IPSJ Transactions on Computer Vision and Applications Vol.5 40–44 (July 2013)

[DOI: 10.2197/ipsjtcva.5.40]

Express Paper

Sparse Isotropic Hashing

Ikuro Sato[1,a)] Mitsuru Ambai[1,b)] Koichiro Suzuki[1,c)]

Received: March 11, 2013, Accepted: April 24, 2013, Released: July 29, 2013

Abstract: This paper address the problem of binary coding of real vectors for efficient similarity computations. It has been argued that orthogonal transformation of center-subtracted vectors followed by sign function produces binary codes which well preserve similarities in the original space, especially when orthogonally transformed vectors have covariance matrix with equal diagonal elements. We propose a simple hashing algorithm that can orthogonally transform an arbitrary covariance matrix to the one with equal diagonal elements. We further expand this method to make the projection matrix sparse, which yield faster coding. It is demonstrated that proposed methods have comparable level of similarity preservation to the existing methods.

Keywords: binary codes, nearest neighbor search, local descriptors, sparse matrix

1. Introduction

Searching a nearest neighbor point from a massive dataset for a given query demands computational resources in both processing time and data storage. Binary coding of feature vectors increasingly gains attention from researchers of machine learning and of computer vision. Advantage of binary codes is two-fold: 1) similarity between a pair of points can be quickly computed by taking the Hamming distance, defined as the sum of trues after element-wise XOR operation, and 2) storage space can be reduced by some order of magnitude. Therefore, use of binary codes can potentially solve the above-mentioned problems, as long as the similarity relations among binary codes resemble the ground truth.

In general there are two types of methods for binary coding. One type uses hash functions to real feature vectors computed in advance [5], [7], [8], [9], [10], [11], [13], [14]. Our work belongs to this type. The other type directly designs binary features without computing real vectors in the middle steps. In this direct approach the designing is problem-specific. It has been successfully applied in designing binary local descriptors of images [1], [2], [3], [4].

The goal in this work is to give a hash function so that the generated binary codes well preserve, for a given code length, the similarity relations in the original space.

1.1 Related Works

Isotropic Hashing (IsoHash) [5] is most related to our work. They claim that orthogonally transforming the original covariance matrix to the one with equal diagonal elements is important because information loss on each axis is equilibrated when binarized. Iterative Quantization (ITQ) [8] uses an orthogonal transformation so as to minimize the quantization error. These two methods are the current state-of-the-art, as long as linear transformation is considered. Spectral Hashing [7] uses partitioned eigenvectors to learn the hashing function. Sequential Projections [9] provides semi-supervised approach using labeled data to learn projection matrix. Sparse random projections [12], [13] use a simple projection matrix whose elements only contains $\{1, 0, -1\}$ for faster coding. A supervised approach is used in [14] to retrieve semantically similar points.

[1] Denso IT Laboratory, Inc., Shibuya, Tokyo 150–0002, Japan
[a)] isato@d-itlab.co.jp
[b)] manbai@d-itlab.co.jp
[c)] ksuzuki@d-itlab.co.jp

1.2 Contribution

We develop an algorithm to generate a hash function that has a sparse, real projection matrix which transform the original covariance matrix to the one with equal diagonal elements. As shown in Section 2, problem of making isotropic variances by orthogonal transformation is highly under-constrained. It means IsoHash does not exploit all the degrees of freedom in parameter space. More constraints can be introduced to the problem, such as supervision or sparsity. Our method, Sparse Isotropic Hashing, adds sparsity constraint to the IsoHash. Our approach finds a solution strictly resided in the $SO(N)$ group by tuning Euler angles of high dimensions, and requires no external solvers.

2. Sparse Isotropic Hashing

2.1 Problem Statement

Suppose we are given data matrix $X_0 = [x_0^1, x_0^2, \cdots, x_0^p] \in \mathbb{R}^{n\times p}$, where p data points are represented as column vectors of length n. We consider two-step coding following many of the previous works: 1) applying linear transformation to produce $Y = [y^1, y^2, \cdots, y^p] \in \mathbb{R}^{m\times p}$, and 2) applying element-wise step function to produce m-dimensional binary codes $B = [b^1, b^2, \cdots, b^p] \in \{0, 1\}^{m\times p}$, where

$$b_j^q = \theta(y_j^i), y^q = W^T(x_0^q - c), c \in \mathbb{R}^n. \quad (1)$$

The function θ is defined as $\theta(a) = 1, a \geq 0$ and $\theta(a) = 0$ otherwise. Problem is to find the projection matrix $W \in \mathbb{R}^{n\times m}$ and

$c \in \mathbb{R}^n$ so that

a) Euclidean distance among Y preserve those among X_0,

b) Y has isotropic variances, and

c) the projection matrix W is sparse for faster coding.

