Random Angular Projection for Fast Nearest Subspace Search

Binshuai Wang[1], Xianglong Liu[1*], Ke Xia[1], Kotagiri Ramamohanarao[2], and Dacheng Tao[3]

[1] State Key Lab of Software Development Environment, Beihang University, China
[2] Department of CIS, The University of Melbourne, Australia
[3] UBTECH Sydney AI Centre, School of IT, FEIT, University of Sydney, Australia

Abstract. Finding the nearest subspace is a fundamental problem in many tasks such as recognition, retrieval and optimization. This hard topic has been seldom touched in the literature, except a very few studies that address it using the locality sensitive hashing for subspaces. The existing solutions severely suffer from poor scaling with expensive computational cost or unsatisfying accuracy, when the subspaces originally distribute with arbitrary dimensions. To address these problems, this paper proposes a new and efficient family of locality sensitive hash for linear subspaces of arbitrary dimension. It preserves the angular distances among subspaces by randomly projecting their orthonormal basis and further encoding them with binary codes. The proposed method enjoys fast computation and meanwhile keeps the strong collision probability. Moreover, it owns flexibility to easily balance the performance between the accuracy and efficiency. The experimental results in the tasks of face recognition and video de-duplication demonstrate that the proposed method significantly outperforms the state-of-the-art vector and subspace hashing methods, in terms of both accuracy and efficiency (up to 16× speedup).

Keywords: Nearest Subspace Search, Locality Sensitive Hash, Linear Subspace, Subspace Hashing

1 Introduction

Subspaces enjoy strong capability of describing the intrinsic structures of the data naturally. Therefore, they often offer convenient means for the data representation in the fields of pattern recognition, computer vision, and statistical learning. Typical applications include face recognition [11, 23], video retrieval [26], person re-identification [24], and 2D/3D matching [17, 19]. In these applications, usually finding the nearest subspaces (*i.e.*, subspace-to-subspace search) is a fundamental problem and inevitably involved, which means that when the amount of data explodes, searching the gigantic subspace database with high ambient dimensions will consume expensive storage and computation [8, 9].

* corresponding author (xlliu@nlsde.buaa.edu.cn)

Locality-sensitive hashing (LSH) serves as one of the most successful techniques for the fast nearest neighbor search over the gigantic database, which achieves a good balance between the search performance and computational efficiency using small hash codes. [2] pioneered the LSH study for the vector data with high dimension (*i.e.*, vector-to-vector search), by introducing the linear projection paradigm to efficiently generate the hash codes for the specific similarity metric. In recent years, extensive research efforts have been devoted to developing learning based hashing methods for vector-to-vector search, and have been widely adopted in many tasks including large-scale visual search [7, 22], object detection [3, 4], and machine learning [14, 15, 18].

Despite the progress of hashing research for the classic vector-to-vector nearest neighbor search, rare studies have been done on developing LSH techniques for subspace-to-subspace search. To deal with this problem, it is desirable to represent subspaces with compact signatures without losing the locality information. Compared to the vector data, the subspace data is more complex and informative, mainly because that there are massive samples distributed in one subspace, and meanwhile there also exist the inclusion relations between different subspaces, even though their dimensions may vary largely. This certainly brings great difficulties to distinguish the similar and dissimilar subspaces, especially among the large-scale high dimensional database with large cardinality.

The most well-known pioneering approach for subspace search was proposed in [1]. It maps each subspace in $\mathbb{R}^n$ to its orthogonal projection matrix, and subsequently performs the nearest neighbor search equivalently over the projection matrix. [11] further turns this matrix into a high-dimensional vector, and then applies the random projection based LSH to generate binary codes. Such a solution can preserve the angular distance among subspaces with high probability. Besides, both [1] and [11] achieved fast search with encouraging performance. However, they heavily rely on the product of the orthogonal projection matrix, which inevitably takes expensive computational cost. [23] provided a new LSH solution by simply thresholding the angular distance for linear subspace. However, at querying stage the number of possible candidates often increases explosively, subsequently degenerating the sublinear search to the exhaustive search.

In this paper, we propose a new and efficient family of locality sensitive hash for fast nearest neighbor search over linear subspaces of arbitrary dimension. It adopts the simple random projection paradigm, named random angular projection (RAP), to efficiently capture the subspace distribution characteristics by the discriminative binary codes. Specifically, RAP can largely preserve the angle of subspaces, which is a fundamental subspace similarity metric. In Figure 1 we demonstrate how RAP generates binary codes for the video sequences. To the best of our knowledge, our method is the fastest subspace LSH till now that enjoys strong locality sensitivity for linear subspaces. Theoretically and practically, we show that the proposed method can simultaneously promise discriminative and fast binary codes generation, compared to the state-of-the-arts.

