBF) algorithm [12] is the extension of the KD tree and breaks through the restriction of the number of dimensions and can search well high-dimensional features. On this basis, Hartley et al proposed the RKD-tree search algorithm[13] and improved the traditional BBF search algorithm. With GPU, Hu et al achieved the parallel processing of the RKD-tree[14], thus greatly enhancing the matching speed. But experimental data show that[15] its correct matching rate is less than that of the traditional SIFT image feature point matching algorithm. The IDistance[16] is another type of tree-based indexing algorithm, uses the K-means clustering method to analyze data distribution and then indexes the B+- tree. This reduces the search scope to a certain degree and increases the search speed. However, because the algorithm takes some clustering time to establish its index structure, it is not effective for the applications based on image feature matching. Different from the indexing method based on tree structure, the locality sensitive hashing (LSH) algorithm[17] is another type of approximate nearest neighbor search matching algorithm based on the hashing indexing. The method is: first, the data points are projected to the Hamming space and the indexes are established with a set of hash functions that satisfy certain constraint conditions, thus obtaining the nearest neighbor of an enquiry point with a rather big probability. To solve the problem that the algorithms based on LSH have a rather big space cost, scholars improved many LSH algorithms[18-19], thus ensuring the accuracy rate and search speed and optimizing the space cost. Ref. [20] applied the LSH algorithm to retrieving SIFT descriptors, thus reducing search time and obtaining satisfactory results.

Starting from the matching of each dimensional element of a feature point, this paper relies on the dimensional matching degree of a feature point pair to screen the suspicious matching point pair and then purifies it with the Euclidean distance, thereby obtaining the ultimate matching pair. Compared with the Euclidean distance exhaustive algorithm, the KD tree algorithm and the LSH algorithm, through discretizing the feature address mapping, our algorithm can directly match each dimensional element, obtain the dimensional matching degree among the feature points, screen out a large number of unmatched points with the dimensional matching degree, reduce the search scope, guarantee the accuracy rate and achieve fast search.

3 The feature dimension matching algorithm(FDMA)

In matching two images, the 128-dimensional SIFT feature descriptor point sets $\{P_i\}, i = 1, 2, \dots, n$ and $\{Q_j\}, j = 1, 2, \dots, m$ of the two images are obtained with the SIFT

algorithm. The feature descriptor point set $\{P_i\}$ contain n number of descriptor points; the feature descriptor point set $\{Q_j\}$ contain m number of descriptors. We employ $\{P_i\}$ as the benchmark set and $\{Q_j\}$ as the target set to match the feature points of the two images. The feature point matching method proposed by Lowe uses the exhaustive search method, with which the Euclidean distance of each SIFT feature point in $\{P_i\}$ and each SIFT feature point in $\{Q_j\}$ is calculated. The feature point pair that satisfies the Euclidean distance threshold requirements and is purified with the secondary nearest neighbor is the matching pair. When $\{Q_j\}$ is a matching pair which satisfies the Euclidean distance threshold requirements, the SIFT feature point P_i, Q_j satisfies the following relationship:

$$\sqrt{\sum_{k=1}^{k=128} (P_i(k) - Q_j(k))^2} \leq \tau \tag{1}$$

where τ donates the set threshold value; $P_i(k)$ is the k -dimensional element of the i number of descriptors in the $\{P_i\}$ benchmark set; $Q_j(k)$ is the k -dimensional element of the j number of descriptors in the $\{Q_j\}$ target set. Then for all the descriptors, the k -dimensional element has the following function:

$$\max |P_i(k) - Q_j(k)| = \sigma \leq \tau \tag{2}$$

Using $\min(P_i(k))$ as the starting point, we discretize each-dimensional element according to the scale σ :

$$N = \left\lfloor \frac{P_i(k) - \min(P_i(k))}{\sigma} \right\rfloor, N \in Z \tag{3}$$

The k -dimensional feature discretized coordinate axis is formed and shown in Fig. 1.

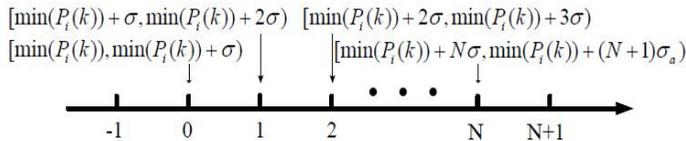


Fig. 1. The k -dimensional feature discretized coordinates.

