2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Object Verification in Two Views Using Sparse Representation

Shih-Chung, Hsu
Department of Electrical Engineering, National Tsing-Hua University
Hsin-Chu, Taiwan
chvjohnff@gmail.com

I-Cheng Chang
Department of Computer Science and Information,
Dong-Hwa University
Hua-Lien, Taiwan
icchang@mail.ndhu.edu.tw

Chung-Lin, Huang
Department of M-Commerce and Multimedia Applications
Asia University
Tai-Chung, Taiwan
clhuang@asia.edu.tw

Abstract

This paper proposes an object verification method by using sparse representation (SR) which has been applied for object representation and recognition. However, SR dictionary does not show sufficient compactness. Our method comprises three major modules. First, we train the sparse matrix by using boost K-Singular Value Decomposition (boost K-SVD) to obtain a sparse vector set. Second, we combine two training sparse vector sets of the same and different objects from two views to generate a positive/negative combined sparse vector set. Finally, a Support Vector Machine (SVM) classifier is applied for verification. Our contributions are (1) obtaining a sparser vector set using K-SVD, (2) demonstrating the SR matrix with better Restricted Isometry Property (RIP), and (3) applying the SR matrix to the object verification process with high accuracy. The experimental results prove that our method has higher accuracy than the other methods.

Keywords — Boost K-SVD, K-SVD, Sparse Representation(SR), Object Verification

1. Introduction

Object verification has been an essential research topics for many computer vision applications such as video surveillance. There are two types of object verification: identification and similarity check. The first method [1~4] picks up an object from the testing set and finds the best match in the gallery set. The second method [5~8, 9, 10~17] classifies the object pair by a binary classifier even though the similar pairs are not in the gallery set. Similarity check can be applied to the similar object matching or tracking in different scenes.

Feature selection is essential for similarity measure. Shafey *et al.* [10] convert an image to Gabor features by using Gabor filters and transform the Gabor features to the discrete cosine transform (DCT) domain. They model DCT coefficients by using generalized moments for face verification through source fusion. Arróspide *et al.* [18] apply the log–Gabor features for verifying the vehicles. Wang *et al.* [9] define the similarity distance measure of a local binary pattern (LBP) and apply weak AdaBoost classifiers to face verification. Guo *et al.* [19] extract the Gabor feature from a refined or warped face image for verification by using the support vector machine (*SVM*) or K-nearest neighbor classifiers. Ying *et al.* [5, 6] construct the input feature vectors for the SVM classifier based on the nonmetric distance.

However, Gabor feature and the other features may contain the redundant components of the training objects. Therefore, Principal Component Analysis (*PCA*), Principal Component Regression (*PCR*) [16], and Sparse representation (*SR*) [20] are proposed to eliminate redundant components. *SR* is used to recover the training instance from the components called atoms of which their composition is represented by sparse vectors.

The matching alignment of the input features from pairs for face verification in arbitrary views has been proposed[11, 13]. However, given the variations in the appearance of the vehicle across disparate observations, direct matching alignment may not be reliable for verification process. In [5, 6], they try to overcome the problem using the nonmetric edge-distance embedded vector with limited success. Here, we propose an object pair verification method by developing the Boost KSVD dictionary training based on the discriminative features from the training set. In the experiments, we show that our method generates more compact sparse matrix for *SR* which is applied for object verification with better results compared with the other methods.

2. Pair-based Object Verification

Our object verification method (as shown in Figure 1) consists of boost K-SVD dictionary training and pair-based on Radial Basis Function (RBF) SVM object verification. The dictionary for each view is learned from boost K-SVD algorithm which shows better restricted isometry property (*RIP*) [21] compared with the original K-SVD. Boost K-SVD improves the RBF SVM method with better reconstruction accuracy by iteratively appending the atoms in the dictionary.