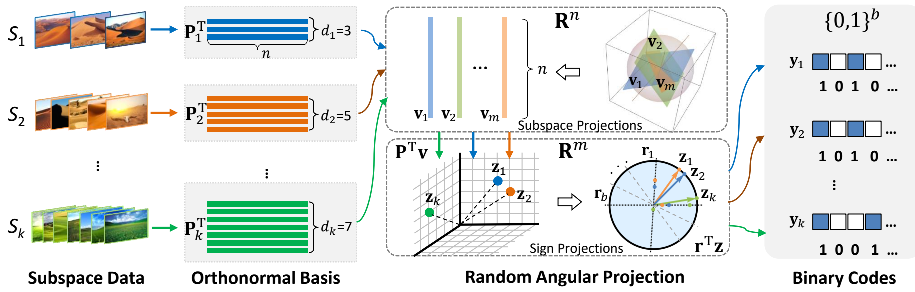


Fig. 1. Random angular projection for subspace hashing (*e.g.*, video sequences)

2 Subspace Angular Projection

Given a d-dimensional subspace $\mathrm{S} \subset \mathbb{R}^n$, our goal is to find a locality sensitive hash $\mathcal{H} : \mathbb{R}^{n \times d} \mapsto \{0,1\}^b$, *i.e.*, it can map an arbitrary d-dimensional subspace into a b-length binary code. Namely, if two subspaces are similar, $\mathcal{H}$ can encode them with similar hash codes.

2.1 Angular Distance

Before we elaborate on the detailed design of our subspace hash function, we first introduce the subspace distance metric commonly used in the literature.

Given any subspace S of d-dimension, it can be fully represented by its orthonormal basis $\mathbf{P} \in \mathbb{R}^{n \times d}$. The orthonormal basis has been proved to be a kind of good representation, which can fully characterize the feature of the subspace. Besides, it can be obtained efficiently by applying Gram-Schmidt process with unit-normalization on the given subspace.

Based on the orthonormal basis representation, the principal angle between subspaces can be defined formally. It reflects the correlations among the subspaces, and serves as an important and fundamental concept for subspaces in the literature. Given subspace S_1 and S_2 with dimension d_1 and d_2 ($d_1 \leq d_2$) in $\mathbb{R}^n$, and their orthonormal basis $\mathbf{P}_1$ and $\mathbf{P}_2$, their principal angles ξ_1, ξ_2, ..., $\xi_{d_1} \in [0, \frac{\pi}{2}]$ can be defined recursively [12]:

$$\cos(\xi_i) = \max_{\mathbf{p}_i \in \mathrm{S}_1} \max_{\mathbf{q}_i \in \mathrm{S}_2} \left(\frac{\mathbf{p}_i^\top \mathbf{q}_i}{\|\mathbf{p}_i\|_2 \|\mathbf{q}_i\|_2} \right), \tag{1}$$

subject to

$$\mathbf{p}_i^\top \mathbf{p}_j = 0 \text{ and } \mathbf{q}_i^\top \mathbf{q}_j = 0, \text{ for } j = 1, 2, \ldots, i-1.$$

Based on the principal angles, we can define the common angular distance:
Definition 1 Angular Distance *Given subspace* S_1 *and* S_2 *in* $\mathbb{R}^n$*, with dimension* d_1 *and* d_2 *(*$d_1 \leq d_2$*) respectively, the angle between* S_1 *and* S_2 *is*

$\xi_{\mathrm{S}_1,\mathrm{S}_2} = \arccos \frac{\sum_{i=1}^{d_1} \cos^2(\xi_i)}{\sqrt{d_1}\sqrt{d_2}}$, *and thus their angular distance can be defined as*

$$dist(\mathrm{S}_1, \mathrm{S}_2) = \frac{1}{\pi} \arccos \frac{\sum_{i=1}^{d_1} \cos^2(\xi_i)}{\sqrt{d_1}\sqrt{d_2}}. \tag{2}$$

The angular distance is a "true" distance metric, satisfying non-negativity, identity of indiscernibles, symmetry and triangle inequality [11]. Subsequently, we could define a metric space for all subspaces and further design a kind of locality-sensitive hash on the metric space later.

