MULTIPLE UNORDERED WIDE-BASELINE IMAGE MATCHING AND GROUPING

Zhoucan He, Qing Wang, Heng Yang
School of Computer Science and Engineering
Northwestern Polytechnical University, Xi'an 710072, P. R. China
qwang@nwpu.edu.cn

ABSTRACT

This paper focuses on the multi-view feature matching problem from unordered image sets. Firstly, an efficient and effective high dimensional feature matching algorithm is proposed, so called ELSH (extended local sensitive hash), which can significantly improve matching accuracy at fast speed. Secondly, a novel unsupervised image grouping strategy is proposed to cluster the unordered images into content-related group, which does not normally require any other constraints. Extensive experimental results have shown that our method can obtain better performance than the classical algorithms in tackling multi-view matching problem.

Key words: multi-view matching, KNN (K nearest neighbor) search, image grouping, extended LSH

1. INTRODUCTION

Unordered multi-view image matching is a fundamental task in computer vision areas, such as image-based modeling [9], panorama image building[12] and so on. Generally speaking, there are three main steps concerned[13]: extracting and describing local invariant features from unordered image set, matching features of each two views and tracking all matches across multi-views. In this paper we firstly extract SIFT (scale invariant feature transform) [1] feature from unordered image sets, then propose a novel feature matching algorithm to match the features of every two-view, and finally cluster content-related images into different groups with the proposed strategy based on the results of feature matching.

Related work on image feature matching. Actually, a feature is a high dimensional vector, and SIFT algorithm declared that the nearest neighbor is selected as a determinate match only if it is much closer than the second one [1,2]. Consequently, the feature matching problem can be transformed to KNN search problem. There are many kinds of classical search methods for high dimensional data, such as kd-tree, BBF (best bin first) [7], iDistance (indexing the distance) [6], (LSH)[8] and so on. Kd-tree is a binary balanced tree which can only solve KNN searching in low dimensional space effectively. BBF is an expanded version of kd-tree, which optimizes the search step and is capable of handling high dimensional data such as SIFT. iDistance based method is used to cluster data into several classes and establish one B+tree for each class according to the distance between data point and the class center. On the other hand, LSH projects data into Hamming space, and chooses some specific hash functions based on some randomly selected Hamming dimensions to hash features, then similar features would be placed into a corresponding hash bucket and so does the query feature. The proposed feature matching algorithm is an extension of LSH, which can improve local sensitivity massively and perform better than traditional LSH in search accuracy and speed.

Related work on image grouping. Organizing unordered image sets is a challenging task for multi-view based 3D scene modeling and other applications. Schaffalitzky et al. [10] established a view-versus-view hash table which can lead to greedy choice for a spanning-view tree for multi-view match. Yao et al.[9] first carried out a partial matching on every two views, and based on the two-view match results a view-similarity measure is proposed to establish a spanning tree. Apart from the two-view matching results, an epipolar geometric constraint is further involved to enhance the performance of image grouping. Zeng[13] proposed an annealing-based algorithm for images clustering, which is more effective than the exhaustive one. In fact our novel robust matching algorithm can provide view-similarity measurement as a basis for any clustering approach for unordered image set.

The remainder of the paper is organized as follows. The extended LSH algorithm for feature matching and a more robust view-similarity measure for content-related image grouping are presented in Section 2 and 3, respectively. Experimental results are discussed extensively in Section 4 and conclusions are drawn in Section 5 finally.

2. HIGH DIMENSIONAL FEATURE MATCHING

Problem description. Given two high dimensional SIFT feature sets *A* and *B*, the aim is to search 2NN points from *A* for each query point in *B*. If the nearest neighbor is much closer than the second one, it is regarded as a determinate match for the query point.

The original LSH algorithm is described as following: firstly, find the maximum value *C* (256 in SIFT feature sets) from all dimensions in *A* and project *A* and *B* into Hamming space. Secondly, select *K* different hash functions, and for each function, L random Hamming dimensions are chosen for hashing features in A. At last, for a query point in B, it is hashed into some corresponding buckets based on the hash functions and the same L Hamming dimensions. In fact, two parameters, K and L enable the designer to select an appropriate trade-off between accuracy and running time [4]. LSH is truly a fine high dimensional search algorithm which has been widely used in image matching and image retrieval applications. However, it also has some limitations, since the local sensitivity of one Hamming dimension drops seriously when C is large. One can change K or L to keep those similar data distributing in a same bucket with high probability, however, it would contain some dissimilar data and the time cost increases consequently.

