Neurocomputing 148 (2015) 455–466

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# VoD: A novel image representation for head yaw estimation

Bingpeng Ma [a], Rui Huang [b,*], Lei Qin [c]

[a] School of Computer and Control Engineering, University of China Academy Science, Beijing, China
[b] Huazhong University of Science and Technology, Wuhan, China
[c] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

*Article history:*
Received 8 November 2013
Received in revised form
30 March 2014
Accepted 8 July 2014
Communicated by Jinhui Tang
Available online 19 July 2014

*Keywords:*
Head yaw estimation
Image representation
Fisher vectors
Metric learning

ABSTRACT

Building on the recent advances in the Fisher kernel framework for image classification, this paper proposes a novel image representation for head yaw estimation. Specifically, for each pixel of the image, a concise 9-dimensional local descriptor is computed consisting of the pixel coordinates, intensity, the first and second order derivatives, as well as the magnitude and orientation of the gradient. These local descriptors are encoded by Fisher vectors before being pooled to produce a global representation of the image. The proposed image representation is effective to head yaw estimation, and can be further improved by metric learning. A series of head yaw estimation experiments have been conducted on five datasets, and the results show that the new image representation improves the current state-of-the-art for head yaw estimation.



## 1. Introduction

During the last decades, there has been a significant progress in the face recognition research. However, one of the most challenging factors influencing the robustness and accuracy of face recognition is pose variation. To achieve robustness to pose variation, one might have to process face images differently according to their poses. Therefore, head pose estimation has been an active research topic for many years.

More precisely, head pose estimation essentially means the computation of three types of rotations of the head: yaw (looking left or right), pitch (looking up or down) and roll (tilting left or right). Among them, the roll rotation can be computed easily by the relative positions of the feature points, but the other two rotations are rather difficult to estimate. As the estimation of the yaw rotation has many important applications, it attracts more attention than pitch estimation [1], with more research data available. Therefore, in this paper, as most previous works have done, we focus on the challenging problem of estimating the head yaw pose from the input face images.

In head pose estimation, one of the crucial steps is to extract image representation characterizing the pose. Generally speaking, the proposed visual features can be roughly categorized into *global* and *local* features. While global features encode the holistic configuration of the image, local features encode the detailed traits within a local region. In the literatures, many methods combine both global and local features as they play different roles in the visual perception process. Among them, perhaps the most commonly used one is the Bag-of-Words (BoW) model [2] in which local descriptors extracted from an image are first mapped to a set of visual words and the image is then represented as a histogram of visual word occurrences. Recently, the Fisher vectors [3], which encode higher order statistics of local descriptors, improved the BoW model greatly for image classification. Instead of encoding only the frequency of visual word occurrences, Fisher vectors encode how the parameters of the model should be changed to represent the image. It can be seen as an extension of BoW, and has been shown to achieve the state-of-the-art performance for several challenging object recognition and image retrieval tasks [4,5].

Inspired by these exciting advances, we present a novel image representation for head yaw estimation in this paper. More specifically, the proposed image representation encodes a new type of concise 9-dimensional local descriptors with Fisher vectors to describe the head images, called *fisher Vector of local Descriptors*, VoD for short. The VoD representation has been experimentally validated on five head pose datasets (FacePix, Pointing'04, MultiPIE, CAS-PEAL and our own dataset). The results on these datasets show that the proposed representation outperforms the state-of-the-art.

The contribution of this paper is three-fold. Firstly, we proposed a 9-dimensional local attribute vector which can be applied in the Fisher vector method. The 9-dimensional vector is extracted at each pixel, which contains the coordinates, intensity, the first- and second-order derivatives and the magnitude and orientation of the gradient of the pixel. Compared with the SIFT feature used in the traditional Fisher vectors, the computational efficiency of the 9-dimensional local descriptor is significantly improved. More

* Corresponding author.
*E-mail addresses:* bpma@ucas.ac.cn (B. Ma), ruihuang@hust.edu.cn (R. Huang), lqin@jdl.ac.cn (L. Qin).

http://dx.doi.org/10.1016/j.neucom.2014.07.019
0925-2312/

importantly, despite its conciseness, the descriptor preserves enough information essential to head pose estimation.