2.2 Extension of Euler Angles to *n*-Dimensions

Here we consider the optimization of the projection matrix W in Eq. (1). To satisfy the condition a), many authors have adopted to use orthogonal matrices since they leave Euclidean distance invariant. In other words, it is a problem of finding a matrix $V \in \mathbb{R}^{n\times n}$ from the *orthogonal group* $O(n)$ to satisfy isotropic variance condition [5], or to minimize the quantization error [8]. The matrix W is then simply constructed by picking m columns from V.

We claim that use of an element from the *special orthogonal group* $SO(n)$ suffices these purposes. Since $O(n)$ differs from $SO(n)$ only in a way that the former has an additional degree of freedom to reverse the direction of an axis, use of $O(n)$ gives no advantage in similarity-preservation compared to $SO(n)$. So we adopt to use $SO(n)$ in this work.

The $SO(n)$ is a Lie-group, since its elements have derivatives of arbitrary order. It is known that any rotation in the n-dimension can be represented in an exponential form,

$$R(\alpha) = \exp\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \alpha_{ij} M_{ij}\right), \tag{2}$$

where M_{ij} is a *matrix* of size $n \times n$ defined as

$$\left(M_{ij}\right)_{kl} = -\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}. \tag{3}$$

In words, all elements of the matrix M_{ij} are zero except for (i, j)-element (-1) and for (j, i)-element (1) [*1]. Note from Eq. (2) that there are $\frac{n(n-1)}{2}$ terms in the exponent. We call α the Euler angles extended to n-dimension in this work, since they are related to the (standard) Euler angles when $n = 3$.

Problem of finding the projection W is now equivalent to finding the set of Euler angles α satisfying isotropic-variance condition b) and sparsity condition c).

2.3 Cost Function

We seek n-dimensional Euler angles that comprise $\frac{n(n-1)}{2}$ parameters. On the other hand, the condition of the isotropic variances only provides $m - 1$ constraints, i.e., variance in the first coordinate equals to the variance in the i-th coordinate $\forall i = 2, 3, \cdots, m \le n$. Thus the problem of satisfying condition a) and b) is under-constrained when $n \ge 3$, i.e., there exist infinitely many rotations that generate isotropic variances. It is natural to impose additional constraints to the problem. In this work we impose sparseness in W, to make the problem over-constrained.

Let $X = [x^1, x^2, \cdots, x^p]$, where $x^j = x_0^j - \langle x_0 \rangle$ ($\langle\cdot\rangle$ represents the expectation value over the data point distribution). The covariance matrix in the original space is $C = \frac{1}{p}XX^T$. Our minimization problem is formulated as follows.

$$R^* = \underset{R\in SO(n)}{\operatorname{argmin}}\, L_0(R) + \eta L_1(R), \tag{4}$$

$$L_0(R) = \frac{1}{4\overline{\lambda}^2}\sum_{i=1}^{m}\left(\sum_{j,k=1}^{n} R_{ji}^T C_{jk} R_{ki} - \overline{\lambda}\right)^2, \tag{5}$$

$$L_1(R) = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|R_{ji}\right|. \tag{6}$$

In Eq. (5), $\overline{\lambda} = \frac{1}{m}\sum_{i=1}^{m}\lambda_i$, where $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$ are the eigenvalues of C. Note that $R \in SO(n)$ satisfies the condition a), L_0 minimization satisfies the condition b), and L_1 minimization satisfies the condition c).

2.4 Algorithm

It is a difficult problem to find the n-dimensional Euler angles α that globally minimize the cost function in Eqs. (4), (5), (6) because they are non-linear function of α. We consider an iterative process to search α. Suppose α is the solution to Eqs. (4), (5), (6). We break down the solution to the sum of many infinitesimal angles, $\alpha_{ij} = \sum_t^T \delta\alpha_{ij}^{(t)}$, where $\delta\alpha_{ij}^{(t)} \ll 1$ and $T \gg 1$. The essential idea in our algorithm is to update the Euler angles by $\delta\alpha^{(t)}$ at t-th iteration so that the cost $L_0 + \eta L_1$ monotonically decreases.