Extended LSH. There are three major improvements in our algorithm. The first and the most important one is that we project the feature sets into an equally-distributed space, not the Hamming space, which can significantly enhance the local sensitivity of search algorithm. The reason is that LSH fails to find good result when C is large. But fixed projection in equally-distributed space does not face this issue. The second one is that a simple mapping method is employed instead of more hash functions and, because L (16 in this paper) is small for the high sensibility in the new projecting space. The last one is that the upper bound of search times is set to 200 to guarantee the search efficiency. The extended LSH algorithm is described as follows:

Step 1: Calculate the data distributions for each dimension in the feature set A, and select M-1 values to divide set A into M equal parts on each dimension.

Step 2: Establish K search indexes $\{I_1, I_2, ..., I_K\}$. For each index $I_m, m = 1,...,k$, we randomly find L dimensions, and for each dimension, a key value is randomly determined from the M-1 partition values. K is set to 120, which is large enough.

Step 3: For every feature in A, if it is bigger than the key value of certain randomly selected dimension, it will be assigned 1, else assigned 0. Certainly, a binary string is obtained based on the L randomly selected dimensions in index $I_m, m = 1,...,k$. Obviously, features with the same binary string are mapped into the same hash bucket.

Step 4: For a query point in B, execute following process:

Input: query point q, k search indexes $\{I_1, I_2, ..., I_K\}$
Output: KNN features for q
search_time=0;
for m:=1 to K
Find out the corresponding hash bucket S_q, where q is placed according to the L key values in I_m.
Execute exhaustive search in the set S_q, and accumulate the comparison times to *search_time*.
If *search_time*>200
break;
end if
end for

3. CONTENT-RELATED IMAGES GROUPING

In this section, we discuss how to classify multi-view unordered images into content-related groups. No matter what method is chosen, a certain view-similarity measure should be defined in advance to evaluate the relevancy between two images. In fact, the precision of similarity measure is the key issue to the performance of clustering method.

Discarding reduplicate structure. Although SIFT feature is the most robust local feature descriptor, there exists a bad phenomenon sometimes when two irrelevant images are matched, which is that one feature may match with lots of points in another image simultaneously (we call this match "reduplicate structure"). It is not a technical problem but hard to find out, which will seriously affect the grouping result and other applications, such as image retrieval, image based modeling and so on. Herein, we highly recommend excluding such reduplicating structures before further processing. One could tackle the issue by labeling the matched features and check the coarse correspondence.

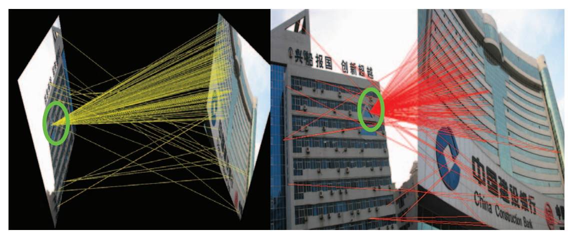


Figure 1. Reduplicate structures

A more robust view-similarity measure. Schaffalitzky[10] clustered images by the number of the matched features, and Yao[9] employed a complex formula, the latter seems more effective but it adopts other assistant methods such as RANSAC which is more complex, and both can not cope with the two related images in the case that one has large size and the other is small. We propose a new view-similarity measure as following:

$$Sim(A,B) = \frac{n_{match}}{\min\{n_A, n_B\}} * \sum_{i=1}^{n_{match}} \frac{1}{d_i} \tag{1}$$

where n_A, n_B denote the size of the feature sets A and B, respectively, n_{match} is the matched number of features, and d_i denotes the distance of two matched features. Obviously, the distance of two matched features is likely to be smaller than those unmatched and it will contribute more on the view-similarity, and the matched number will count slightly. Based this basic measure, one can use any cluster strategy such as view-spanning tree employed to organize the unordered images, in this paper, an adjacency matrix is used. The adjacency matrix is generated as follows: firstly, add the image pairs with highest view-similarity value to the adjacency matrix, then add a new image which has the highest view-similarity value with existing images, repeat the process until no image's view-similarity value surpass a

Table 1. The accuracy (%) and time cost (ms) of 2NN search. (Acc1 and Acc2 are the accuracies of 1-NN and 2-NN.)