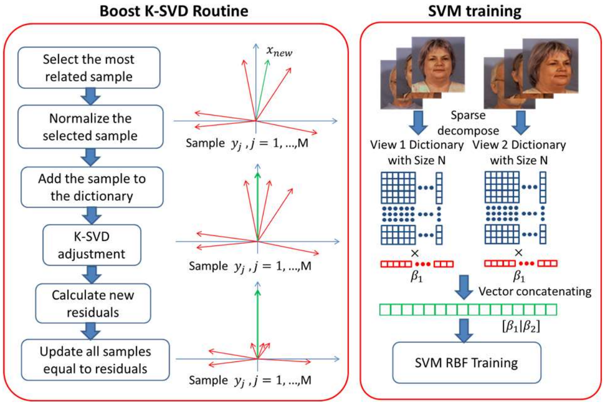
Boost K-SVD Routine
Select the most related sample
Normalize the selected sample
Add the sample to the dictionary
K-SVD adjustment
Calculate new residuals
Update all samples equal to residuals
x_{new}
Sample y_j , $j = 1, \ldots$,M
Sample y_j , $j = 1, \ldots$,M
Sample y_j , $j = 1, \ldots$,M
SVM training
Sparse decompose
View 1 Dictionary with Size N
View 2 Dictionary with Size N
×
×
β_1
β_1
Vector concatenating
$[\beta_1|\beta_2]$
SVM RBF Training

Fig. 1. The system framework of object verification

In [9, 11~13], the object verification methods align the pair of input images by applying image warping. However, the vehicles show different appearance in various views so that we cannot warp the vehicle to the same view by the corresponding points. The size and illumination normalization are not realistic because they are totally different in two views.

To resolve the above problems, we propose an object verification method without using object alignment and illumination normalization. The object verification is simplified as one-by-one similar/dissimilar object pair classification. To verify the same object in two different views, we define the likelihood of the object pair (*i.e.*, object *t* and object *o*) as

$$\mathrm{P}(t \leftrightarrow o|V_1, V_2) = e^{\left(\frac{-\left\|x_{t,V_1}-x_{o,V_1}\right\|^2}{\sigma}\right)} \cdot e^{\left(\frac{-\left\|x_{t,V_2}-x_{o,V_2}\right\|^2}{\sigma}\right)} \quad (1)$$

where x_o is the object vector , $\boldsymbol{x_t}$ is the testing vector, and σ is the variance in input. If $\boldsymbol{x_t}$ and $\boldsymbol{x}_o$ in different views V_1 and V_2 are enough close to each other, then object *t* and object *o* are the same object. We assume that the observations of the same object in two views are independent and the dimension of the observation vector is an independent variable. Here, we combine a pair of vectors into one vector and rewrite eq.(1) as

$$\mathrm{P}(t \leftrightarrow o|V_1, V_2) = e^{\left(\frac{-\left\|x_{pt}-x_{po}\right\|^2}{\sigma}\right)} \quad (2)$$

where $x_{pt} = \left[x_{t,V_1}^T, x_{t,V_2}^T\right]^{\mathrm{T}}$ and $x_{ot} = \left[x_{o,V_1}^T, x_{o,V_2}^T\right]^{\mathrm{T}}$ indicate the same object pairs in two different views. The pair similarity between testing object $\boldsymbol{t}$ and observation $\boldsymbol{o}$ in two views V_1 and V_2 can be examined by applying eq.(2). However, exhaustive object pair searching requires considerably high computation cost in a large object set. Efficient object pair verification involves training the margin function constructed by using several limited similar and different pairs.

For comparisons, we may try other transformation, such as nonmetric distance transform [5~7] and *PCR* [16]. These methods decrease the correlation among extracted features to generate a unique object representation. The nonmetric distance measure based on the canny edge of the object is used to compute the dissimilarity between two objects. Because the distance is not related to the object dis-similarity, the distance measure of the observation vectors of the object pairs has less curse of dimensionality. We add the norm penalty constraint to suppress the curse of dimensionality. In the experiments, we will show the verification results of using *PCR* as the version of L2 norm penalty and Least Angle Regression (*LARS*) [22, 23] as the version of L1 norm penalty.