Note that usually it is complicated to directly compute the angular distance based on (1). But fortunately we have the following fact [25]:

$$\|\mathbf{P}_1^\top \mathbf{P}_2\|_F^2 = \sum_{i=1}^{d_1} \cos^2(\xi_i). \tag{3}$$

Therefore, we can simplify the computation of the angle $\xi_{\mathrm{S}_1,\mathrm{S}_2}$ by $\|\mathbf{P}_1^\top \mathbf{P}_2\|_F^2$.

2.2 Random Subspace Projection

It should be pointed out that the orthonormal basis is not unique for one subspace. To efficiently characterize a subspace, it is very critical to find an unique and compact representation for subspace data.

Suppose both $\mathbf{P}$ and $\bar{\mathbf{P}}$ are the orthonormal basis for subspace S of d-dimension, then given a unit vector $\mathbf{v} \in \mathbb{R}^n$, we have the following observation

$$\|\mathbf{P}^\top \cdot \mathbf{v}\|_F^2 = \|\bar{\mathbf{P}}^\top \cdot \mathbf{v}\|_F^2. \tag{4}$$

Inspired by this fact, we can introduce an α function that can uniquely characterize the orthonormal matrix form of subspace.

Definition 2 α **Function** *A* α *function is defined based on a unit vector* $\mathbf{v} \in \mathbb{R}^n$, *mapping a* d*-dimensional subspace* S *in* $\mathbb{R}^n$ *to a real number in* $[0, 1]$

$$\alpha_{\mathbf{v}}(\mathrm{S}) = \left\|\mathbf{P}_{\mathrm{S}}^\top \cdot \mathbf{v}\right\|_F^2 \tag{5}$$

where $\mathbf{P}_{\mathrm{S}}$ *is an orthonormal matrix of* S *and* $\|\cdot\|_F$ *denotes the Frobenius norm.*

The α function is invariant to different orthonormal matrices for the same subspace, which means that it is able to capture the unique characteristic of the subspace data.

Example 1 *Let* $\mathbf{P}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$ *and* $\bar{\mathbf{P}}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 \end{pmatrix}$ *be the orthonormal matrice of subspace* S_1 *in* $\mathbb{R}^3$, $\mathbf{P}_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$ *be the orthonormal matrix of another subspace* S_2 *in* $\mathbb{R}^3$. *Then, for a unit vector* $\mathbf{v} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right)^\top$, *we have*

$$\|\mathbf{P}_1^\top \mathbf{P}_2\|_F^2 = \|\bar{\mathbf{P}}_1^\top \mathbf{P}_2\|_F^2 = 1, \;\; with\ \xi_1 = 0, \xi_2 = \pi/2,$$

$$\alpha_{\mathbf{v}}(\mathrm{S}_1) = \|\mathbf{P}_1^\top \cdot \mathbf{v}\|_F^2 = \|\bar{\mathbf{P}}_1^\top \cdot \mathbf{v}\|_F^2 = 1,\ \alpha_{\mathbf{v}}(\mathrm{S}_2) = \|\mathbf{P}_2^\top \cdot \mathbf{v}\|_F^2 = \frac{1}{2}.$$

Based on α function, we can further define the angular projection function for subspace S of d dimension, by appending a proper basis $\alpha_0 d$,

$$z_{\mathbf{v}}(\mathrm{S}) = \alpha_{\mathbf{v}}(\mathrm{S}) + \alpha_0 d. \tag{6}$$

The projection function $z_{\mathbf{v}}$ serves as a useful and efficient representation of the subspace. Later we will show that such kind of projection induces a locality sensitive hash function for the angular distance, so that the similar subspaces will share the similar hash code. In the following theorem, we show that $z_{\mathbf{v}}$ has strong connections to the angular distance.

Theorem 1 *Given subspaces* S_1 *and* S_2 *in* $\mathbb{R}^n$*, with dimension* d_1 *and* d_2 *respectively, and*

$$\alpha_0 = \frac{\sqrt{2}}{\sqrt{n^3+2n^2}} - \frac{1}{n}, \tag{7}$$

if $\mathbf{v}$ *is a random variable vector uniformly distributed in the unit sphere of* $\mathbb{R}^n$*, then we have the following equation*

$$\mathbf{E}\left[z_{\mathbf{v}}(\mathrm{S}_1) z_{\mathbf{v}}(\mathrm{S}_2)\right] = \frac{2}{(n+2)n} \left\|\mathbf{P}_1^\top \mathbf{P}_2\right\|_F^2, \tag{8}$$

where $\mathbf{P}_1$ *and* $\mathbf{P}_2$ *are any orthonormal basis of subspace* S_1 *and* S_2*.*