According to Equation. (2), the $\langle P_i, Q_j \rangle$ matching pair meets the following condition.

If

$$P_i(k) \in [\min(P_i(k)) + N\sigma, \min(P_i(k)) + (N + 1)\sigma] \tag{4}$$

Then

$$Q_j(k) \in [\min(P_i(k)) + (N - 1)\sigma, \min(P_i(k)) + (N + 2)\sigma] \tag{5}$$

In other words, on the k -dimensional feature discretized coordinates axis, the mapping of $P_i(k)$ and $Q_j(k)$ is certainly adjacent or equal:

If $P_i(k) \rightarrow N$

Then $Q_j(k) \rightarrow N - 1$ or $Q_j(k) \rightarrow N$ or $Q_j(k) \rightarrow N + 1$

If extended to other 128-dimensional feature discretized coordinates axis, the $\langle P_i, Q_j \rangle$ matching pair is shown in Fig.2.

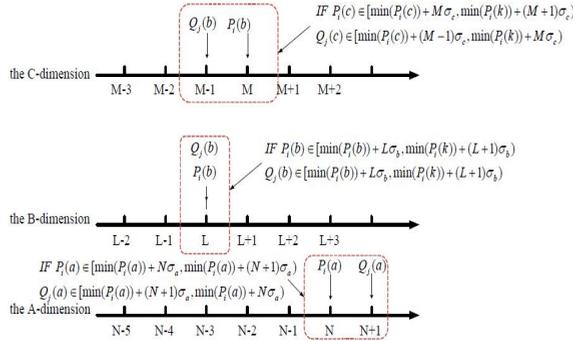


Fig. 2. The distribution of the i number of feature points in the benchmark set and the j number of feature points in the target set along each-dimensional feature discretised coordinate axis.

The above analysis draws the following conclusion: For the 128-dimensional SIFT feature descriptor point sets $\{P_i\}, i = 1, 2, \dots, n$ and $\{Q_j\}, j = 1, 2, \dots, m$, if the discrete feature obtained after every dimensional feature of one pair of feature points P_i and Q_j is discretized with a given threshold σ is adjacent or equal, then $\langle P_i, Q_j \rangle$ is the matching pair whose every dimensional distance is smaller than σ and whose Euclidean distance is smaller than 128σ .

Therefore, after using the threshold σ to discretize the feature coordinate axis, each-dimensional feature matching descriptor automatically obtains its adjacent or equal discrete values. In actual application, we transform into unit cell array address the discrete coordinate of the each-dimensional element of feature descriptor in $\{P_i\}$ and label the corresponding feature points stored and mapped according to the array address, thus transforming the each-dimensional feature descriptor in $\{P_i\}$ into their array address and obtaining the unit cell array T_p , which contains $(N + 3) \times n$ number of cells. $T_p(i, j)$ is the (i, j) number of cells in T_p ; the elements in the cell denote the labels of all the feature points in the i -th discrete intervals that correspond to the feature of this dimension which the j -dimensional feature satisfies. If no feature points are satisfied, then the element is \emptyset label. For example, the unit cell array element $T_p(N, j) = \{a, b, c, \dots\}$ denotes that all the j -dimensional features of the feature points a, b, c, \dots in $\{P_i\}$ fall within the N -th discrete interval at the discrete coordinates of the j -dimensional feature. The diagram for creating the j -dimensional unit cell array is shown in Fig. 3:

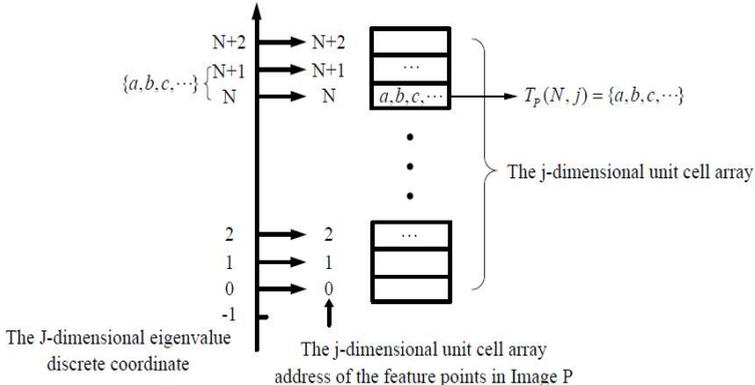


Fig. 3. The diagram for creating the j -dimensional unit cell array.