Secondly, the proposed local descriptors are encoded and aggregated into Fisher vectors to form the new VoD representation. To keep the spatial structure of the head in the global representation, we divide a head image into many rectangular bins and compute one VoD per bin.

Finally, we further improved the discriminative ability of VoD by supervised metric learning. Considering the great success of Keep It Simple and Straightforward Metric Learning (KISSME) [6], we train kVoD from VoD using KISSME under a supervised setting. The final product improved the accuracy of head pose estimation greatly over the state of the art.

The remainder of this paper is organized as follows: in Section 2, we introduce the related methods for head pose estimation; In Section 3, the proposed representation is introduced in detail. Experiments on five challenging datasets are shown in Section 4 to demonstrate the effectiveness of the proposed representations. Conclusions are drawn in Section 5 with some discussions on the future work.

## 2. Related work

Head pose estimation from images is a challenging problem due to large variations of illumination, facial expressions, subject variability, occlusions, noise and perspective distortion. A generic (i.e., person-independent) algorithm for head pose estimation has to be robust to such factors. There exists a large amount of literatures on this topic, see [7] for a review. Broadly speaking, most previous work can mainly be categorized into three groups: algorithms based on facial features [8–10], model-based algorithms [11,12], and appearance-based algorithms [1,13].

For the algorithms based on facial features, the 3D face structure is exploited along with a priori anthropometric information in order to define the head pose. The elliptic shape of the face, the mouth–nose region geometry, the line connecting the eye centers, the line connecting the mouth corners and the face symmetry are some of the geometric features used to estimate the head pose. This category of algorithms has a major disadvantage: they are sensitive to the misalignment of the facial feature points, while the accurate and robust localization of facial landmarks remains an open problem, especially for the non-frontal faces.

Using the 3D structure of human head, the model-based algorithms build a priori known 3D models for human faces and attempt to match the facial features such as the face contour and the facial components of the 3D face model with their 2D projections. Once the correspondences from 3D to 2D are found between the input data and the face model, conventional pose estimation techniques are exploited to provide the head pose. The main problem for these algorithms is that it is difficult to precisely build the head model for different persons and to define the best mapping of the 3D model to the 2D face image.

The appearance-based algorithms typically assume that there exists a certain relationship between the 3D face pose and some properties of the 2D face image and infer the relationship by using a large number of training images and statistical learning techniques. Intuitively, these appearance-based algorithms can naturally avoid the drawbacks of the algorithms based on facial features and the model-based algorithms. Therefore, they have attracted more and more attention. In these algorithms, instead of using facial landmarks or face models, the whole image of the face is used for pose estimation.

Generally speaking, there are two steps in appearance-based algorithms: feature extraction and classification. For feature extraction, the subspace-based algorithms have been widely used since they can reduce the data dimensionality. Specifically, Gong et al. studied the trajectories of multi-view faces in linear Principal Component Analysis (PCA) feature space [14,15]. They used two Sobel operators (horizontal and vertical) to filter the training images. PCA was then performed to reduce the dimensionality of the training examples. Finally, Support Vector Machine (SVM) regression was utilized to construct two pose estimators for the pitch and yaw angles. Darrell et al. computed a separate eigenspace for each face under each possible pose [16]. The head pose was determined by projecting the input image onto each eigenspace and selecting the one with the lowest residual error. In some sense, this method can be formulated as a Maximum A Posteriori (MAP) estimation problem. Li et al. exploited Independent Component Analysis (ICA) and its variants, subspace analysis and topographic ICA for pose estimation [17]. ICA takes into account higher order statistics required to characterize the view of objects and suitable for the learning of view subspaces. Wei et al. proposed that the optimal orientation of the Gabor filters can be selected for each pose to enhance pose information and eliminate other distractive information like variable facial appearance or changing environmental illumination [13]. In their method, a distribution-based pose model was used to model each pose cluster in Gabor eigen-space. Haj et al. created a system based on a kernelized variant of Partial Least Squares (PLS) that was insensitive to data misalignment, while achieving excellent accuracy on several datasets [18]. Their work shows that regression tools can be very effective for the case of estimating the orientation of a face.