3. Dictionary Training using Boost K-SVD

To generate the sparser dictionary, we propose Boost K-SVD to speed-up the process by appending a representative atom to the dictionary. Inspired from orthogonal match pursuit (*OMP*) [25] that fits the sample to the most related atom, we find the initial atom which is the most related to all the samples. The atom selection process finds one of the training samples as the initial atom based on the maximum summation of the correlation with the rest of samples.

$$\boldsymbol{x}_{new} = \underset{\mathrm{y}_j}{\operatorname{argmax}} \textstyle\sum_{i=1}^{N} \left|Corr(\mathbf{y}_j, \mathbf{y}_i)\right|, \quad (3)$$

add $\boldsymbol{x}_{new}$to dictionary $\mathbf{X}$ as $\mathbf{X}$=[$\mathbf{X}$, $\boldsymbol{x}_{new}$], $\mathbf{Y}$=[$\boldsymbol{y}_1, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$] is the training sample set, and $\mathbf{X}$=[$\boldsymbol{x}$] is the dictionary with an initial atom $\boldsymbol{x}$ obtained with maximal correlation with all the training samples. If the initial x_{new} is the principle atom with the maximum absolute correlation to all training samples, then the iteration of K-SVD can converge to the principle atom faster.

However, the initial selection do the exhaustive searching to find the sample which is the closest to the principle atom. To reduce the computation, we adapt particle filtering algorithm on the initial atom selection. We assume the training set is variable $\mathbf{s}$ and the observation feature projection on each object sample is variable $\mathbf{z}$, and the distribution of the observation density is $P(\boldsymbol{z}|\boldsymbol{s})$. The observation expectation is obtained as $\mathrm{E}[f(\boldsymbol{s})]$ from the input samples corresponding to the selected particles with the weighting observation density. Here, we ignore the state transition. The k*th* weight $\boldsymbol{W}^{(\boldsymbol{k})}$ on the each selected sample is defined as

$$\boldsymbol{W}^{(k)} = \boldsymbol{W}^{(k-1)} \times P(\boldsymbol{z}_k|\boldsymbol{s}_k), \quad (4)$$

where $\boldsymbol{W}^{(k)}$=$[w_1,\ldots w_N]$, k is the number of iteration. So, we can compute the expectation $\mathrm{E}[f(\boldsymbol{s})]$ as

$$\mathrm{E}[f(\boldsymbol{s})] = \mathrm{E}[\mathbf{y}] = \textstyle\sum_{p=1}^{N} w_p \times \boldsymbol{y}_p \quad (5)$$

where N is the number of particles, and $f(\boldsymbol{s}_p) = \boldsymbol{y}_\mathrm{p}$ represents the *pth* particle sample.

```
Particle Initial Searching:
Initial:
●   Set $P(\boldsymbol{z}_k|\boldsymbol{s}_k)$ as an uniform distribution
For t = 1 to T
Observation measurement
●   Select the particle according to the initial $P(\boldsymbol{z}_k|\boldsymbol{s}_k)$
●   For each particle weight $\omega_p = |\boldsymbol{\beta}|_1, \boldsymbol{\beta} = \mathbf{Y} \times \boldsymbol{y}_p$, Y is
    the training set, and $\boldsymbol{y}_p$ is a selected sample.
●   Calculate the observation atom $\mathrm{E}[f(\boldsymbol{s})] = \mathrm{E}[\mathbf{y}] =$
    $\sum_{p=1}^{P} \omega_p \times \boldsymbol{y}_p$
Importance Updating
●   Let $\boldsymbol{\alpha} = \mathbf{Y} \times \mathrm{E}[f(\boldsymbol{s})]$
●   updating $P(\boldsymbol{z}_k|\boldsymbol{s}_k = \boldsymbol{y}_i) = \frac{\alpha_i}{|\alpha|_1}, i = 1,\ldots,N$, where N
    is the number of training samples
●   The final initial atom $\boldsymbol{x}_{\boldsymbol{new}} =$
    $\underset{\boldsymbol{y}_i}{\mathrm{argmax}}\,|Corr(\boldsymbol{y}_i, \mathrm{E}[f(\boldsymbol{s})])|$
End For
```