Due to the space limit, we leave the proof to the supplementary material. As Equation (3) states that the angular distance between two subspaces has a close relationship with the Frobenius norm of their orthonormal matrices product. The theoretical result here further indicates that the product of their projection values can be regarded equivalent to the Frobenius norm. Namely, we have the following fact

$$\mathbf{E}\left[z_{\mathbf{v}}(\mathrm{S}_1) z_{\mathbf{v}}(\mathrm{S}_2)\right] = \frac{2}{(n+2)n} \sum_{i=1}^{d_1} \cos^2(\xi_i). \tag{9}$$

We see that the projection function $z_{\mathbf{v}}$ owns the surprising property that it can preserve the angular distances.

Example 2 *Let* $\mathbf{P}_1 = (1,0)^\top$ *and* $\mathbf{P}_2 = (\cos\gamma, \sin\gamma)^\top$ *(*γ *is constant) respectively be the orthonormal matrix of subspace* S_1 *and* S_2 *in* $\mathbb{R}^2$*, with dimension* $d_1 = d_2 = 1$*. Then, for a unit vector* $\mathbf{v} = (\cos\tau, \sin\tau)^\top$*,* $\tau \sim \mathrm{U}[0, 2\pi]$*, we have*

$$\mathbf{E}[\alpha_{\mathbf{v}}(\mathrm{S}_1)\alpha_{\mathbf{v}}(\mathrm{S}_2)] = \frac{\cos^2(\gamma)}{4} + \frac{1}{8},\ \mathbf{E}[\alpha_{\mathbf{v}}(\mathrm{S}_2)] = \frac{1}{2},\ \mathbf{E}[\alpha_{\mathbf{v}}(\mathrm{S}_1)] = \frac{1}{2}.$$

Thus, with $\alpha_0 = \frac{\sqrt{2}}{4} - \frac{1}{2}$ *and* $\xi_1 = \gamma$*,*

$$\mathbf{E}[z_{\mathbf{v}}(\mathrm{S}_1) z_{\mathbf{v}}(\mathrm{S}_2)] = \mathbf{E}[\alpha_{\mathbf{v}}(\mathrm{S}_1)\alpha_{\mathbf{v}}(\mathrm{S}_2)] + \alpha_0\left(\mathbf{E}[\alpha_{\mathbf{v}}(\mathrm{S}_1)] + \mathbf{E}[\alpha_{\mathbf{v}}(\mathrm{S}_2)]\right) + \alpha_0^2 = \frac{\cos^2(\xi_1)}{4}.$$

Practically, to completely characterize the subspace with a large probability, we can employ a series of random projection function $\{z_{\mathbf{v}_1}, z_{\mathbf{v}_2}, \ldots, z_{\mathbf{v}_m}\}$ as one composite projection function $\mathcal{Z}$ and approximate the expectation in (9) by average. Specifically, for any subspace S, we can have an m-dimensional vectorial representation $\mathbf{z}_{\mathrm{S}}$ using projection $\mathcal{Z}$:

$$\mathbf{z}_{\mathrm{S}} = \mathcal{Z}(\mathrm{S}) \doteq \left(z_{\mathbf{v}_1}(\mathrm{S}), z_{\mathbf{v}_2}(\mathrm{S}), \ldots, z_{\mathbf{v}_m}(\mathrm{S})\right)^\top . \tag{10}$$

And it is easy to see that $\lim_{m\to+\infty} \frac{1}{m}\mathbf{z}_{\mathrm{S}_1}^\top \mathbf{z}_{\mathrm{S}_2} = \mathbf{E}\left[z_{\mathbf{v}}(\mathrm{S}_1) z_{\mathbf{v}}(\mathrm{S}_2)\right]$. The projection vector enjoys both fast computation and good approximating power. When m is large enough, we can guarantee that the approximation nearly converges.

3 Locality Sensitive Subspace Hashing

The vectorial representation can be generated efficiently by random subspace projections. However, to promise a satisfying approximation to the subspace similarities, usually we have to rely on a large number of projections, forming a long vectorial representation. This inevitably brings expensive computation in the nearest subspace search. Fortunately, we have a new kind of locality sensitive hash (LSH) for subspace data, which can faithfully speed up the search by further encoding the vectorial representation into the binary codes, while preserving the locality among the subspaces.