		Boat	Bricks	Car	East park	Ensimag	Graffiti	Inria	Resid	Toy	Trees
Image number		10	6	6	10	10	9	10	10	10	6
Exhaustive	Time	15756	39862	3147	7092	8309	3816	4512	2743	432	42712
LSH	Acc1	50.52	45.14	66.28	58.53	47.00	56.78	75.53	72.39	46.20	37.17
	Acc2	26.92	16.33	34.67	38.50	28.34	38.25	60.14	56.66	10.26	18.76
	Time	1031	2181	274	753	677	492	786	522	41	2084
BBF	Acc1	48.15	45.66	62.56	54.06	49.52	56.67	63.07	68.04	78.57	43.89
	Acc2	21.99	17.47	31.72	27.97	24.44	32.24	38.40	44.64	52.58	18.78
	Time	795	2631	318	487	531	464	321	276	123	2367
iDistance	Acc1	59.17	58.62	68.25	55.54	55.12	56.26	61.10	59.92	73.58	52.90
	Acc2	33.90	31.57	40.71	30.52	32.59	33.24	38.55	36.78	46.05	28.88
	Time	1432	4012	390	562	1664	466	309	224	99	3687
Our algorithm	Acc1	61.71	59.79	82.57	67.49	62.96	66.05	70.48	71.23	69.26	54.13
	Acc2	33.09	27.02	56.21	41.34	38.24	43.16	46.64	47.55	32.32	25.51
	Time	712	1725	315	553	578	447	340	300	66	1494

given threshold. A new adjacency matrix would be generated when there are some remaining images. We make a lot of tests on different image sets, and even resize the size of the corresponding images greatly on purpose. As a result, we can still obtain the accurate grouping results.

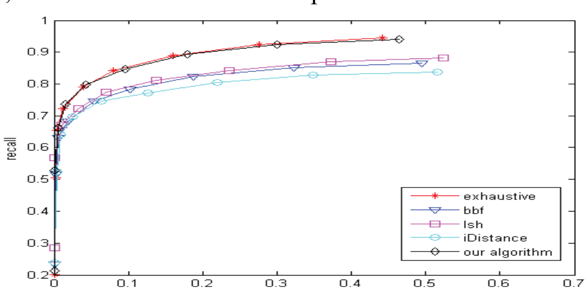
4. EXPERIMENTAL RESULTS AND ANSLYSES

In this section, the results of feature matching and image grouping on two standard databases [14,15] are shown and discussed in details. The experiment environment is Intel(R) Core(TM)2 Duo CPU E6550 @2.33GHz,2.0 GB memory.

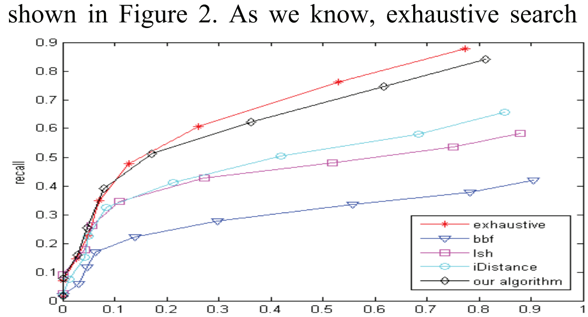
4.1 Feature matching

2NN. We have realized three classical methods, including BBF, iDistance and LSH for comparison. The first task is to compare the performance for KNN search and then image matching on the database [14]. Table 1 shows the result of accuracy and time cost for 2NN search on 10 teams of images. For each team, we randomly select an image as the first one to build index and other images are used to search on it. The ground truth results are obtained in advance by using exhaustive search. From Table 1 one can find that our proposed algorithm has the highest accuracy and lowest time consuming for 2NN search.

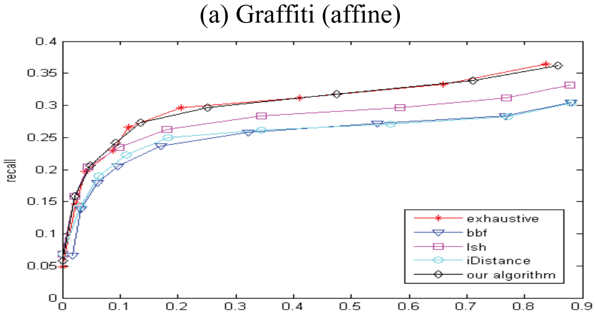
Image matching. We take the widely used criterion of recall vs. 1-precision curve[5] to evaluate the capability of these algorithms for image matching. Experiments on four pairs of images, including scale, illumination, noise and weak affine transforms, are carried out, and the results are shown in Figure 2. As we know, exhaustive search is the