The K-SVD [20, 26] algorithm is applied by using generalized K-means clustering method to train the sparse dictionary. K-SVD algorithm avoids the data overfitting and its performance depends on the initial atom selection. However, K-SVD does not involve the selection of initial atoms. Boost K-SVD dictionary consists of atoms which are sparser and more effective for object representation. Atom finding for the dictionary is constructed by minimizing the objective function J defined as

$$J = \min \textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{1}{2} \left|\boldsymbol{y}_\mathrm{i} - \sum_{\mathrm{j}=1}^{\mathrm{K}} \boldsymbol{x}_\mathrm{j} \boldsymbol{\beta}_{\mathrm{ji}}\right|^2 + \sum_{\mathrm{j}=1}^{\mathrm{K}} \lambda |\boldsymbol{\beta}_{\mathrm{ji}}|_1 \quad (6)$$

where N is the size of the training set, K is the size of the dictionary, $\boldsymbol{y_i}$ is the i*th* input sample, $\boldsymbol{x_j}$ is the j*th* atom of the dictionary, $\boldsymbol{\beta}_{\mathrm{ji}}$ is the vector decomposed from atom $\boldsymbol{x}_\mathrm{j}$ for input y_i defined in eq. (8), and λ is the penalty term. To maximize J, we do the partial differential of J *w.r.t* to $\boldsymbol{x}_\mathrm{j}$ and $\boldsymbol{\beta}_{\mathrm{ji}}$ (*i.e.*, $\frac{\partial J}{\partial \boldsymbol{x}_j} = 0$ and $\frac{\partial J}{\partial \boldsymbol{\beta}_{\mathrm{ji}}} = 0$) . To maximize J and find atom $\boldsymbol{x}_j$, we need to solve the following equations

$$\mathrm{Y}^\mathrm{T} x_\mathrm{j} = \textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \boldsymbol{\beta}_{\mathrm{ji}}{}^2\, \mathrm{x}_\mathrm{j} \,,$$

$$\mathrm{x}_\mathrm{j}{}^\mathrm{T}(\mathrm{y}_\mathrm{i} - \mathrm{x}_\mathrm{j}\boldsymbol{\beta}_{\mathrm{ji}}) = \lambda\, \mathrm{sign}(\boldsymbol{\beta}_{\mathrm{ji}}) \quad (7)$$

The K-SVD dictionary is constructed by using eq.(7). The projection on atom $\boldsymbol{x}_\mathrm{j}$ must be larger than the Lagrange multiplier λ. Here, we modify the K-SVD process by re-training the $\boldsymbol{x}_\mathrm{j}$ component and removing the other components. Therefore, we modify eq.(7) in the iteration form as

$$\boldsymbol{x}_\mathrm{j}(\mathrm{t}+1) = \frac{1}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \boldsymbol{\beta}_{\mathrm{ji}}{}^2} \textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \left(\mathrm{y}_\mathrm{i} - \sum_{\mathrm{k}=1,\mathrm{k}\neq\mathrm{j}}^{\mathrm{K}} \mathrm{x}_\mathrm{k}(\mathrm{t}) \boldsymbol{\beta}_{\mathrm{ki}}\right) \cdot \boldsymbol{\beta}_{\mathrm{ji}} \quad (8)$$

subject to $\boldsymbol{\beta}_{\mathrm{ji}} =$

$$\begin{cases} \boldsymbol{x}_{\boldsymbol{j}}{}^\mathrm{T}(\mathrm{t})\boldsymbol{y}_\mathrm{i} - \lambda \mathrm{sign}(\boldsymbol{\beta}_{\mathrm{ji}}), & |\boldsymbol{x}_\mathrm{j}{}^\mathrm{T}(t) \cdot \boldsymbol{y}_\mathrm{i}| - \lambda > 0 \\ 0, & \text{otherwise} \end{cases}$$

Eq.(8) does not provide a closed form solution so that the atoms must be evaluated by using linear programing with the pre-selected initials. The online dictionary learning [27] selects several random training samples as the initials, but some initials may be trapped in the local minima. To ensure that the initial is sufficiently close to the global minima, we randomly select a sample that has large correlation with the other samples as the initial selected atom. In Boost K-SVD, to keep the atoms being nearly mutual orthogonal, we remove some atoms from the selected atom set, *i.e.*, $\mathbf{Z}(n+1) = \mathbf{Z}(\mathrm{n}) - \mathbf{X}\boldsymbol{\beta}$, where $\mathbf{Z}(\mathrm{n}+1)$ is the newly generated atom set at n+1 iteration, $\mathbf{Z}(\mathrm{n})$ is the previous atom set. Then, we select a new initial atom from the updated training set and perform dictionary training again. Repeating the above process to generate the atoms shows more effective representation property and less reconstruction error.