Definition 3 Sign Random Projection *A sign random projection $f_{\mathbf{r}}(\cdot)$, mapping a vector $\mathbf{x} \in \mathbb{R}^m$ to a binary bit using $\mathbf{r} \in \mathbb{R}^m$, is defined as*

$$f_{\mathbf{r}}(\mathbf{x}) = sgn(\mathbf{r}^\top \mathbf{x}). \tag{11}$$

For any two vectors $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{r}$ is a random variable vector, uniformly distributed in the unit sphere of $\mathbb{R}^m$, then we have the following fact [6]

$$\Pr[f_{\mathbf{r}}(\mathbf{x}) = f_{\mathbf{r}}(\mathbf{y})] = 1 - \frac{1}{\pi}\arccos\left(\frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}\right). \tag{12}$$

For the subspace search problem, we could further define a metric space (Ω, dist) for all subspaces of $\mathbb{R}^n$ based on the angular distance for subspaces in Equation (2), where $\Omega = \{\mathrm{S} \subset \mathbb{R}^n | \mathrm{S} \text{ is a subspace of } \mathbb{R}^n\}$. Then, we find that the sign random projection combined with the random subspace projection function can serve as a new family of locality sensitive hash function in the metric space (Ω, dist). Namely, the following theorem holds:

Theorem 2 *When m is large enough, $\mathcal{H} = f_{\mathbf{r}} \circ \mathcal{Z}$ is a locality sensitive hash defined on $(\Omega, dist)$ approximately, i.e., $\mathcal{H}$ is $(\delta, \delta(1+\epsilon), 1-\delta, 1-\delta(1+\epsilon))$-sensitive with $\delta, \epsilon > 0$.*

Now we have our random angular projection (RAP) based LSH function, and the theoretical result indicates that the composite function $\mathcal{H}$ satisfies the locality sensitivity to the angular distance, when using a large number of angular projections. The LSH function for subspace uses the following two steps: first

Table 1. Comparison of collision probability p and the complexity t of encoding a subspace from orthonormal basis to a b-length binary code.

	RAP	BSS	GLH
p	$1-\delta$	$1-\delta$	-
t	$O(bm+dmn)$	$O(bn^2)$	$O(bdn)$

projecting the subspace into a high-dimensional vector, and second encoding the vector into a binary code. Table 1 compares the locality sensitivity and time complexity of state-of-the-art solutions. The time complexity of RAP is linear to the space dimension n, which largely reduces the computation, compared to $O(n^2)$ of existing subspace hashing methods [11].

With the LSH property, following [10, 13] we can theoretically guarantee the sublinear search time based on the binary codes generated for all subspaces.

Theorem 3 *Given a subspace database $\mathcal{D}$ with N subspaces in $\mathbb{R}^n$, for a subspace query S, if there exists a subspace S^* such that $dist(\mathrm{S},\mathrm{S}^*) \le \delta$, then with $\rho = \frac{\ln p_1}{\ln p_2}$ (1) using N^ρ hash tables with $\log_{1/p_2} N$ hash bits, the random angular hash of an even order is able to return a database subspace $\bar{\mathrm{S}}$ such that $dist(\mathrm{S},\bar{\mathrm{S}}) \le \delta(1+\epsilon)$ with probability at least $1-\frac{1}{c}-\frac{1}{e}$, $c \ge 2$; (2) the query time is sublinear to the entire data number N, with $N^\rho \log_{1/p_2} N$ bit generations and cN^ρ pairwise distances computation.*

4 Experiments

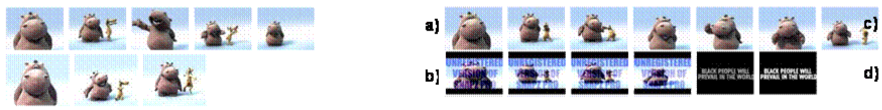
Subspace search has been widely involved in many applications, including face recognition, video search, action recognition, etc. To comprehensively evaluate our method RAP, here we choose the two popular tasks: face recognition and video search. For more experimental results on gesture recognition, please refer to the supplementary material. We compare our RAP method with the state-of-the-art subspace hashing methods, including BHZM [1], Binary Signatures for Subspaces (BSS) [11] and Grassmannian-based Locality Hashing (GLH) [23]. All these methods rely on the random way to generate hash codes. We also adopt several state-of-the-art unsupervised vector hashing methods as the baselines, including the random locality sensitive hashing (LSH) [2], and the learning based methods: iterative quantization (ITQ) [7], dubbed discrete proximal linearized minimization (DPLM) [20] and adaptive binary quantization (ABQ) [16].