We can calculate the elements $Q_j(k)$ of each dimension through the feature-points of $\{Q_j\}$ by Eq. (6) to obtain the corresponding cell array address,

$$\begin{aligned}
 & \text{if } n_{Q_j(k)} \leq 0, n_{Q_j(k)} = 0; \\
 & \text{if } n_{Q_j(k)} \geq M + 3, n_{Q_j(k)} = M + 3; \\
 & \text{else} \\
 & n_{Q_j(k)} = \frac{T_2(n, k) - \min(T_1(i, k))}{\sigma_k} + 2
 \end{aligned} \tag{6}$$

With the direct query of $T_p(ad, k), T_p(ad + 1, k), T_p(ad - 1, k)$ elements, the labels of the extracted correspondent feature points are the feature points that match P in the K-dimension and Q that has the N-th feature-points in the K-dimension. The feature dimension matching is achieved by the ratio between number of matching dimensions and total number of dimensions using the feature points in $\{P_i\}$ and $\{Q_j\}$, $D = I / L, 0 \leq I \leq L$, where I represents the number of matching dimensions; L represents the total number of feature dimensions; D represents the feature dimension matching degree. Because the different dimensional values of the feature points with a certain invariance response differently in different variations, we assume that the matching feature points may not necessarily satisfy the requirements for all-dimensional matching; namely their feature dimension matching should be $D < 1$. Therefore, the feature dimension matching degree threshold must be introduced into practical applications. After calculating the feature dimension matching degree between P_i and Q_j , it should be compared with the given feature dimension matching threshold. If it is larger than the threshold, it is assumed to be mutually matching. The feature dimension matching threshold that has SIFT features generally assumes to be around 0.7; the larger the value, the fewer the matching points. The smaller the value, the more matching errors.

The above operations can achieve the suspect matching point pair sets between all the feature points in P and all the feature points in Q, thus establish the suspect feature point matching matrix $T_{match} \in \mathbb{N}^{2 \times c}$, where the j column of the matrix denotes the alternative feature point matching pair in P and Q.

Table 1. Algorithm process and steps.

1. Input two images P and Q
2. Using SIFT algorithm to calculate the SIFT feature description sets of two images: benchmark set $\{P_i\}$ and target set $\{Q_j\}$
3. Use $\{P_i\}$ to construct the 128-dimensional feature discrete coordinates
4. Transform the feature discrete coordinate of $\{P_i\}$ into unit cell array and use the feature point label inside P as the array element value of the corresponding address
5. Use $\{Q_j\}$ to calculate the unit cell array address that corresponds to the each-dimensional value of feature points in Q
6. Extract the feature point labels in the array according to the address in 5, Obtain the dimension matching degree of each point, and remove the feature points that do not satisfy the dimension matching degree threshold.
7. Purification by Euclidean distance to eliminate mismatches and achieve one-for-one matching
8. Delete the farthest Delete-counts pairs of feature point
9. Output the matching point pairs

However, the analysis and experiments show that the matching point pairs do not have one-for-one correspondence. One point in P may correspond to several points in Q or several points in P may correspond to one point in Q . To solve this problem, we calculate the Euclidian distance of the repeatedly matching feature point pair and use the nearest point as the matching point, ultimately obtaining the one-for-one correspondent feature point pair, whose number is c' .

Finally, to purify the experimental results on the algorithm, we introduce the ultimate screening threshold $DeleteCounts = floor(c'/beta)$ and use $beta$ as the screening threshold to calculate the Euclidian distance in the same way, rank it and delete the $DeleteCounts$ feature point pair which is the farthest away from the Euclidean distance and the least matching. Algorithm process and steps are as shown in Table1.

4 Experimental results and analysis

We use the Visual Studio 2013 to implement the novel feature point matching method proposed in this paper and calculate the matching time and feature points. The configuration of the computer is Intel (R) Core (TM) i5-3210M CPU @ 2.50Hz, 8GB memory.