The dictionary shows more effective reconstruction capabilities with less unrelated noise. Boost K-SVD is more effective for reconstruction than the original K-SVD. As shown in Fig. 2, the original K-SVD reconstructs the input by using the holistic vehicle dictionary. More details are reconstructed for larger dictionary, however, the reconstruction may contain the outlier information such as background. The K-SVD algorithm provides a set of atoms to fit the inputs. However, it does not ensure that each atom is sufficiently close to the training set.

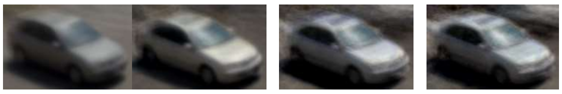

Fig. 2. The reconstruction of original K-SVD [26,27]

Comparing with K-SVD, we use the reconstruction results to show that all atoms generated by our boost K-SVD have better fitting to all dataset, and the reconstruction results have less shape distortion and less background information. With major components learning, our boost K-SVD generates better reconstruction results than original K-SVD as shown in Fig. 3.

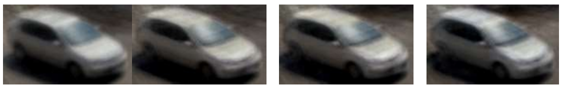

Fig. 3. FThe reconstruction of our atom initializing K-SVD

```
Boost K-SVD Algorithm:
Initial:
  ● Dictionary size $K$ = 1, training set $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]$
  ● Initialize atoms pool Z=Y
Dictionary training iteration:
For m = 1 to M
  ● Select $\boldsymbol{x}_{new}$ by particle initial searching and X=[ X, $\boldsymbol{x}_{new}$]
Single atom training:
For t = 1 to T
  ● Use any L1-solver method[22-24] to solve the sparse vectors $\beta$ for all samples
  ● Update the atom by using (7)
  ● Normalize $x_j(t+1)$=$x_j(t+1)/norm(x_j(t+1))$
  ● If$|x_j(t+1) - x_j(t)| > \varepsilon$, retain single atom training
End for
  ● Remove the dictionary component from the training set $\mathbf{Z}(\mathrm{n}+1) = \mathbf{Z}(\mathrm{n}) - \mathbf{X\beta}$
  ● Stop dictionary training until $|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}|_F^2 \le \epsilon$ or M times iterations are reached
```

4. Experiments

Here, we evaluate the effectiveness of our boost K-SVD for object pair verification applied to different databases. To prove our boost K-SVD dictionary having more accurate representation for the clustering data, we construct the side/frontal view dictionaries using FERET [28], MUCT [29], and vehicle pairs in Tsing Hua University (VPTHU) database. The VPTHU database contains on-campus and on-street vehicle datasets. Forty-two vehicle pairs are captured in a 2-hour long video of the 20 meter long road in the campus, and 500 vehicle pairs are captured in the 8-hour long video of the 1.5 km long city street. Finally, we compare the performance of our method with the other methods for the face and vehicle databases.

4.1 Improvement of Boost K-SVD

Boost K-SVD generates larger sparse components than K-SVD which can be proved by the experiment applied to VPTHU on-campus database. It generates similar results using the VPTHU on-street dataset. In vehicle verification experiment, we pick 250 positive pairs and random 750 negative pairs as the training samples, and the rest of the positive pairs and other 750 random negative pairs as the testing samples. The sparsity comparison between Boost K-SVD and K-SVD is shown in Fig. 4. Boost K-SVD offers the sparser vectors than K-SVD constrained by minimizing L1-norm. It also shows that our SR-based boost K-SVD is suitable for vehicle verification.

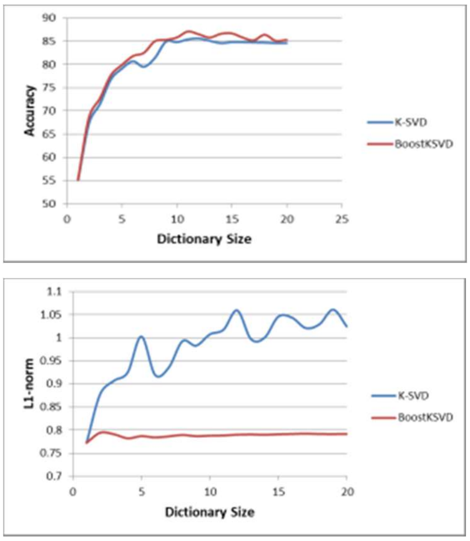

Fig. 4. Accuracy and sparsity between K-SVD and Boost K-SVD.

