1.5 Million Subspaces of a Local Feature Space for 3D Object Recognition

Koichi Kise Takahiro Kashiwagi

Department of Computer science and Intelligent Systems
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuencho, Naka, Sakai, Osaka 599-8531, Japan
kise@cs.osakafu-u.ac.jp kashiwagi@cs.osakafu-u.ac.jp

***Abstract*—We propose 3D object recognition methods whose characteristic point is the use of a large number of subspaces (1.5 million) generated from a billion of local features for the recognition of 1002 objects. In order to match query local features to a lot of subspaces, a simple approximate nearest neighbor search is utilized. Based on this approximation the proposed three methods are as follows: a method with an ordinary subspace method (match each query local feature to subspaces), a method of two-step matching which employs two versions of subspaces for efficient matching, and a method with a mutual subspace method (both queries and models are represented as subspaces). These methods are compared with two baselines with and without subspaces. From experimental results, we have confirmed the advantage of proposed methods in both accuracy and efficiency. In particular, the mutual subspace method achieves 95% accuracy with processing time of 3.5 sec./query, which improves the accuracy of a baseline without subspaces about 60%. As compared to the subspace method without approximate matching, the mutual subspace method is more than 240 times faster.**

I. Introduction

Appearance-based approach is a well-known and powerful strategy for object recognition. Representative methods include the parametric eigenspace method [1] for 3D object recognition and pose estimation. A disadvantage of such methods is that they require segmentation of objects prior to recognition, since they employ a global feature assuming that an input image only captures an object to be recognized. The segmentation is, however, not an easy task as compared to recognition, due, for example, to occlusion as well as view and illumination changes.

A possible strategy to solve this problem is to employ local features such as SIFT [2]. Since such features are local and invariant to a certain class of geometric distortion, they are less sensitive to the above problems. Based on this strategy, several methods [3], [4], [5] have already been proposed. Unfortunately however, these methods are not easy to use, since they require long processing time because of the number of local features for matching.

The main contribution of this paper is to propose a method of 3D object recognition with local features that improves the efficiency. Taking as query a single or a sequence of images, the method returns the best match among objects modeled by using sequences of images. The key strategy is to use subspace methods and their approximate matching. We utilize subspaces each of which represents a set of local features extracted from the same part of an object captured in different viewing angles. This allows us to represent local features compactly. The number of subspaces is still large, i.e., 1.5 million for 1002 objects, which is far more than the one to be matched in reasonable time. We solve this problem by approximate matching of subspaces. One-dimensional subspaces allow us to employ a standard approximate nearest neighbor (NN) search for extremely efficient matching of subspaces. The result of object recognition is obtained by *voting* for objects based on matching. With the *mutual* subspace method, a query object can be recognized with the accuracy of 95% in 3.5 sec.

II. Related Work

A. 3D Object Recognition

We focus here on methods which employ local features. The simplest way of recognition is to store, for each object, all local features extracted from images taken around the view sphere as a model of the object, and find the best match to each local feature extracted from a query. This approach does not scale well because of a huge number of local features.

At an early stage of 3D object recognition with local features, Lowe proposed a method called *view clustering* [6] for more compact models of objects. In this method, training images that are visually close with one another are combined to form visual clusters, which result in reducing the number of images in the model. Kim and Kweon have proposed a more advanced method [5] that allows feature sharing in addition to the view clustering. In the method, features are shared even among different objects for more compact models.

Since local features are extracted from the surface of an object, they are geometrically constrained. In order to employ this constraint, Rothganger and Lazebnik build a 3D object model consisting of small image patches [3]. Once affine invariant patches are placed in the 3D space, images of any views can be matched to the model. Ferrari *et al.* have also proposed a method that takes into account 3D geometric constraints. Their method starts with initial small matches and tries to expand matching areas by the process called *image exploration* which takes into account the constraints.