We first perform the scale variation, rotation variation, noise variation, illumination variation and deformation variation of the Lena images to analyze the matching accuracy and speed, and then use the standard image dataset [21] for evaluation.

The FDAM algorithm uses the following parameters: $N = 20$, $\alpha = 0.7$, $beta = 3$, where N is the number of effective discrete segments; α is the feature dimension matching degree threshold; $beta$ is the screening threshold. The threshold value of the exhaustive algorithm for comparison is 0.2.

In order to evaluate the effectiveness of the proposed FDAM algorithm, we compare its performances with those of the three ANN algorithms: the hierarchical k-means tree (HKMT) algorithm, the random KD tree (RKDT) algorithm and the local sensitive hashing (LSH) algorithm and the exhaustive algorithm. We use the algorithms in the FLANN library[22] in the OpenCV3.0 to implement the three algorithms.

4.1 Evaluation of artificial transformation test

This paper carries out the artificial transformation of the Lena images (512*512, the number of SIFT feature points is 1998). In this case, we can know the exact position of the feature points, so as to better analyze the matching results. The experimental results are shown in Fig 4. In Fig. 4(a), salt and pepper noise is added to the Lena image, and the number of SIFT feature points is 1931. In Fig. 4(b), the ratio of areas of the two matched images after scale variation is 1:0.25, and the number of SIFT feature points is 1192. In Fig 4(c), the illumination variation of the Lena image is conducted, and the number of SIFT feature points is 2256. In Fig. 4(d), the Lena image is rotated 45 degrees clockwise, and the number of SIFT feature points is 2927. In Fig. 4(e), the Lena image is deformed, and its upper edge varies into the original 3/4; the number of SIFT feature points is 2105. Fig 4 shows the matching results of HKMT, RKDT, LSH, FDMA and Exhaust algorithm from left to right.

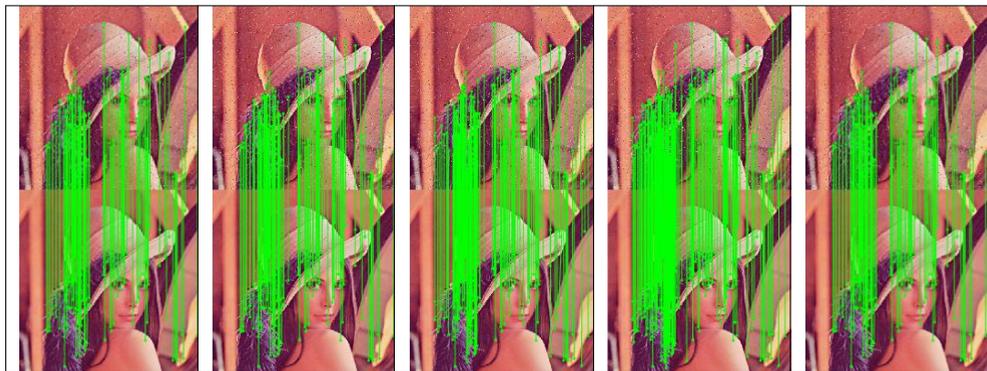
The running time of the algorithm and its logarithm pairs of matching feature points are obtained, as given in Tables 2 and 3, where the running time only includes that of feature point matching but does not include that of feature point detection.

Table 2. Match point number comparison result.

	HKDT	RKDT	LSH	Exhaust	FDMA
Noise	129	122	152	204	121
Scale	175	172	143	137	171
Illumination	153	143	228	376	139
Rotational	484	467	610	548	463
Deformation	69	51	226	359	49

Table 3. Match time comparison table (unit s).

	HKDT	RKDT	LSH	Exhaust	FDMA
Noise	2.5	1.72	2.39	4.51	0.84
Scale	0.601	1.38	1.4	1.28	0.26
Illumination	1.767	1.71	2.63	5.241	0.96
Rotational	1.611	1.76	2.57	7.271	1.26
Deformation	1.581	1.67	2.73	4.99	0.92