4.2 Restricted Isometry Property (RIP) Comparison

Restricted isometry property (RIP) illustrates one-to-one mapping relationship between input and output. If the columns transform matrix are independent, the *RIP* inequality is shown as

$$1 - \delta \le \frac{\|X\beta\|^2}{\|\beta\|^2} \le 1 + \delta\,, 0 \le \delta < 1 \qquad (8)$$

where δ is the correlation between the atoms. If the transform has the smaller parameter δ, the columns of transform matrix are closer to orthogonal so that less noise may be projected on each basis.

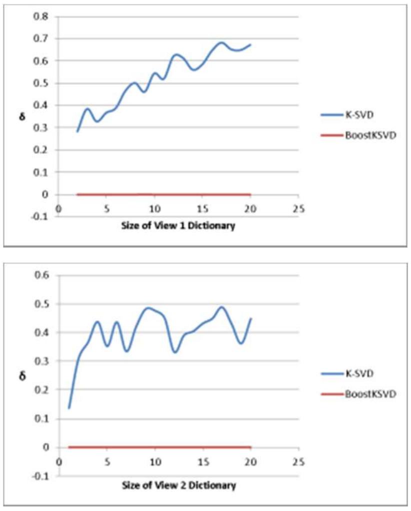

Fig. 5. The RIP δ between Boost K-SVD(red) and K-SVD(blue).

Fig. 5 shows that our boost K-SVD algorithm generates smaller δ for the dictionaries. Our updating process, *i.e.,* $\mathbf{Z}(\mathrm{n}+1) = \mathbf{Z}(\mathrm{n}) - \mathbf{X}\boldsymbol{\beta}$, makes each trained atom near orthogonal with each other. These results show that the accuracy of our method is higher and more stable than others. It shows that our algorithm generates more discriminative features for object verification.

4.3 Performance Comparisons

Performance comparisons consists of three parts. (1) The calibration effect for face and vehicle verification with different feature transform, (2) the results of vehicle verification with nonmetric distance embedded feature, and (3) the accuracy of face verification from various views. Based on the experiments, we can demonstrate the effectiveness of our method applied to object verification.

For face verification, we compare our boost K-SVD method with several methods such as reconstruction based metric learning (*RML*)[13], large margin nearest neighbor (*LMNN*)[17] and principle component regression (*PCR*)[16], and K-SVD. Then we test RML and LMNN by using the dense SIFT features and PCR, K-SVD, boost K-SVD by using the centered vector from the image with calibration[30] and without calibration on the part of MCUT dataset of 375 positive pairs and 1050 negative pairs. In Table I, the calibrated face pairs slightly improve the accuracy of face verification and our boost K-SVD feature shows better results than the others.

TABLE I. THE COMPARISON OF THE FACE VERIFICATION WITH AND WITHOUT CALIBRATION.

Face verification	Calibration	Non-Calibration
Boost K-SVD	95.82%	94.53%
K-SVD	95.07%	93.88%
PCR	93.61%	90.55%
RML	91.2%	87.4%
LMNN	87.3%	84.2%

Table II shows the results of PCR, K-SVD, boost K-SVD features transformed from the VPTHU on-street vehicles. In contrasts of Table I, vehicle calibration by recovering the side view of the vehicle to the frontal view is difficult. Therefore, LMNN and RML are not applied for vehicle verification.

TABLE II. THE COMPARISON OF THE VEHICLE VERIFICATION WITH AND WITHOUT CALIBRATION

Vehicle verification	Calibration	Non-Calibration
Boost K-SVD	87.1%	89.3%
K-SVD	84.2%	88.2%
PCR	85%	87.2%

Besides PCR, K-SVD, boost K-SVD, nonmetric distance[5] is another type of feature applied for vehicle verification. We compare our boost K-SVD and nonmetric distance with the training samples in VPTHU on-campus dataset consisting of 42 positive pairs and 360 negative pairs. Table III shows that our boost K-SVD feature generates better accuracy than nonmetric distance under the same representative components.