Besides the traditional subspace-based algorithms, since the set of the face images with various poses intrinsically form a manifold in the image space, manifold learning [19–21] for head pose estimation is thus getting popular recently [22–26]. In [22], by thinking globally and fitting locally, Fu and Huang proposed to use the graph embedded analysis method for head pose estimation. They first constructed the neighborhood weighted graph in the sense of supervised locally linear embedding [19]. The unified projection was calculated in a closed-form solution based on the graph embedding linearization, and then they projected new data into the embedded low-dimensional subspace with the identical projection. To overcome the disadvantage that most embedding based methods are unsupervised in nature and do not extract features that incorporate class information, in [27], Huang et al. presented the method Supervised Local Subspace Learning ($SL^2$), which learns a local linear model from a sparse and non-uniformly sampled training set. The authors argued that $SL^2$ was robust to under-sampled regions, over-fitting and image noise. In [28], the authors presented a two layer system (coarse/fine). They assumed that for local patches of the latent manifold, neighborhood-dependent linear functions can be used to effectively describe the modes of variation that correspond to pose changes. Then, they modeled the global nonlinear pose manifold in terms of local linear transforms.

After extracting the representation of the face images, classifiers are trained and then used to determine the actual pose of an input image. Besides the above-mentioned SVMs, some widely used classifiers in pattern recognition, such as neural networks, Bayesian approaches, and Boosting, have all been applied in head pose estimation. In [29], a neural network-based approach was presented in which a multi-layer perception was trained for each pose angle (pan and tilt) by feeding it with preprocessed face images captured by a panoramic camera. In [30,31], based on a Bayesian formulation, Ba et al. proposed an algorithm that couples head tracking and pose estimation in a mixed state particle filter framework. In [32], the authors used Boosting regression and simple Haar-type features to estimate the head pose.

More recently, 3D sensing technologies are becoming ever more affordable and reliable. More and more researchers used the additional depth information to overcome some problems inherent of methods based on 2D data [33,34].

Our method is a novel appearance-based method. Different from the subspace-based algorithms or the manifold learning algorithms, the proposed method aggregates the local descriptors to a global descriptor based on statistics. Specifically, we use the framework of Fisher vector to extract the representations of head images. Fisher vectors are a powerful tool for aggregating local descriptors. It combines the benefits of generative and discriminative features, and has been shown to achieve the state-of-the-art performance for several challenging object recognition and image retrieval tasks [4,5]. To improve the computational efficiency, we propose a concise 9-dimensional local descriptor. And to improve the discriminative power, we further integrate our image representation with a metric learning method, KISSME, in a supervised setting.

## 3. Fisher vectors of local descriptor

This section presents the proposed novel image representation. In the following sections, we first introduce each component of VoD in detail, followed by the extension on how to improve its performance by using metric learning in the supervised setting.

### 3.1. VoD: Fisher vectors of local descriptor

In this section, we introduce VoD in detail. In Fig. 1, we show the flowchart of VoD. From the figure, we see that there are three components for VoD. In the first component, a 9-dimensional local descriptor is extracted for each pixel. In the second component, a GMM is learned based on all the local descriptors in the training set. In the third component, local descriptors are integrated to a global descriptor by computing the gradient of the local descriptor to the GMM center and deviation.

#### 3.1.1. Local descriptor

The first step of VoD is extracting the local descriptors of the input image. In the traditional Fisher vector method, local descriptors are usually the SIFT descriptor reduced to 64 dimensions by PCA. In our case, in order to efficiently capture the spatial, color, gradient and orientation of gradient information, we have designed a very concise 9-dimensional descriptor for each pixel, as shown below:

$$m(x,y,I(x,y)) = (x,y,I(x,y),I_x(x,y),I_y(x,y),I_{xx}(x,y),I_{yy}(x,y),H(x,y),A(x,y)) \tag{1}$$

where $x$ and $y$ are the pixel coordinates, $I(x,y)$ is the raw pixel intensity at position $(x,y)$, $I_x$ and $I_y$ are the first-order derivatives of image $I$ with respect to $x$ and $y$ directions, respectively, while $I_{xx}$ and $I_{yy}$ are the second-order derivatives. The image derivatives are calculated through the filters [1 0 1] and [−1 2 −1]. Considering that HoG descriptor has been successfully applied in many related areas, we also use the magnitude and orientation of the gradient in our descriptor. In Eq. (1), $H(x,y)$ and $A(x,y)$ are the magnitude and the orientation of the gradient, respectively:

$$H(x,y) = \sqrt{I_x(x,y)^2 + I_y(x,y)^2} \tag{2}$$

$$A(x,y) = \arctan\frac{I_y(x,y)}{I_x(x,y)} \tag{3}$$

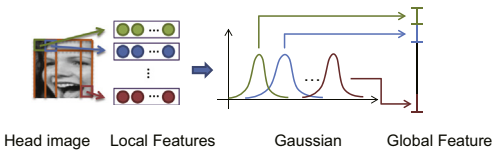


**Fig. 1.** The flowchart of VoD.

After finishing this step, each pixel in the image is represented by a vector with nine dimensions, and the image with the height $h$ and width $w$ is represented by the matrix with row 9 and column $(w \times h)$.

#### 3.1.2. Fisher vectors

After getting the local descriptors, we use the Fisher vectors to aggregate the local descriptors into a global descriptor. Generally speaking, the most widely used way of aggregating local descriptors into a global descriptor may be the BoW model. In BoW, local descriptors are mapped to a set of visual words and the image is represented as a histogram of visual word occurrences. Fisher vectors can be seen as an extension of BoW by going beyond count statistics [5]. Compared with BoW, it provides a more general way to learn a kernel from a generative process of the data. And it has a lower computational cost because its vocabularies are much smaller. Recently, Fisher vectors have been shown to outperform BoW on some large-scale image retrieval tasks [3]. Based on these developments, we select Fisher vectors as the means of aggregating local descriptors into a global descriptor in our representation.

Let $m_t$ be the local descriptors computed from Eq. (1) and $M = \{m_t, t = 1, \ldots, T\}$ be the set of the $T$ local descriptors. The key idea of Fisher vectors is that the generation process of $M$ can be modeled by a probability density function $u_\lambda$ with parameter $\lambda$. In other words, $M$ can be described by a gradient vector:

$$G_\lambda^M = \frac{1}{T}\nabla_\lambda \log u_\lambda(M) \tag{4}$$

The gradient of the log-likelihood describes the contribution of the parameters to the generation process.

The probability density function $u_\lambda$ can be modeled with a Gaussian mixture model (GMM):

$$u_\lambda(m) = \sum_{i=1}^{K} w_i u_i(\mu_i, \sigma_i) \tag{5}$$

where $K$ is the number of Gaussian components. The parameters of the model are $\lambda = \{w_i, \mu_i, \sigma_i, i = 1, \ldots, K\}$, where $w_i$ denotes the weight of the $i$-th component, while $\mu_i$ and $\sigma_i$ are the mean and the standard deviation of the model component, respectively. We assume that the covariance matrices are diagonal. The GMM $u_\lambda$ is trained on a large number of images using Maximum Likelihood (ML) estimation. It is worth pointing out that, in our case, considering the computational efficiency, for each image in the training set, a randomly selected subset of local features are sufficient to train the GMM.

After getting the GMM, the image representations are computed using Fisher vectors. The Fisher vectors of the data $M$ can be denoted by $\mathcal{G}_\lambda^M$. In the Fisher vector method, learning a kernel classifier is equivalent to learning a linear classifier on the Fisher vectors $\mathcal{G}_\lambda^M$. Since learning a linear classifier can be done extremely efficiently, the Fisher vector method is very fast.

Let $\gamma_t(i)$ be the soft assignment of the descriptor $m_t$ to the component $i$:

$$\gamma_t(i) = \frac{w_i u_i(m_t)}{\sum_{j=1}^{K} w_j u_j(m_t)} \tag{6}$$

$\mathcal{G}_{\mu,i}^M$ and $\mathcal{G}_{\sigma,i}^M$ are the 9-dimensional gradients with respect to $\mu_i$ and $\sigma_i$ of the component $i$. They can be computed using the following derivations:

$$\mathcal{G}_{\mu,i}^M = \frac{1}{T\sqrt{w_i}} \sum_{t=1}^{T} \gamma_t(i)\left(\frac{m_t - \mu_i}{\sigma_i}\right) \tag{7}$$