A major drawback of the above methods is their computational load. Even for the matching of a single image with an object model, these methods require processing time of several seconds or minutes. Özuysal *et al.* have proposed a method of fast keypoint recognition [7]. As compared to the above methods, the method can recognize a keypoint, which can be thought of as a class of local features, in real-time. However the effectiveness of the method for a large-scale 3D object recognition has not yet been known. If the number of 3D objects to be recognized increases, the number of similar keypoints could also be increased. This may result in deteriorating the recognition accuracy.

As compared to these methods we propose a method of 3D object recognition applicable to a large-scale database. This method is based on a large number of subspaces spanned by view clusters of local features.

B. Subspace Methods

Next, let us summarize subspace methods. In subspace methods, principal component analysis (PCA) is applied to a collection of images of *each* object so as to represent each object as a subspace. The original subspace method proposed by Watanabe [8], which is referred to as the *ordinary subspace method* in this paper, projects an input feature vector onto the subspace of each object to calculate similarity between the feature vector and the subspace.

The ordinary subspace method has been extended in many different ways. A representative extension is that a query is also represented as a subspace and canonical angles between subspaces are employed to define their similarity. This method is called the *mutual subspace method* [9].

For the application of 3D object recognition, for example, Fukui *et al.* have proposed a framework of kernel constrained mutual subspace method [9]. Most methods including this are to deal with the whole images to obtain the subspace of an object. In contrast to these methods we construct subspaces for clusters of local features. The number is more than 1,500 for each object. To deal with such a large number of subspaces we need to employ efficient matching.

C. Approximate Matching

There are many methods for efficient matching of feature vectors with the help of approximation such as ANN [10] and LSH [11]. However these methods are based upon NN search with Euclidean distance so that we cannot apply them directly to matching of subspaces.

Basri *et al.* have recently proposed a new method of approximate matching for subspaces called *approximate nearest subspace search* [12], which is applicable to both the ordinary and the mutual subspace methods. In their work the problem of finding the best subspace is converted to the problem of approximate NN search between vectors. This theoretically solid method is, however, not easy to apply due to the computational complexity caused by the conversion. As mentioned in [12], the complexity of $O(d^2)$ is intolerably large for large-scale applications, where d is the dimension of vectors.

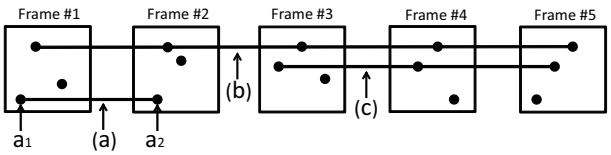

Fig. 1. Chains of local features for subspace construction.

This paper presents a simpler but still powerful approach by taking advantage of a large number of subspaces. If we have many subspaces that describe each object, and the recognition is done by voting based on matching of subspaces, each matching is not necessarily be so accurate thanks to the voting effect. In other words, we can introduce a drastic approximation for matching to make the overall process efficient.

III. Proposed Methods

The proposed methods employ a large number of subspaces generated from local features of images with different viewing angles of an object. The representation as subspaces enables us to reduce the memory space as compared to storing all local features. In addition, we utilize an approximate NN search for matching subspaces so that efficient recognition is still possible even with a large number of subspaces.

A. Selection of Local Features

The first step is to select local features for constructing subspaces. We apply a kind of view clustering not for images but for local features. In the proposed methods, each object is captured as a sequence of frame images by rotating it on a turntable. From each frame image we extract SIFT features [2]. Although there are some SIFT features that are not stable due to the shape of an object, etc., some SIFT features stably exist in many consecutive frames. We focus here on such features.

Figure 1 shows this process where black dots represent local features. We apply NN matching of local features between consecutive frames. Let f be a local feature of a frame i, g_1 and g_2 be the nearest and the second nearest local features of a frame $i+1$, respectively. The local feature f and g_1 are matched if $d(f,g_1) < d(f,g_2)/2$, where $d(f,g)$ is the Euclidean distance between local features f and g.

By applying this process to all pairs of frames, we obtain sequences of matched local features, or *chains* shown as lines in Fig. 1, where (a), (b) and (c) represent chains of two, five, and three local features, respectively. The minimum number of local features in a chain, or the minimum *length* T of a chain, is utilize to control the number of generated chains. In this paper, the above process of obtaining chains is called *tracing*.