TABLE III. THE ACCURACY OF BOOST K-SVD AND NONMETRIC DISTANCE

Accuracy		
4 atoms in each dictionary	***6 atoms in each dictionary***	***8 atoms in each dictionary***
93.15%	93.6%	94.55%
4 embedded pairs	***6 embedded pairs***	***8 embedded pairs***
91%	93%	94%

Fig. 6. shows Boost K-SVD has highest accuracy and is the most stable when the number of atoms increases. The original K-SVD and PCR show acceptable results but less stable than boost K-SVD. Apparently, nonmetric distance is not an efficient feature for VPTHU on-street dataset. It shows that the embedded pairs cannot describe the data sufficiently because the nonmetric distance is not suitable for VPTHU on-street dataset.

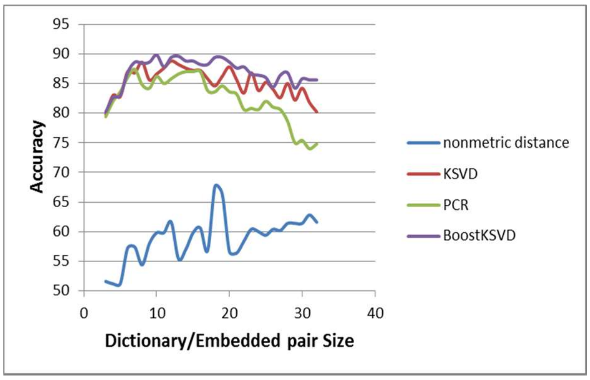

Fig. 6. The accuracy of Boost K-SVD, KSVD, PCR and nonmetric distance

Next, we do the verification experiment of different facial views. We consider two type of dataset to test boost K-SVD. In FERET [28], we pick 4 types of pair views such as frontal to half right, half left, right profile and left profile. The faces in FERET has less common facial features than the faces in MUCT[29]. MUCT consists of 4 types of pair views, frontal to 30 degree right, half right, upward and downward. There are three types of illumination for each face in MUCT. Those results are listed in Table IV. In FERET results, the accuracy is related to the overlapped facial area shared between the faces in the frontal and another angle of views In MUCT results, our boost K-SVD feature can be insensitive to illumination changes.

TABLE IV. THE ACCURACY AMONG THE DIFFERENT PAIR VIEWS ON FERET AND MUCT

FERET Face Verification				
	Half Right	***Half Left***	***Right Profile***	***Left Profile***
Accuracy	86.06%	85.12%	81.38%	80.77%
MCUT Face Verification				
	30° right	***Half Right***	***Upward***	***Downward***
Accuracy	96.27%	95.83%	95.14%	95.22%

5. Conclusion

We develop a verification method to differentiate the same object pairs and the different object pairs. We modify K-SVD by proposing the boost K-SVD. It not only improves the dictionary representation but also generates better accuracy and stable verification outcome.

References

[1] K. Oliba, Y. Sato and K. Ikeuclii, "Visual Learning and Object Verification with Illumination invariance," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, 1997.

[2] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Trans. on PAMI,* pp. 210-227, 30 12 2008.

[3] Y.-N. Chen, C.-C. Han, C.-T. Wang and K.-C. Fan, "Face Recognition Using Nearest Feature Space Embedding," IEEE Trans. on PAMI, pp. 1073-1086, 6 2011.

[4] J. Lu, Y.-P. Tan and G. Wang, "Discriminative Multimanifold Analysis for Face Recognition from a Single Training Sample per Person," *IEEE Trans. on PAMI,* pp. 39-51, 2013.

[5] S. Ying, H. S. Sawhney and R. Kumar, "Vehicle Identification between non Overlapping Cameras without Direct Feature Match," *IEEE Int. Conf. on Computer Vision*, Beijing, 2005.

[6] Y. Shan, H. S. Sawhney and R. Kumar, "Unsupervised Learning of Discriminative Edge Measures for Vehicle Matching between Nonoverlapping," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* pp. 700-711, 4 4 2008.