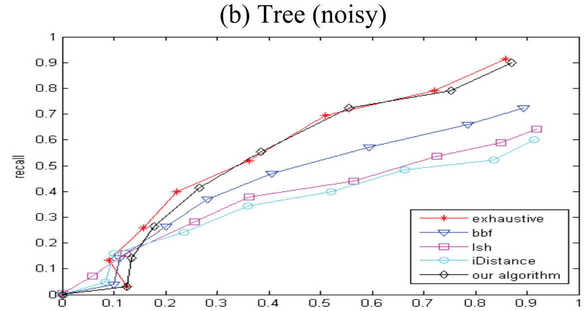
(a) Graffiti (affine)



(b) Tree (noisy)



(c) East_park (rotation vs scale)



(d) Car (illumination)

Figure 2. SIFT based image matching results under different transforms.

Figure 3 The results of content-related image grouping.

baseline, and the curve of our proposed method curve is very close to it. Moreover, the time cost is the lowest comparing to other methods.

4.2 Content-related image grouping

Just like paper [9] and [10], we also use the Church and Castle images [15] to evaluate the performance of image grouping. To be different, we mix all 15 Church images and 46 Castle images together into one set and adopt the proposed measure to classify these images. By the way, we use adjacency matrix to organize the group data and the grouping results are shown in Figure 3. The results are encouraging and similar to Yao's one, and the largest group has 18 views. However, no more constraint is required in our method, which is different from Yao's method. The reason is that the proposed view-similarity measure is reasonable and effective.

5. CONCLUSIONS

This paper proposes an extended LSH algorithm which can produce high accurate image matching results in the lowest time cost comparing to those classical methods. And a robust view-similarity measure is also proposed, upon which multiple unordered images can be classified correctly. In the future, our new algorithm can be applied to other computer vision applications, such as image retrieval.

ACKNOWLEDGMENTS

This work is supported by National Hi-Tech Development Programs under grant No. 2007AA01Z314 and National Natural Science Fund (60873085), P. R. China.

REFERENCES

[1]David G. Lowe. Distinctive Image Features from Scale-Invariant Key points. Int Journal of Computer Vision 60(2), pp. 91-110, 2004

[2]Dusan,Omercevic, Ondrej Drbohlav, and Ales Leonardis .High-Dimensional Feature Matching: Employing the Concept of Meaningful Nearest Neighbors. In Proc. ICCV,2007.

[3]Lu Yansheng, Rao Qi .A self-tuning method of LSH index[J]. Journal of Huazhong University of Science and Technology (Nature Science).34(11), pp.34-37,2006

[4]Y. Ke, R. Sukthankar, and L. Huston, An efficient parts-based near-duplicate and sub-image retrieval system. in Proc. ACM-MM, pp. 869–876,2004

[5]Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," Proc. Conf. Computer Vision and Pattern Recognition, pp. 511-517, 2004.

[6]H.V. Jagadish, Beng Chin Ooi,Kian-Lee Tan, Cui Yu, Rui Zhang. iDistance : An adaptive B + 2tree based indexing method for nearest neighbor search [J] . ACM Transactions on Data Base Systems , pp364-397. , 2005

[7]Jeffrey S. Beis and David G. Lowe. Shape Indexing Using Approximate Nearest-Neighbor Search in High-Dimensional Spaces. Proc. Conf. Computer Vision and Pattern Recognition, pp.1000–1006, 1997

[8]Aristides Gionis, Piotr Indyky and Rajeev Motwaniz. Similarity Search in High Dimensions via Hashing. In The VLDB Journal, pp. 518–529, 1999.

[9]J. Yao and W.K. Cham. Robust multi-view feature matching from multiple unordered views. Pattern Recognition, 40:3081-3099, 2007.

[10] F. Schaffalitzky, and A. Zisserman, Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?" In: ECCV, vol. 1, pp. 414-431, 2002.

[11]Krystian Mikolajczyk and Cordelia Schmid. A Performance Evaluation of Local Descriptors. IEEE Transactions on PAMI, (2005)1615–1630.

[12]M. Brown, D.G. Lowe, Recognising panoramas, in: International Conference on Computer Vision (ICCV), vol. 3, Nice, France, October 2003, pp. 1218-1225.

[13] X. Zeng, Q. Wang and J. Xu, MAP Model for Large-scale 3D Reconstruction and Coarse Matching for Unordered Wide-baseline Photos In:BMVC ,2008

[14] http://lear.inrialpes.fr/people/Mikolajczyk

[15]http://www.robots.ox.ac.uk/%7Evgg/data/data-mview.html