B. Construction of Subspaces

A subspace is generated from each chain of local features by applying PCA. The resultant subspace represents a variation of local features caused by different viewpoints. This allows us to interpolate local features between frames.

Let $\boldsymbol{x}$ be a vector of a local feature in a chain. The auto-correlation matrix Q of $\boldsymbol{x}$ is defined as

$$Q = E\{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\}. \tag{1}$$

By solving the eigenvalue problem

$$Q\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \tag{2}$$

where λ_i ($\lambda_i \geq \lambda_j$ if $i > j$) and $\boldsymbol{u}_i$ are an i-th eigenvalue and its eigenvector, respectively, the subspace for $\{\boldsymbol{x}\}$ is defined as the space spanned by eigenvectors $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_n\}$ which correspond to the first n largest eigenvalues. We can decrease matching time and memory for storage if we use a smaller n. As the object model in the database, we record with the object ID the above subspaces extracted from all chains.

C. Recognition by Distance Calculation

The simplest proposed method is based on the ordinary subspace method. Let $\boldsymbol{q}$ be a local feature extracted from a query image. The similarity S between $\boldsymbol{q}$ and a subspace is obtained by projecting $\boldsymbol{q}$ onto the subspace $\{\boldsymbol{u}_i\}$:

$$S^2 = \sum_{i=1}^{n} (\boldsymbol{q}^{\mathrm{T}} \boldsymbol{u}_i)^2. \tag{3}$$

The subspace that gives the largest S is employed to vote for the corresponding object. Let v_j be the number of votes for an object j. In order to equalize the difference of the number of subspaces for each object, we utilize the weighted votes $w_j = v_j / \sqrt{N_j}$ where N_j is the number of subspaces of an object j. The recognition result is obtained as the object with the largest sum of weighted votes.

However, this simple process of projection is computationally prohibitive even with a small n because of the large number of subspaces. In order to speed up the process, we introduce an approximate NN search.

In general, we cannot simply convert projection to NN search; a special care such as described in [12] is required. However in the special case of one-dimensional subspace, the NN search is equivalent to projection. Since

$$\|\boldsymbol{q} - \boldsymbol{u}_1\|^2 = \|\boldsymbol{q}\|^2 - 2(\boldsymbol{q}^{\mathrm{T}} \boldsymbol{u}_1) + \|\boldsymbol{u}_1\|^2, \tag{4}$$

the similarity S is calculated by

$$S^2 = \boldsymbol{q}^{\mathrm{T}} \boldsymbol{u}_1 \tag{5}$$

$$= \frac{\|\boldsymbol{q}\|^2 + \|\boldsymbol{u}_1\|^2 - \|\boldsymbol{q} - \boldsymbol{u}_1\|^2}{2} \tag{6}$$

$$= -\frac{1}{2}\|\boldsymbol{q} - \boldsymbol{u}_1\|^2 + C. \tag{7}$$

with the normalized $\boldsymbol{q}$ and $\boldsymbol{u}_1$, where C is a constant which does not affect the matching. This allows us to introduce approximate NN search.

Readers may consider that using just one dimension is too approximate to represent meaningful information. However, as the experimental results described below show, it is not a bad choice thanks to the voting effect.

D. Two-step Matching

In general, however, it is better to use more dimensions to improve the accuracy. To achieve this with little influence on efficiency, we use two-step matching. At the first step the one-dimensional subspaces are used only for selecting k candidate subspaces. At the second step, the candidate subspaces are examined with larger dimensions $n(> 1)$ to select the best. To be precise, a query vector is projected onto each of the k subspaces of n dimensions and the one with the highest similarity is employed for voting.

Fig. 2. Examples of objects in the database.

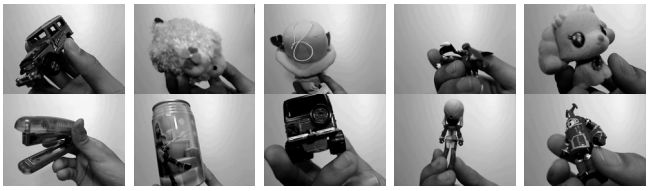

Fig. 3. Examples of query images.