[7] D. W. Jacobs, D. Weinshall and Y. Gdalyahu, "Class Representation and Image Retrieval with Non-Metric Distances," *IEEE Trans. on PAMI*, pp. 583-600, 6 2000.

[8] Y. Guo, Y. Shan, H. Sawhney and R. Kumar, "PEET: Prototype Embedding and Embedding Transition for Matching Vehicles over Disparate Viewpoints," *IEEE CVPR*, 2007.

[9] X. Wang, C. Zhang and Z. Zhang, "Boosted Multi-Task Learning for Face Verification With Applications to Web Image and Video Search," *IEEE CVPR*, Miami, 2009.

[10] L. E. Shafey , R. Wallace and S. Marcel, "Face Verification using Gabor Filtering and Adapted Gaussian Mixture Models," *Int. Conf. of the Biometrics Special Interest Group*, 2012.

[11] C.-H. Duan, C.-K. Chiang and S.-H. Lai, "Face Verification With Local Sparse Representation," *IEEE Signal Processing Letters,* pp. 177-180, 3 1 2013.

[12] H. Guo, R. Wang, J. Choi and L. S. Davis, "Face Verification Using Sparse Representations," *IEEE CVPR Workshops*, 2012.

[13] J. Lu, W. Gang, D. Weihong and J. Kui, "Reconstruction-Based Metric Learning for Unconstrained Face Verification," *IEEE Trans. on Inf. Forensics and Security*, pp. 79-89, 1 2015.

[14] M. Kan, D. Xu, S. Shan and W. Li,"Learning Prototype Hyperplanes for Face Verification in the Wild," IEEE Trans. on Image Processing, pp. 3310-3316, 8 2013.

[15] Y. Wong, M. T. Harandi, C. Sanderson and B. C. Lovell, "On Robust Biometric Identity Verification Via Sparse Encoding of Faces: Holistic vs Local Approaches," *Int. Joint Conf. on Neural Networks*, Brisbane, 2012.

[16] I. T. Jolliffe, "A Note on the Use of Principal Components in Regression," Journal of the Royal Statistical Society, pp. 300-303, 1982.

[17] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," Proc. NIPS, 2005, pp. 1473–1480.

[18] J. Arróspide and L. Salgado, "Log-Gabor Filters for Image-Based Vehicle Verification," *IEEE Trans. on IP,* pp. 2286-2295, 26 2 2013.

[19] J.-M. Guo, H. Prasetyo and K. Wong, "Vehicle Verification Using Gabor Filter Magnitude with Gamma Distribution Modeling," *IEEE Signal Processing Letters,* vol. 21, pp. 600-604, 18 3 2014.

[20] M. Elad, Sparse and Redundant Representations, New York: Springer, 2010.

[21] E.J. Candes, "The restricted isometry property and its implications for compressed sensing," Comptes Rendus Mathematique, vol. 346, issue 9-10, pp. 589-592, May 2008

[22] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society,* pp. 267-288, 1994.

[23] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, "Least Angle Regression," *The Annals of Statistics,* vol. 32, no. 2, pp. 407-499, 2004.

[24] R. Rubinstein, M. Zibulevsky and M. Elad, "Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit," Computer Science Department - Technical Report, New York, 2008.

[25] S. G. Mallat and Z. Zhang , "Matching pursuits with time-frequency dictionaries," IEEE Transactions on Signal Processing, pp. 3397-3415, 6 12 1993.

[26] R. Rubinstein, T. Peleg and M. Elad, "Analysis K-SVD: A Dictionary-Learning Algorithm for the Analysis Sparse Model," *IEEE Trans. on SP*. 661-677, 24 10 2012.

[27] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online Dictionary Learning for Sparse Coding," *Int. Conf. on Machine Learning*, New York, 2009.

[28] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms," *IEEE Trans. on PAMI*, vol. 22, no. 10, Oct. 2000.

[29] S. Milborrow and J. Morkel and F. Nicolls, "The MUCT Landmarked Face Database," *Pattern Recognition Association of South Africa*, 2010

[30] L. Wolf, T. Hassner, and Y. Taigman, "Similarity Scores Based On Background Samples," ACCV (2), 2009, pp. 88–97.