(a) Noise Variation



(b) Scale Variation

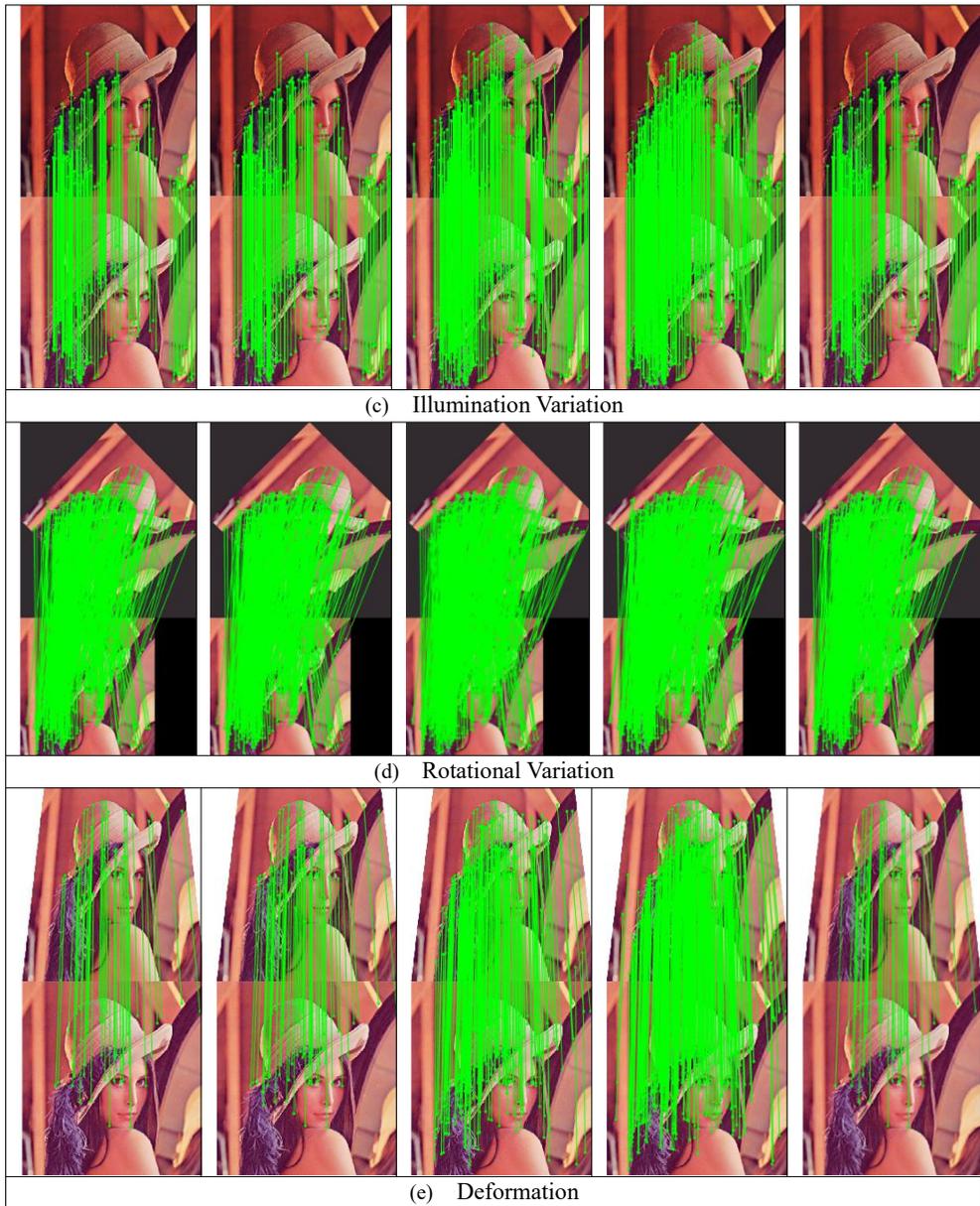


Fig. 4. Matching effect after transformation of Lena image.

4.2 Evaluation of the benchmark

To further evaluate the search performance of the FDAM algorithm, we use data sets as ours test data set. to compare the matching accuracy and matching speed. The test data set uses six different scene images, with a total of 164 images, and 809471 SIFT feature points. For each set of images, the first image is used as a query image, and its feature is matched with that of another image (reference image) in the image set respectively. We record the time consumption and the number of matched feature used by the Exhaustive, HKMT, RKDT, LSH and our FDAM algorithms. Table 4 gives the average search time spent on the

16 sets of images by the above-mentioned algorithms. Table 5 shows the average number of points in the 16 sets of images matched by the algorithms.