E. Mutual Subspace Method

If a query is given as video, the mutual subspace method can be applied. From a sequence of query images, chains of local features with the minimum length T_q can also be extracted. This allows us to generate subspaces for queries.

The mutual subspace method is to compute similarity between a query and a database subspace. In general, the similarity is defined based on canonical angles, which are not easily handled by approximate matching. However, likewise the previous case, it can be easily converted to NN search if subspaces are of one dimension. The similarity can be calculated using the Eq. (7) by replacing $\boldsymbol{q}$ to the eigenvector of the largest eigenvalue for the query subspace. The same weighted voting process is employed to obtain recognition results.

IV. Experiments

A. Setup

We evaluated the proposed methods by using our own dataset which will be publicly available. The dataset consists of 1002 small objects that include many mini cars, figurines and stuffed dolls as shown in Fig. 2. Since these objects tend to be of similar shapes and textures they are not easy to distinguish. In addition, figurines are with a lot of peaked parts that cause rapid change of their appearances.

Each object was rotated on a turntable and captured using a web cam with VGA resolution. Three elevation angles were employed: 0° (horizontal), 15° and 30°. Subspaces were generated with the minimum length $T = 50$ of chains. The statistics of the database are listed in Table I.

Queries were obtained by capturing 100 objects randomly selected from the database. In order to make queries realistic,

TABLE I
STATISTICS OF THE OBJECT DATABASE.

No. of objects	1,002
No. of images / obj.	2,335
No. of local features / obj.	1.14×10^6
No. of local features selected by tracing / obj.	3.0×10^5
No. of subspaces / obj.	1,567

TABLE II
STATISTICS OF QUERIES.

	ordinary subspace	mutual subspace
No. of objects	100	100
No. of images / obj.	5	600
No. of queries	500	100
No. of local features / query	350	2.31×10^5
No. of local features selected by tracing / query	—	3.2×10^3 ($T_q = 50$) $\sim 3.8 \times 10^4$ ($T_q = 10$)
No. of subspaces / query	—	50 ($T_q = 50$) $\sim$ 1,950 ($T_q = 10$)

these objects were captured as video while holding them by hand and rotating them. For the ordinary subspace, we sampled five frames randomly from the captured images. For the mutual subspace, we utilized all frames with various minimum lengths $T_q \in \{10, 20, 30, 40, 50\}$ of chains in a query. The statistics of queries are shown in Table II.

The proposed methods employed for experiments were:

- P1 method based on the ordinary subspace method,
- P2 method based on the two-step matching,
- P3 method based on the mutual subspace method,

all of which are with approximate NN search. In order to evaluate the effectiveness of the above methods, we prepared two baseline methods:

- B1 method based on matching of local features without subspaces but with approximate NN search,
- B2 method based on the ordinary subspace method without approximate NN search.

By comparing to the baseline B1, we can evaluate the effect of subspace methods. Since the number of local features are so large that we cannot apply this simple matching method without approximate NN search. In this method local features selected by tracing and sampled from every four frames were stored in the database for matching. The baseline method B2 was employed to evaluate the effect of approximation.

As the method of approximate NN search we employed ANN [10] which allows us to control the approximation by reducing the radius of search space with the ratio $1/(1+\varepsilon)$ where ε is a parameter. The matching is more efficient but approximate with a larger ε.

B. Effectiveness of Subspaces (B2)

The first experiment was to evaluate the effectiveness of subspaces. We employed the method B2 with a range of dimensions $n = 1$ to 10. For the purpose of comparison we also employed the method B1. The number of local features / obj. for B1 was 6.0×10^4.