The rotaion in Eq. (2) can be rewritten as

$$\begin{aligned} R &= \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta\alpha_{ij}^{(t)} M_{ij}\right) \\ &= \prod_{t=1}^{T}\left(\mathbb{I}_n + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta\alpha_{ij}^{(t)} M_{ij}\right) \\ &= \prod_{t=1}^{T}\left(\mathbb{I}_n + \mathcal{D}^{(t)} + C^{(t)} + \mathcal{I}^{(t)}\right) \end{aligned} \tag{7}$$

where $\mathbb{I}_n$ is the $n \times n$ identity, and $\mathcal{D}, C$ and $\mathcal{I}$ are the *direct*, *cross* and *irrelevant* terms, respectively, defined as

$$\begin{aligned} \mathcal{D}^{(t)} &= \sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\delta\alpha_{ij}^{(t)} M_{ij}, C^{(t)} = \sum_{i=1}^{m}\sum_{j=m+1}^{n}\delta\alpha_{ij}^{(t)} M_{ij}, \\ \mathcal{I}^{(t)} &= \sum_{i=m+1}^{n-1}\sum_{j=i+1}^{n}\delta\alpha_{ij}^{(t)} M_{ij}. \end{aligned} \tag{8}$$

Among these three terms above, the direct and the cross terms affect L_0 and L_1; therefore, corresponding set of α_{ij} (totally $\frac{m(m-1)}{2} + n(n-m)$ parameters) needs to be optimized. The irrelevant terms affect neither L_0 nor L_1. There is no need to update $\alpha_{ij}, \forall i = m+1, \cdots n-1, j = i+1, \cdots n$. So we simply drop the irrelevant terms from Eq. (7) and factorize it into two rotations as follows

$$\begin{aligned} R &= \prod_{t=1}^{T} R^{(t)}, R^{(t)} = R_{\mathcal{D}}^{(t)} R_C^{(t)}, \\ & R_{\mathcal{D}}^{(t)} \equiv \mathbb{I}_n + \mathcal{D}^{(t)}, R_C^{(t)} \equiv \mathbb{I}_n + C^{(t)}. \end{aligned} \tag{9}$$

By dropping the irrelevant terms the dimension of the parameter space is reduced.

Let us first explain how to optimize the cross terms by gradient-descent approach. From here on we drop (t) except when we intend to stress it. By substituting R_C in Eq. (9) into Eq. (5), one obtains with straightforward algebra

[*1] Note from Eq. (3) that matrices M_{ij} are linearly independent for $j > i$ and anti-symmetric (thus traceless). Such matrices are generators of $SO(n)$.

Algorithm 1 Sparse Isotropic Hashing (SIH)

Input: $C \in \mathbb{R}^{n\times n}$ $(C^T = C), \eta \in \mathbb{R}, m \le n, T \in \mathbb{N}, \epsilon \ll 1, r \ll 1$

1 Compute $\overline{\lambda} = \frac{1}{m}\sum_{i=1}^{m}\lambda_i$, where λ_i's are the eigenvalues of C in decreasing order.

2 set $R = \mathbb{I}_n$.

3 **for** $t = 1, 2, \cdots, T$
  for $i = 1, 2, \cdots, m$
    for $j = m+1, m+2, \cdots, n$
      compute $\frac{\delta L_0}{\delta \alpha_{ij}}$ according to Eq. (10).
      compute: $\frac{\delta L_1}{\delta \alpha_{ij}} = \frac{1}{n}\sum_k \mathrm{sgn}(R_{ki})R_{kj}$
      compute: $\delta\alpha_{ij} = -\epsilon\left(\frac{\delta L_0}{\delta \alpha_{ij}} + \eta\frac{\delta L_1}{\delta \alpha_{ij}}\right)$
      compute R_C according to Eqs. (8), (9).
    end for
    for $j = i+1, i+2, \cdots, m$
      compute $\frac{\delta L_0}{\delta \alpha_{ij}}$ according to Eq. (11).
      compute: $\frac{\delta L_1}{\delta \alpha_{ij}} = \frac{1}{n}\sum_k(\mathrm{sgn}(R_{ki})R_{kj} - \mathrm{sgn}(R_{kj})R_{ki})$
      compute: $\delta\alpha_{ij} = -\epsilon\left(\frac{\delta L_0}{\delta \alpha_{ij}} + \eta\frac{\delta L_1}{\delta \alpha_{ij}}\right)$
      compute $R_{\mathcal{D}}$ according to Eqs. (8), (9).
    end for
  end for
  $R \leftarrow RR_{\mathcal{D}}R_C$
  $R \leftarrow UV^T$, $U(V)$ is the left(right)-singular vectors of R
  $C \leftarrow R^TCR$
end for

4 $W \in \mathbb{R}^{n\times m}$, $W_{ij} = \begin{cases} R_{ij}, & |R_{ij}| > r \\ 0, & \text{otherwise,} \end{cases}$.