Compared with the other conventional algorithms, our FDAM algorithm performs better in terms of matching speed. Even when some image sets have a small number of matching points, the FDAM algorithm has a much higher number of matching points and faster matching speed than other algorithms. However, the FDAM algorithm is relatively slower when there are a large number of matching points because it spends some time on ranking during final screening.

Table 4. the average search time of the algorithm on 16 sets of images(unit s).

DataSet		HKMT	RKDT	LSH	FDMA	Exhaust
Rotation	VanGogh	2.645	3.258	4.525	2.232	14.162
	Marseil	5.449	11.236	25.117	20.848	138.182
	Monet	2.795	3.726	7.164	3.155	18.427
Zoom	Bip	4.106	6.064	10.140	5.687	34.478
	Belledonnes	5.200	5.757	7.030	4.057	30.125
	Crolles	3.719	5.082	6.979	4.422	29.802
Rotation&zoom	UBC	3.282	5.370	9.096	4.879	41.851
	Resid	3.333	5.592	13.299	6.926	37.917
	Inria	4.176	6.910	15.442	7.945	48.641
Viewpoint	Graffiti2	3.787	6.203	13.149	8.773	46.727
	Graffiti6	2.693	4.087	11.277	6.033	31.385
	Graffiti7	2.490	3.798	9.821	5.189	26.723
Illumination	MOVI	0.109	0.280	0.377	0.055	0.172
	graph	0.479	0.839	1.332	0.354	1.189
	fruit	2.561	2.144	2.884	1.213	6.647
Blur	bikes	7.468	8.076	12.620	6.592	40.040

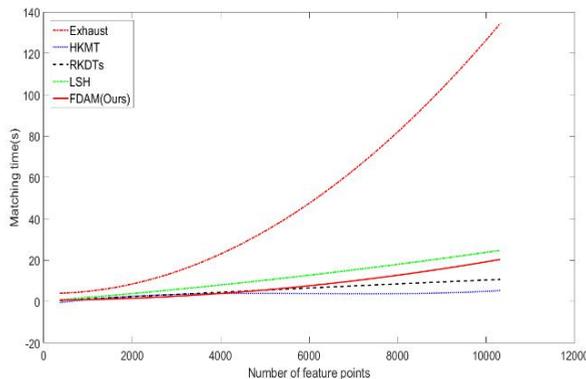


Fig. 5. The curve of matching time between RKDT, HKMT, LSH,Exhaust and FDAM with feature points.

Fig. 5 shows the curves of variation of matching time with the number of feature points among the RKDT, HKMT, LSH and FDAM algorithms.

The Table 4 and Table 5 shows that with suitable parameters, when images have variations in zoom, illumination, rotation, blur and Viewpoint, the FDAM algorithm does not have much difference in number of matching points from other algorithms and may not affect the robustness of the existing algorithms. Even in the case of viewpoint variation, the number of feature matching points searched is much large than that of other algorithms. This is because the distance function selected is different. Other algorithms use the ratio

between nearest neighbor search and secondary nearest neighbor search as the main reference quantity. In doing initial selection, the FDAM algorithm uses the feature dimension matching degree as main reference quantity and allows some dimensions to have a rather big difference. The Euclidean distance is only taken as a relative quantity to do final purification and mainly used for screening one-for-many and many-for-one cases; the FDAM algorithm also screens out a certain proportion of matching pairs that sequence to the rear in the suspect matching pairs screened out through distance sequencing after initial selection is done. This is in agreement with the actual conditions of feature points after image variations.

Table 5. The average matching points in 16 sets of images.

	DataSet	HKMT	RKDT	LSH	FDAM	Exhaust
Rotation	Van Gogh	555	520	679	635	510
	Marseil	2172	2111	2725	2253	2026
	Monet	400	385	570	485	379
Zoom	Bip	135	118	176	117	256
	Belledonnes	79	67	96	162	68
	Crolles	256	234	318	458	217
Rotation&zoom	UBC	762	729	862	690	701
	Resid	125	121	267	602	114
	Inria	81	67	292	541	64
Viewpoint	Graffiti2	6	5	56	229	4
	Graffiti6	60	49	283	671	43
	Graffiti7	44	40	199	481	33
Illumination	MOVI	45	44	59	85	44
	Graph	295	292	300	270	297
	Fruits	38	40	14	40	41
Blur	Bikes	185	176	299	260	173

5 Conclusion

This paper proposes a new SIFT feature point matching algorithm based on feature dimension matching (FDAM). The descriptor vector element is discretized into a cell array address, and the nearest neighbor search is conducted near the mapping address of the descriptor vector element. The matching speed increases linearly with the increase of the number of feature points. The proposed FDAM algorithm has matching accuracy and speed similar to, or even better than, those of the conventional RKDT, HKMT and LSH algorithms. This is particularly true when an image has viewpoint variation because the FDAM algorithm uses the feature dimension matching degree.