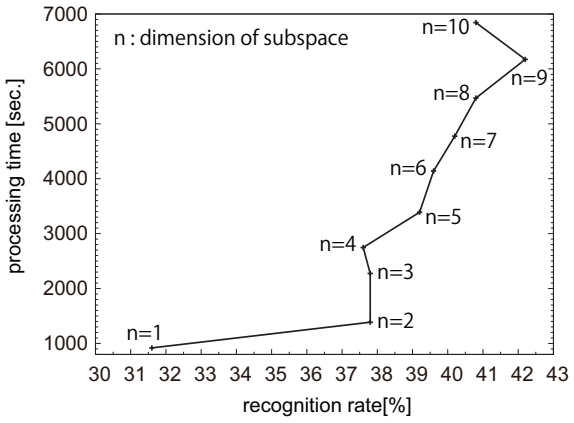

Fig. 4. Results of a subspace method without approximation (B2).

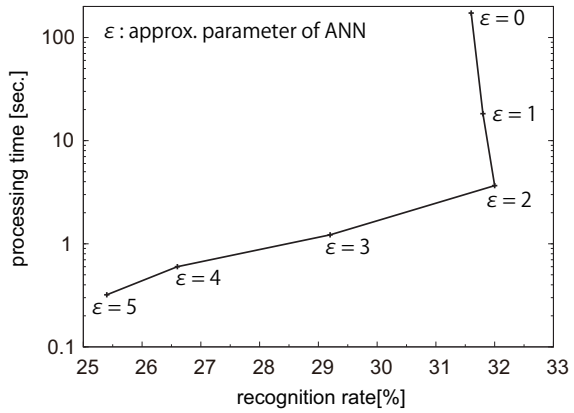

Fig. 5. Results of a subspace method with approximation (P1).

Figure 4 shows results of the method B2, where the vertical axis indicates average time of matching for a single query. As shown in this figure, larger dimensions tend to allow us better recognition rates up to $n = 9$ at the price of longer processing time. The recognition rate by the method B1 was 35.8% with the processing time 75.5 sec./query in the case with the approximation parameter $\varepsilon = 1$.

From this experiment, we have confirmed that the ordinary subspace method allows us better recognition, though it is not practical due to intolerably long processing time.

C. Effectiveness of Approximate NN search (P1)

Next, we tested the effect of approximate NN search for subspace methods by changing the parameter ε. The method employed in this experiment was P1 with one-dimensional subspaces ($n = 1$). Figure 5 shows the results where $\varepsilon = 0$ indicates exact NN search. From $\varepsilon = 0$ to 2, we were able to keep the recognition rate, while cutting drastically the processing time. We achieved more than 47 times faster processing with $\varepsilon = 2$ as compared to the exact NN search.

D. Effectiveness of Two-step Matching (P2)

Although the approximate NN search is powerful to reduce the processing time, the accuracy achieved by one-dimensional

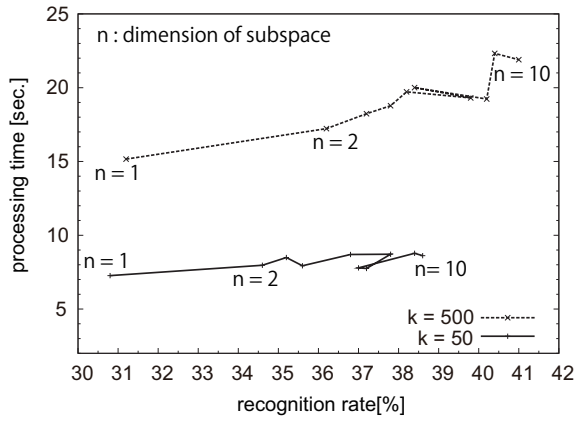

Fig. 6. Results of two-step matching (P2).

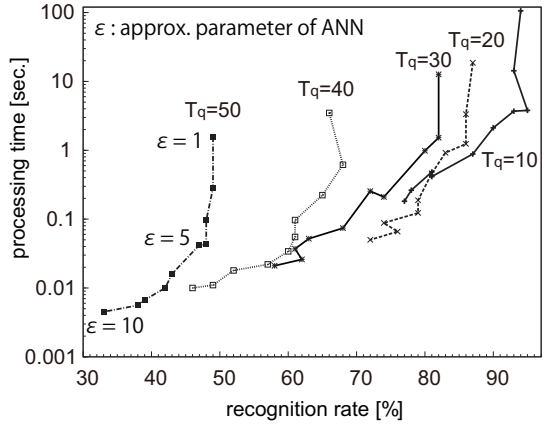

Fig. 7. Results of the mutual subspace method (P3).

subspaces is limited. Thus in this experiment, we tested the method P2 designed to overcome this problem.