4.1 Face Recognition

Face images with fixed pose under different illumination conditions lie near linear subspaces of low dimension [11]. Therefore, the face recognition in this scenario can be regarded as a nearest subspace search, where a set of face images belonging to the same person can be easily represented by a subspace.

Dataset: The two popular datasets are adopted: the Extended Yale Face Database B [5] containing 38 individuals each with about 64 near frontal images

(a) The Extended Yale Face Database B



(b) UQ_VIDEO database

Fig. 2. The examples of face recognition and video retrieval datasets

under different illuminations, and PIE database [21] consisting of 68 individuals each with 170 images under 5 near frontal poses, 43 different illumination conditions and 4 different expressions. On PIE database, we choose the frontal face images (C26 and C27). On the Extended Yale Face Database B, we randomly choose 30 images per individual as the database set, and 11 images from the rest images as the testing set. Similarly, on the PIE database the database and testing set contain 20 and 15 images respectively per individual.

Experiment Setup: Each face image is represented by a $1,024$-length vector, and thus images of the same individual will constitute a subspace in $\mathbb{R}^{1024}$. For each individual in Yale Face, we extract a representative subspace of dimension $d = 9$ ($d = 14$ for PIE datasets) by taking the top principal components as BSS did. For the testing set we generate the query subspaces with different dimensions $d_q = 7, 9, 11$ on Yale Face and $d_q = 13, 14, 15$ on PIE respectively. For the vector hashing methods, without reduction we directly utilize the original face vectors.

Results: Table 2 and 3 list the precision of the top-1 results and search time of different methods for the face recognition task. Here, the search time consists of two parts: the encoding time and the retrieval time per query subspace. In the table, the learning based vector hashing methods outperform the random LSH method. However, they heavily rely on the training process, which requires extra computational cost (see the train time). Moreover, we can see that the random subspace hashing methods RAP and BSS can perform much better than all vector hashing methods. This is because that the vector hashing, considering the vector data independently, can hardly capture the intrinsic distributions in the subspace. In all cases, our RAP achieves close, sometimes even better performance to BSS, which is consistent with our analysis in Table 1 that RAP and BSS share the same locality sensitivity power to the angular distance. As to the query time, RAP can get up to $16\times$ speedup, compared to BSS.

Figure 3 (a) and (b) further investigate the effect of the code length b on the precision performance and the speed acceleration (*i.e.*, speedup). Here, we attempt to simulate the real world case by randomly choosing d and d_q of each subspace from $\{3, 5, 7\}$. Here, since GLH works in a different way from BSS and RAP, its performance isn't affected by the code length. In the figures, we

Table 2. Face recognition performance on Yale Faces

	Methods	Precision (%), $b = 512, d = 9, m = 10^4$			Search Time (s)		
		$d_q = 7$	$d_q = 9$	$d_q = 11$	$d_q = 7$	$d_q = 9$	$d_q = 11$
Yale Face	LSH	45.84 ± 3.49			0.0025		
	ITQ	47.89 ± 3.86			0.0016 (train time 5.73)		
	DPLM	46.84 ± 5.10			0.0022 (train time 0.49)		
	ABQ	58.42 ± 1.05			0.0028 (train time 22.26)		
	BHZM	14.21 ± 5.41	16.31 ± 8.55	16.84 ± 3.93	0.0080	0.0080	0.0087
	BSS	74.21 ± 8.05	72.63 ± 4.27	73.68 ± 3.72	0.1349	0.1349	0.1219
	RAP	72.10 ± 6.13	**73.10** ± 8.58	**82.10** ± 6.09	0.0078	0.0078	0.0084

Table 3. Face recognition performance on PIE Faces.

	Methods	Precision (%), $b = 1024, d = 14, m = 10^4$			Search Time (s)		
		$d_q = 13$	$d_q = 14$	$d_q = 15$	$d_q = 13$	$d_q = 14$	$d_q = 15$
PIE Face	LSH	57.64 ± 2.35			0.0075		
	ITQ	52.64 ± 1.44			0.0077 (train time 13.48)		
	DPLM	79.41 ± 2.07			0.0079 (train time 6.82)		
	ABQ	81.47 ± 1.99			0.0137 (train time 3.17)		
	BHZM	38.82 ± 4.97	41.76 ± 2.72	37.94 ± 5.76	0.0166	0.0175	0.0186
	BSS	97.05 ± 2.63	95.58 ± 2.27	95.29 ± 2.35	0.1252	0.1274	0.1259
	RAP	96.17 ± 0.72	**96.47** ± 0.72	**97.35** ± 2.35	0.0089	0.0111	0.0114