$$\mathcal{G}_{\sigma,i}^M = \frac{1}{T\sqrt{2w_i}} \sum_{t=1}^{T} \gamma_t(i)\left[\frac{(m_t - \mu_i)^2}{\sigma_i^2} - 1\right] \tag{8}$$

where the division between vectors is taken as a term-by-term operation. The gradient vector $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{2\times 9\times K})$ is the

concatenation of $\mathcal{G}^M_{\mu,i}$ and $\mathcal{G}^M_{\sigma,i}$ for $i=1,\ldots,K$ and is therefore $2\times 9\times K$-dimension.

In [5], the authors suggest using normalization to obtain competitive results when Fisher vectors are combined with a linear classifier. In this paper, we apply the power normalization and the $\ell_2$-normalization to normalize each dimension of the image vector to zero mean and unit variance:

$$\hat{\mathcal{G}}=\left(\mathrm{sign}(\mathcal{G}_1)\sqrt{|\mathcal{G}_1|},\mathrm{sign}(\mathcal{G}_2)\sqrt{|\mathcal{G}_2|},\ldots,\mathrm{sign}(\mathcal{G}_{2\times 9\times K})\sqrt{|\mathcal{G}_{2\times 9\times K}|}\right) \quad (9)$$

$$\tilde{\mathcal{G}}=\left(\frac{\hat{\mathcal{G}}_1}{\sum_{i=1}^{2\times 9\times K}\hat{\mathcal{G}}_i},\frac{\hat{\mathcal{G}}_2}{\sum_{i=1}^{2\times 9\times K}\hat{\mathcal{G}}_i},\ldots,\frac{\hat{\mathcal{G}}_{2\times 9\times K}}{\sum_{i=1}^{2\times 9\times K}\hat{\mathcal{G}}_i}\right) \quad (10)$$

To provide a rough approximation of the spatial information, we divide the head image into many rectangular bins and compute one VoD descriptor per bin. Note that to do this we also need to train one different GMM per bin. All the features of different bins are then concatenated to form the final VoD representation of the head image.

In our paper, both the GMM number $K$ and the bin number are empirically chosen as 16 (we justify this in the experiments). Therefore, the dimension of the representation of a head image is 4608 ($=2\times 9\times 16\times 16$). Reducing the dimensionality makes the proposed method more efficient. Here we show that the simple method, such as PCA [35], can work well for the proposed representation. PCA is a traditional linear transformation technique, which can greatly reduce the dimensionality of features. In PCA, the projection matrix $\mathbf{W}_p$ is composed of the orthogonal eigenvectors of the covariance matrix of all the training samples. In most of the cases, we keep the eigenvectors whose eigenvalues account for 97% of the total sum of all eigenvalues, unless otherwise stated in the experiments. After using PCA, we can get the low dimensional representation $\tilde{\mathcal{G}}^p$ of VoD by

$$\tilde{\mathcal{G}}^p=\mathbf{W}_p\times\tilde{\mathcal{G}} \quad (11)$$

Table 1 summarizes the entire algorithm.

### 3.2. kVoD: the improvement of VoD by using metric learning

VoD can be used with any classifiers for head pose estimation. However, as head pose estimation is a very hard problem, which evidently requires the image representation to be discriminative besides its good representation ability. Therefore, we propose to improve VoD in a supervised setting by combining it with discriminant analysis and metric learning. Discriminant analysis and metric learning are supervised methods that improve the discriminative ability of features by using the information of training samples' labels. Generally speaking, the performance of supervised methods is much better than that of unsupervised methods. Specifically, keeping all pairs of positive samples close while separating all negative pairs, metric learning methods based on the class of Mahalanobis distance functions have recently gained considerable interest in the computer vision area.

In this paper, considering its great success in face recognition and person re-identification, and its advantage in efficiency, we use KISSME to learn the metric of VoDs. This variant is denoted as kVoD.

As mentioned in [6], the main advantage of KISSME is the simplicity and efficiency of the learning stage, which only requires the computation of two small sized covariance matrices, one for the positive class (pairs of vectors of the same class) and the other for the negative class (pairs of vectors from different classes). The similarity of the two vectors to be compared is based on a likelihood-ratio test, computing plausibility that their difference belongs either to the positive or the negative class.