Acknowledgment

This work is supported by the Open Project Fund for the Key Laboratory of Photoelectric Confrontation Testing and Evaluation Technology (GKCP2016003).

References

1. Li L I, Lin F, Jun W U, et al. Exploiting global and local features for image retrieval[J]. Journal of Central South University, 2018, 25(2):259-276.

2. Luo Q M, Shi L. Review on image stitching methods[J]. *Transducer & Microsystem Technologies*, 2017..
3. Bi J, XU J, Xin Q, Shang D. An Augmented Reality Method Based on SURF and Geographic Grid Model [J]// *JISUANJI YU XIANDAIHUA*, 2017(6):47-53.
4. Jianfang X U, Liu Z, Han Z, et al. Application of Point Cloud Registration in 3D Reconstruction of Catenary Parts Based on SIFT and LBP[J]. *Journal of the China Railway Society*, 2017, 39(10):76-81..
5. Huang D, Zhang G, Ardabilian M, et al. 3D Face recognition using distinctiveness enhanced facial representations and local feature hybrid matching[C]// *IEEE International Conference on Biometrics: Theory Applications & Systems*. IEEE, 2016:1-7.
6. Lin C, Lin J, Chen Q, et al. A 3D Reconstruction Method for The Common View Field in New Hybrid Vision System[J]. *Machine Design & Research*, 2017.
7. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. *IJCV* 60 (2004) 91-110
8. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: *CVPR* (2). (2004) 506–513
9. Bay, H., Tuytelaars, T., Gool, L.: SURF: Speeded Up Robust Features. In: *ECCV* (2006) 404 – 417
10. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2005) 1615-1630
11. Bentley, L.: K-D trees for semi-dynamic point sets. In: *Symposium on Computational Geometry* (1990) 187–197
12. Nene, S., Nayar S.: A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions On Pattern Analysis and Machine Intelligence* 19(9), 989–1003(1997)
13. Silpaanan, C., Hartley, R.: Optimised KD-trees for fast image descriptor matching. In: *CVPR*. (2008) 1–8
14. Hu L, Nooshabadi S, Ahmadi M. Parallel randomized KD-tree forest on GPU cluster for image descriptor matching[C]// *IEEE International Symposium on Circuits and Systems*. IEEE, 2016:582-585.
15. Muja, M., Lowe, D.: Fast approximate nearest neighbors with automatic algorithm configuration. In: *ICCV* (1). (2009) 331-340
16. Jagadish, H., Ooi, B., Tan, K., Yu, C., Zhang, R.: Adaptive B+-tree based indexing method for nearest neighbor search. *Acm Transaction on Database Systems* 30(2), 364-397(2005)
17. Yu, C., Ooi, B., Tan, K., Jagadish, H.: Indexing the distance: An efficient method to KNN processing. *Vldb* (2001)
18. Kulis, B., Grauman, K., Kernelized,: Locality-sensitive hashing for scalable image search. In: *ICCV*(2009) 2130-2137
19. Chakrabarti A, Satuluri V, Srivathsan A, et al. A Bayesian Perspective on Locality Sensitive Hashing with Extensions for Kernel Methods[J]. *Acm Transactions on Knowledge Discovery from Data*, 2015, 10(2):1-32.
20. Yang,K,Chen ,L.,Liu, Y. ,Fast Image Feature Matching Based on Extended Cascade Original Locality Sensitive Hashing. *Computer Engineering*, 2016, 42(8): 211-219
21. Image database:<http://lear.inrialpes.fr/people/Mikolajczyk/Database/index.ht ml>
22. FLANN:<https://docs.opencv.org/3.0-beta/modules/flann/doc/flann.html>