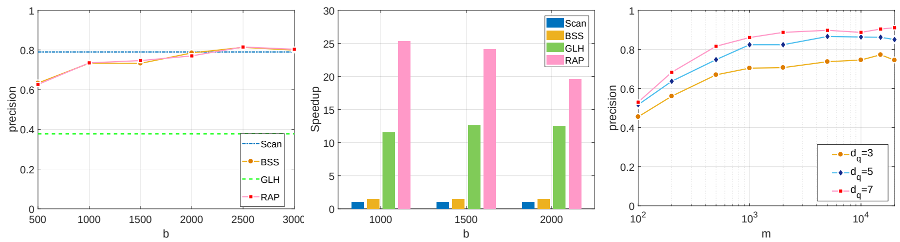
also show the results using exhaustive search ('Scan'). From the figures, we can see that all the subspace hashing methods can obviously improve the search efficiency, and our RAP gains close performance to BSS, due to the same collision probability. Moreover, with more hash bits, RAP is able to approximate or even outperform exhaustive search strategy, but completing the search at a much higher speed (more than 10× speedup over BSS and Scan).

Figure 3 (c) shows RAP performance with respect to the dimension m of the subspace projection vector (here we set $b = 2,000$). In general, a larger m promises better performance, which is consistent with our approximation analysis in Section 2.2. We can see that $m > 500$ is sufficiently large for a satisfying performance in practice.

4.2 Near-duplicated Large-Scale Video Retrieval

Next, we evaluate the performance in the video retrieval task. Here, traditional vector hashing methods individually encode each video frame into a binary code, but neglect the temporal relations among frames. The subspace hashing can address this problem by treating the consecutive frames as one subspace.

Dataset: The UQ_VIDEO [22] is one of largest databases for near duplicate video retrieval. It contains 169,952 videos in total and 3,305,525 key frames extracted from these videos. The database provides 24 selected query videos, each of which have a number of near duplicate videos, and other videos serve as



(a) Precision w.r.t. b (b) Speedup w.r.t. b (c) RAP Precision w.r.t. m

Fig. 3. The effect of different b and m on Yale Face. To simulate the real world scenarios, we adopt random d and d_q in (a) and (b), and random d in (c).

Table 4. Video retrieval performance on UQ_VIDEO.

Methods	MAP (%), $d = d_q = 5$			Search Time (s)		
	$b = 128$	$b = 256$	$b = 512$	$b = 128$	$b = 256$	$b = 512$
LSH	30.73 $\pm$1.01	30.20 $\pm$0.34	31.53 $\pm$0.06	5.5660	7.5578	16.7660
ITQ	28.92 $\pm$0.30	-	-	5.3990	-	-
DPLM	34.09 $\pm$1.60	37.24 $\pm$0.95	39.15 $\pm$0.91	5.0560	6.9249	16.8364
ABQ	25.39 $\pm$0.29	24.72 $\pm$0.04	25.92 $\pm$0.23	9.6779	17.3563	29.2905
BSS	36.91 $\pm$3.57	50.32 $\pm$0.10	57.47 $\pm$1.32	1.2842	1.2261	1.7796
RAP	**37.48** $\pm$1.60	**51.50** $\pm$0.04	**58.12** $\pm$2.20	1.2688	1.2444	1.7593

background videos. We use the provided global color histogram feature with 162 dimensions as the input feature for each frame.

Experiment Setup: We first construct the database set with all the videos except those with very few frames, forming a database set with nearly 3,000,000 key frames. For each video, we use every $d = 5$ consecutive frames without overlap to form a subspace. At the search stage, we use the Hamming distance ranking strategy to rank all the video clips in the database set according to the averaged Hamming distances to the query.

Results: Table 4 shows the mean average precision (MAP) and search time using Hamming distance ranking in the video search task. It is easy to see that the subspace hashing methods perform much better than the state-of-the-art vector hashing methods. Besides, when using different number of hash bits, our RAP method can consistently achieve the best performance. Note that here we only adopt features of 162 dimensions, and thus RAP and BSS spend similar time cost on the search. But in practice, as the feature dimension increases RAP's speedup will become much more significant (see the speedup in Figure 3(b) and the analysis in Table 1). Moreover, in this large-scale task, RAP obtains more than 5$\times$ faster than the vector hashing methods.