Output: W

$$L_0(R_C) = \frac{1}{4\overline{\lambda}^2}\sum_{i=1}^{m}\left(C_{ii} + \sum_{j=m+1}^{n}\delta\alpha_{ij}\left[M_{ij}, C\right]_{ii} - \overline{\lambda}\right)^2,$$

where the square bracket is the matrix commutator, $[A, B] = AB - BA$. The first derivative of L_0 with respect to α_{ij} (in the cross terms) at zeros is

$$\left.\frac{\delta L_0}{\delta \alpha_{ji}}\right|_{\substack{\alpha_{ij}=0,\\ i=1,..,m,\\ j=m+1,..,n}} = \frac{\left(C_{ii} - \overline{\lambda}\right)C_{ij}}{\overline{\lambda}^2}. \tag{10}$$

Next we explain how to optimize the direct terms. Since $R_{\mathcal{D}}$ has nonzero elements only in first m-coordinates, sum of the first m diagonal elements in the covariance C does not "leak" to the complementary space, i.e., $\sum_{i=1}^{m}(R_{\mathcal{D}}^TC'R_{\mathcal{D}})_{ii} = \sum_{i=1}^{m}C'_{ii}$. Here C' is a covariance after arbitrary rotation ($C' \equiv R^TCR$). Knowing this, it is not very difficult to show that L_0 is minimized for given C', if $\left[(\prod_{s=1}R_{\mathcal{D}}^{(s)})^TC'(\prod_{s=1}R_{\mathcal{D}}^{(s)})\right]_{ii} = \frac{1}{m}\sum_{i=1}^{m}C'_{ii}$ is met, and such series of $R_{\mathcal{D}}^{(s)}$ does exist according to the Schur-Horn Theorem [5], [6]. We skip the rest of algebraic details and only show the first derivative of L_0 for the direct terms

$$\left.\frac{\delta L_0}{\delta \alpha_{ji}}\right|_{\substack{\alpha_{ij}=0,\\ i=1,..,m-1,\\ j=i+1,..,m}} = \frac{\left(C_{ii} - C_{jj}\right)C_{ij}}{\overline{\lambda}^2}. \tag{11}$$

We summarize the proposed algorithm in Algorithm 1. It computes $SO(n)$ group element R which locally minimizes the l_1-norm and makes nearly isotropic variances. The projection matrix W is constructed by taking the first m columns of R, and setting (i, j) elements zero if $R_{ij} \simeq 0$. We omit the derivation of $\frac{\delta L_1}{\delta\alpha}$ due to the limited space.

3. Experiment

3.1 Methods

Following unsupervised binary coding methods are tested.

SIH(η): The proposed Sparse Isotropic Hashing.

IH-LP [5]: The Isotropic Hashing (Lift and Projection).

IH-GF [5]: The Isotropic Hashing (Gradient Flow).

PCA-ITQ [8]: PCA is taken to X, and the eigenvectors of the m-largest eigenvalues are orthogonally transformed to minimize quantization error.

PCA-RR [8]: A randomly-generated $O(m)$ element is multiplied to the matrix created by the PCA step.

PCA [8], [9]: The PCA step is used to generate W.

RP [10], [11]: Random Projections. W_{ij} is a random number set by normal distribution with zero mean.

VSRP [13]: Very Sparse Random Projection. W_{ij} is set to 1, 0, or -1 with probabilities $\frac{1}{2\sqrt{n}}$, $1 - \frac{1}{\sqrt{n}}$, $\frac{1}{2\sqrt{n}}$, respectively. Sparseness, the ratio of the number of nonzero elements to the total number of elements in W, is about 0.9, when $n \simeq 100$.

All the methods use Eq. (1) with $c = \langle x_0\rangle$ for binary coding.

3.2 Dataset

We use the same image dataset used in Ref. [14]. The dataset consists of 42 images, on which the front pages of 7 magazines are taken at 6 different camera poses. Let $u = 1, \cdots, 6$ denote particular camera poses. Images are divided into 3 sets: $u = 1$ being the query set, $u = 2, 3, 4$ being the training set, and $u = 5, 6$ being the test set.