The parameter of approximation ε was fixed to 2 since this achieved the best result in the previous experiment. We varied the dimension n of subspaces in the second step. As for the number k of candidate subspaces, we tried 50 and 500.

Results are shown in Fig. 6. For the smaller $k = 50$, there were only small differences of processing time even when n changed. This is simply because of the smaller number of candidate subspaces. In contrast, the results with $k = 500$, longer processing time was required with a larger n. Larger n and k improve the results at the sacrifice of processing time. As compared to the result of B2 with $n = 9$, P2 achieved a slightly better recognition rate 41.0% ($n = 10$ and $k = 500$) with the processing more than 300 times faster than B2. However, P2 did not allow a significant improvement from P1.

E. Effectiveness of Mutual Subspace Method (P3)

Finally, we evaluated the mutual subspace method (P3). Results are illustrated in Fig. 7. As compared to the results by B1, B2, P1 and P2, the recognition rate was improved more than 50%. It is remarkable that around $\varepsilon = 2 \sim 5$, big falls of processing time were observed while keeping recognition rates. The best recognition rate 95% was achieved at $\varepsilon = 3$ with the parameter $T_q = 10$ that generated the largest number of query subspaces. The processing time was 3.8 sec./query, which is more than 240 times faster than B2 with $n = 1$. If the user is interested in faster processing, a recognition rate 42% close to the best 41% by the method P2 ($n = 10$, $k = 500$) was obtained with the processing time 0.01 sec., which is more than 2,000 times faster than P2. From these results we have confirmed that the mutual subspace method is the best within the proposed methods and far better than the baselines.

V. Conclusion

We have presented three methods of 3D object recognition which utilize 1.5 million subspaces spanned by chains of local features. From the experimental results the method based on mutual subspaces, which employ approximate matching between query subspaces and object subspaces, allows us the recognition rate of 95% with processing time 3.8 sec./query. This is much better than ordinary subspace methods as well as methods without subspaces. Future work includes further speed-up of recognition as well as the improvement of recognition accuracy by using more advanced subspace methods such as the kernel mutual subspace method.

Acknowledgment

This work was supported in part by the Grant-in-Aid for Scientific Research (B) (20300049) from Japan Society for the Promotion of Science (JSPS).

References

[1] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *Int'l Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] F. Rothganger and S. Lazebnik, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int'l Journal of Computer Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[4] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Simultaneous object recognition and segmentation by image exploration," in *Toward Category-Level Object Recognition*, ser. LNCS 4170, J. P. et al., Ed. Springer-Verlag, 2006, pp. 145–169.

[5] S. Kim and I. S. Kweon, "Scalable representation for 3d object recognition using feature sharing and view clustering," *Pattern Recognition*, vol. 41, pp. 754–773, 2008.

[6] D. G. Lowe, "Local feature view clustering for 3d object recognition," in *Proc. CVPR2001*, 2001.

[7] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Trans. PAMI*, vol. 32, no. 3, pp. 448–461, March 2010.

[8] S. Watanabe and N. Pakvasa, "Subspace method in pattern recognition," in *Proc. 1st IJCPR*, 1973.

[9] K. Fukui, B. Stenger, and O. Yamaguchi, "A framework for 3d object recognition using the kernel constrained mutual subspace method," in *Proc. ACCV2006*, 2006, pp. 315–324.

[10] S. Arya, D. M. Mount, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching," *Journal of the ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[11] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Comm. of the ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[12] R. Basri, T. Hassner, and L. Zelnik-Mnor, "Approximate nearest subspace search," *IEEE Trans. PAMI*, vol. 33, no. 2, pp. 266–278, February 2011.