5 Conclusions

This paper mainly concentrated on the fast nearest neighbor search over massive subspace data, and proposed a new family of locality sensitive hashing method named random angular projection (RAP) for linear subspaces of arbitrary dimension. RAP encodes the subspace by random subspace projections and binary quantization, and theoretically enjoys fast computation and strong locality sensitivity. Our experiments in several popular tasks show that the proposed RAP method, obtaining state-of-the-art performance, can be a promising and practical solution to the nearest subspace search.

References

1. Basri, R., Hassner, T., Zelnik-Manor, L.: Approximate nearest subspace search. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2), 266–278 (2011)
2. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: ACM Symposium on Theory of Computing. pp. 380–388 (2002)
3. Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H., et al.: Fast and accurate image matching with cascade hashing for 3d reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2014)
4. Dean, T., Ruzon, M.A., Segal, M., Shlens, J., Vijayanarasimhan, S., Yagnik, J.: Fast, accurate detection of 100,000 object classes on a single machine. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1814–1821 (2013)
5. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 643–660 (2001)
6. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM 42(6), 1115–1145 (1995)
7. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(12), 2916–2929 (2013)
8. Hong, R., Hu, Z., Wang, R., Wang, M., Tao, D.: Multi-view object retrieval via multi-scale topic models. IEEE Transactions on Image Processing 25(12), 5814–5827 (Dec 2016)
9. Hong, R., Zhang, L., Tao, D.: Unified photo enhancement by discovering aesthetic communities from flickr. IEEE Transactions on Image Processing 25(3), 1124–1135 (2016)
10. Jain, P., Vijayanarasimhan, S., Grauman, K.: Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning. In: Advances in Neural Information Processing Systems, pp. 928–936 (2010)
11. Ji, J., Li, J., Tian, Q., Yan, S., Zhang, B.: Angular-similarity-preserving binary signatures for linear subspaces. IEEE Transactions on Image Processing 24(11), 4372–4380 (2015)
12. Knyazev, A.V., Argentati, M.E.: Principal angles between subspaces in an a-based scalar product: algorithms and perturbation estimates. SIAM Journal on Scientific Computing 23(6), 2008–2040 (2002)
13. Liu, W., Wang, J., Mu, Y., Kumar, S., Chang, S.F.: Compact hyperplane hashing with bilinear functions. International Conference on Machine Learning pp. 1–8 (2012)

14. Liu, X., Fan, X., Deng, C., Li, Z., Su, H., Tao, D.: Multilinear hyperplane hashing. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5119–5127 (2016)
15. Liu, X., He, J., Deng, C., Lang, B.: Collaborative hashing. In: IEEE International Conference on Computer Vision. pp. 2139–2146 (2014)
16. Liu, X., Li, Z., Deng, C., Tao, D.: Distributed adaptive binary quantization for fast nearest neighbor search. IEEE Transactions on Image Processing 26(11), 5324–5336 (2017)
17. Marques, M., Stošić, M., Costeira, J.: Subspace matching: Unique solution to point matching with geometric constraints. In: IEEE International Conference on Computer Vision. pp. 1288–1294 (2009)
18. Mu, Y., Hua, G., Fan, W., Chang, S.F.: Hash-svm: Scalable kernel machines for large-scale visual classification. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 979–986 (2014)
19. Peng, C., Kang, Z., Cheng, Q.: Subspace clustering via variance regularized ridge regression. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2931–2940 (2017)
20. Shen, F., Zhou, X., Yang, Y., Song, J., Shen, H.T., Tao, D.: A fast optimization method for general binary code learning. IEEE Transactions on Image Processing 25(12), 5610–5621 (2016)
21. Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression (pie) database. In: Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on. pp. 53–58 (2002)
22. Song, J., Yang, Y., Huang, Z., Shen, H.T., Hong, R.: Multiple feature hashing for real-time large scale near-duplicate video retrieval. In: ACM on Multimedia Conference. pp. 423–432 (2011)
23. Wang, X., Atev, S., Wright, J., Lerman, G.: Fast subspace search via grassmannian based hashing. In: IEEE International Conference on Computer Vision. pp. 2776–2783 (2013)
24. Yang, X., Wang, M., Hong, R., Tian, Q., Rui, Y.: Enhancing person re-identification in a self-trained subspace. ACM Transactions on Multimedia Computing, Communications, and Applications 13(3), 27 (2017)
25. Zhang, F.: Matrix theory: basic results and techniques. Springer Science & Business Media (2011)
26. Zhou, L., Bai, X., Liu, X., Zhou, J.: Binary coding by matrix classifier for efficient subspace retrieval. In: ACM International Conference on Multimedia Retrieval. pp. 1–8 (2018)