Real, local descriptors are extract by SIFT ($n = 128$) [15] and by CARD ($n = 136$) [1] (without binary coding) from the images. There are 17917, 67009, and 33774 SIFT descriptors from the query, training, and test sets, respectively. There are 12896, 50053, and 25238 CARD descriptors.

3.3 Evaluation Criteria

These local descriptors are transformed to binary codes by methods listed in Section 3.1. Quality of similarity preservation is examined for each method in terms of mean Average Precision (mAP), which is commonly used as an indicator of performance [5], [7], [8].

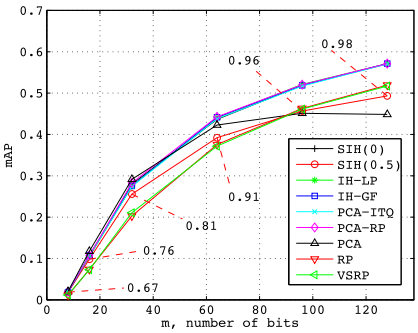
Following steps are proceeded to compute mAP.

1. For a given query, use real vectors to compute Euclidean distance to each point in the test set, and mark 5 closest points. They are the ground truth.
2. For the query, use the binary codes to compute Hamming distances to each point in the test set.
3. For each of the marked points, calculate the precision at which the marked point is just retrieved. Take the average of all 5 (Average Precision).
4. Compute mean value of the Average Precisions over the entire query set (mAP).

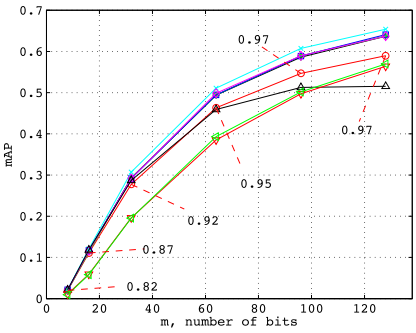
3.4 Results

3.4.1 mAP

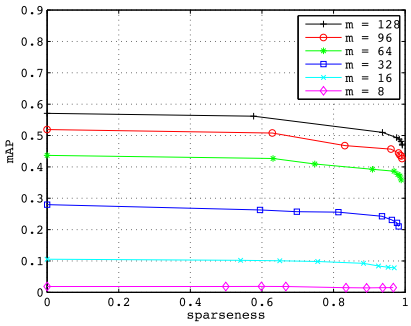
The mAP is plotted in **Fig. 1** (a, b). The Isotropic Hash-



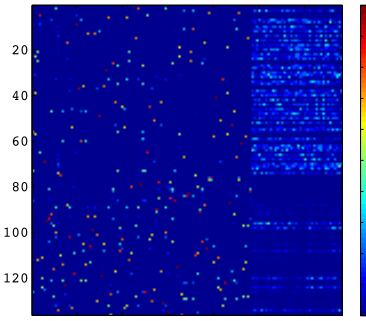
(a) mAP vs m (# bits) (SIFT)



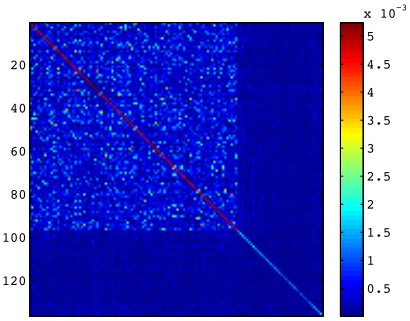
(b) mAP vs m (# bits) (CARD)



(c) mAP vs sparseness (SIFT)



(d) $|R|$ (CARD, $m = 96, \eta = 0.5$)



(e) $|R^T CR|$ (CARD, $m = 96, \eta = 0.5$)

Fig. 1 (a, b): mAP vs m. Numbers in the figures denote the sparseness of SIH(0.5). (c): mAP vs the sparseness. (e): Visualization of the rotational matrix, produced by our method. (d): Visualization of the covariance matrix after the rotation.

ing methods (SIH(0), IH-LP, IH-GF), PCA-ITQ, and PCA-RP clearly perform better than the rest of the methods. These high-performing methods exhibit very similar results for SIFT data so that their mAP curves look almost on top of each other. For CARD data, PCA-ITQ exhibits slightly better results among these methods, but the difference in mAP is no larger than 0.02 among top-3 methods. We show the top-3 methods with mAP values (in square brackets) for $m = 96$ in the table below.