In kVoD, the squared distance between two features $\tilde{\mathcal{G}}^p_i$ and $\tilde{\mathcal{G}}^p_j$ under metric $\mathbf{M}$ can be computed by the following equation:

$$d^2_{\mathbf{M}}(i,j)=(\tilde{\mathcal{G}}^p_i-\tilde{\mathcal{G}}^p_j)^T\,\mathbf{M}\,(\tilde{\mathcal{G}}^p_i-\tilde{\mathcal{G}}^p_j) \quad (12)$$

where $\mathbf{M}$ is a positive semi-definite matrix. Based on the positive sample pair and the negative sample pair in the training set, $\mathbf{M}$ is computed by

$$\mathbf{M}=\boldsymbol{\Sigma}^{-1}_{y_{ij}=1}-\boldsymbol{\Sigma}^{-1}_{y_{ij}=0} \quad (13)$$

where

$$\boldsymbol{\Sigma}_{y_{ij}=1}=\sum_{y_{ij}=1}(\tilde{\mathcal{G}}^p_i-\tilde{\mathcal{G}}^p_j)(\tilde{\mathcal{G}}^p_i-\tilde{\mathcal{G}}^p_j)^T \quad (14)$$

$$\boldsymbol{\Sigma}_{y_{ij}=0}=\sum_{y_{ij}=0}(\tilde{\mathcal{G}}^p_i-\tilde{\mathcal{G}}^p_j)(\tilde{\mathcal{G}}^p_i-\tilde{\mathcal{G}}^p_j)^T \quad (15)$$

where $y_i$ is the label of sample $\tilde{\mathcal{G}}^p_i$. $y_{ij}=1$ means positive pairs, i.e., if the samples share the same class label ($y_i=y_j$) and $y_{ij}=0$ otherwise.

Since a projection matrix is more convenient than matrix $\mathbf{M}$ in computing the projection of a new sample, in kVoD, we use the Cholesky factorization to produce an upper triangular matrix $\mathbf{W}_k$ and take it as the projection matrix of the new samples. $\mathbf{W}_k$ is fitted to

$$\mathbf{M}=\mathbf{W}_k^T\times\mathbf{W}_k \quad (16)$$

It must be pointed out that KISSME does not reduce the dimension of representations. So, before using KISSME, we again use PCA to reduce the dimension of the representations. Finally, for the input VoD representation $\tilde{\mathcal{G}}$, the final representation of kVoD can be obtained by the following equation:

$$\mathcal{G}^k=\mathbf{W}_k\times\tilde{\mathcal{G}}^p=\mathbf{W}_k\times\mathbf{W}_p\times\tilde{\mathcal{G}} \quad (17)$$

and Table 2 summarizes the algorithm for extracting kVoD.

### 3.3. Nearest centroid classifier for VoD and kVoD

Since the extraction of VoDs/kVoDs can be regarded as the feature extraction for head yaw estimation, classifiers are still needed to get the yaw pose of the input image. In this paper, the Nearest Centroid (NC) classifier is selected as the classifier to determine the yaw pose given VoDs/kVoDs. In NC, for the training samples with the same class, the $k$-means method is applied to find the $k$ centroids. Then we compute the distance between the input feature and each class centroid, and take the label of the class with the smallest Euclidean distance as the output label.

**Table 1**
The flow of VoD.

Algorithm 1: VoD

**Input**: image **I**,
**STEP 1**:
for each pixel in image **I**, extract a 9-d local descriptor $m(x,y,I(x,y))$ by Eq. (1)
**STEP 2**:
in the stage of training, calculate the parameter $\lambda=\{w_i,\mu_i,\sigma_i,i=1,\ldots,K\}$ of GMM using Eq. (5)
in the stage of testing, skip this step.
**STEP 3**:
calculate $\mathcal{G}^M_{\mu,i}$ and $\mathcal{G}^M_{\sigma,i}$ by Eqs. (7) and (8).
$\mathcal{G}^M_{\mu,i}$ and $\mathcal{G}^M_{\sigma,i}$ are concatenated to $\mathcal{G}$
**STEP 4**:
calculate $\tilde{\mathcal{G}}$ by Eqs. (9) and (10)
**STEP 5**:
in the stage of training, $\mathbf{W}_p$ is computed using PCA based on all $\tilde{\mathcal{G}}$ in the training set
in the stage of testing, skipping this step
**STEP 6**:
$\tilde{\mathcal{G}}^p$ is computed by Eq. (11)
**Output**: the image representation $\tilde{\mathcal{G}}^p$