	SIFT	CARD
1st	IH-LP [**0.5211**]	PCA-ITQ [**0.6075**]
2nd	PCA-RP [0.5204]	IH-LP [0.5894]
3rd	SIH(0) [0.5191]	PCA-RP [0.5894]

The author of Ref. [5] states that an isotropic hashing performs equally well as the state-of-the-art anisotropic hashing such as PCA-ITQ. Our results support this statement.

3.4.2 Sparseness

Next we examine the sparsity level of the proposed method. Sparseness is a crucial feature for real-time applications because it requires small amount of sum-of-products in binary coding. In our algorithm, sparseness can be tuned by changing η. At $\eta = 0.5$ and $m = 96$, W has sparseness of 0.97 for CARD data (see Fig. 1 (b, d)). Such a sparse W enables much faster coding than dense W. It performs better than dense hashing such as PCA, albeit very sparse. Making the matrix W sparse degrades mAP; however, the amount of degradation is quite moderate. From Fig. 1 (c), it can be read that mAP is quite flat for sparsity $\lesssim 0.8$, and degrades rapidly, say for sparsity $\gtrsim 0.95$. Despite the fact that W is very sparse, the resultant covariance has nearly identical diagonal elements (see Fig. 1 (e)). In the case of the CARD data with $m = 96$, the ratio of the standard deviation of the first m diagonal elements to $\bar{\lambda}$ is only 3.5%. One can tune η to balance the sparsity, related to coding speed, and the uniqueness of the diagonal elements, related to the mAP.

4. Conclusion

We revealed that the problem of finding a projection matrix that projects real vectors to the others so that their variances are identical is highly under-constrained. Therefore, there exist infinitely many solutions to this problem. To exploit the under-constrained degrees of freedom, we introduced additional constraints, the l_1-norm minimization to the projection matrix, to the problem. The proposed algorithm gives a solution strictly belonging to the $SO(n)$ by tuning Euler angles of high dimension. Sparseness projection matrix enables faster coding.

Experimental results using image features exhibit that our method performs comparable to the state-of-the-art, when our matrices are dense. Very sparse matrices generated by our algorithm perform worse, but the level of degradation is quite moderate.

References

[1] Ambai, M. and Yoshida, Y.C.: CARD: Compact and real-time descriptors, *ICCV*, pp.97–104 (2011).
[2] Calonder, M., Lepetit, V., Strecha, C. and Fua, P.: BRIEF: Binary robust independent elementary features, *ICCV*, pp.778–792 (2011).
[3] Leutenegger, M.C.S. and Siegwart, R.: BRISK: Binary robust invariant scalable keypoints, *ICCV*, pp.2548–2555 (2011).
[4] Rublee, K.K.E., Rabaud, V. and Bradski, G.: ORB: An effcient alternative to SIFT or SURF, *ICCV*, pp.2564–2571 (2011).
[5] Kong, W. and Li, W.: Isotropic hashing, *NIPS*, pp.1655–1663 (2012).
[6] Horn, A.: Doubly stochastic matrices and the diagonal of a rotation matrix, *Am. J. Mathematics*, Vol.76, No.3, pp.620–630 (1954).
[7] Weiss, Y., Torralba, A. and Fergus, R.: Spectral hashing, *NIPS*, pp.1753–1760 (2008).

[8] Gong, Y. and Lazebnik, S.: Iterative quantization: A procrustean approach to learning binary codes, *CVPR*, pp.817–824 (2011).
[9] Wang, J., Kumar, S. and Chang, S.-F.: Sequential projection learning for hashing with compact codes, *ICML*, pp.1127–1134 (2010).
[10] Goemans, M.X. and Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM*, Vol.42, pp.1115–1145 (Nov. 1995).
[11] Min, K., Yang, L., Wright, J., Wu, L., Hua, X.-S. and Ma, Y.: Compact projection: Simple and efficient near neighbor search with practical memory requirements, *CVPR*, pp.3477–3484 (2010).
[12] Achlioptas, D.: Database-friendly random projections, *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp.274–281 (2001).
[13] Li, P., Hastie, T.J. and Church, K.W.: Very sparse random projections, *Proc. International Conference on Knowledge Discovery and Data Mining*, pp.287–296 (2006).
[14] Ambai, M. and Sato, I.: Fast binary coding of local descriptors based on supervised learning, *MIRU* (2012). (*The title is translated into English by the author.*)
[15] Lowe, D.G.: Distinctive image features from scale- invariant keypoints, *IJCV*, Vol.60, pp.91–110 (Nov. 2004).

(Communicated by *Shinichiro Omachi*)