**Table 2**
The flow of kVoD.

Algorithm 2: kVoD

**Input**: image **I**,
**STEP 1**:
calculate $\tilde{\mathcal{G}}^p$ using VoD in Table 1. Go to STEP 6, if image **I** is a testing sample
**STEP 2**:
prepare the positive and negative sample pair
**STEP 3**:
calculate $\mathbf{\Sigma}_{y_{ij}=1}$ by Eq. (14), and $\mathbf{\Sigma}_{y_{ij}=0}$ by Eq. (15)
**STEP 4**:
calculate **M** by Eq. (13)
**STEP 5**:
calculate $\mathbf{W}_k$ by Eq. (16)
**STEP 6**:
$\mathcal{G}^k$ is computed using $\mathcal{G}_k = \mathbf{W}_k \times \tilde{\mathcal{G}}^p$
**Output**: the image representation $\mathcal{G}_k$

Compared with the Nearest Neighbor (NN) classifier, the NC classifier can eliminate the error caused by the identity since the image difference between the different poses of the same person might be smaller than the image difference between different persons with the same pose.

## 4. Experiments

In this section, to validate the effectiveness of the proposed representations, we perform experiments on five different head pose datasets. These datasets have been used extensively in the recent literatures, allowing direct comparisons with other approaches. And the experimental results show that the proposed representations can improve the performance of the state-of-the-art on head pose estimation.

### 4.1. Datasets and performance evaluation

The proposed representations are evaluated on the following datasets:

*FacePix* [36]: FacePix consists of 181 images for each of 30 individuals, which means the total image number is 5430. Pose variation spans only the yaw direction from $-90^\circ$ to $90^\circ$ with images taken at $1^\circ$ increments.
*Pointing*'04 [37]: Pointing'04 Head Pose Image dataset consists of 2790 color face images for 15 subjects. The head pose of each person ranges from $-90^\circ$ to $90^\circ$ both in the horizontal and vertical directions. The image of each subject has 93 different poses, including 13 yaw angles and 7 pitch angles, plus two extreme cases with pitch angles $90^\circ$ and $-90^\circ$. The bounding box containing the face for each image is provided.
*CMU Multi-PIE* [38]: For the four sessions in the CMU Multi-PIE face dataset, we only use the images in the first session. This session contains 3735 images from different subjects. The head angles vary from $-90^\circ$ to $90^\circ$ with an interval of $15^\circ$.
*CAS-PEAL* [39]: The CAS-PEAL dataset contains 21 poses combining seven yaw poses ($[-45^\circ, 45^\circ]$ with intervals of $15^\circ$ and three pitch poses ($30^\circ$, $0^\circ$ and $-30^\circ$). We use a subset containing totally 4200 images of 200 subjects whose IDs range from 401 through 600.
*Multi-Poses*: The private Multi-Poses dataset consists of 3030 images of 102 subjects taken under normal indoor lighting conditions and fixed background. Both the yaw poses and the pitch poses range within $[-50^\circ, 50^\circ]$ with intervals of $1^\circ$. The sample number is 30 for each class (i.e., yaw pose).

Some example images of these datasets are shown in Figs. 2–6.
For the images in the FacePix and Pointing'04 datasets, the head region has been cropped from the original image. We use

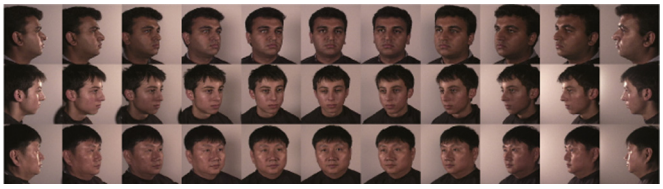

**Fig. 2.** The head images in the FacePix dataset.

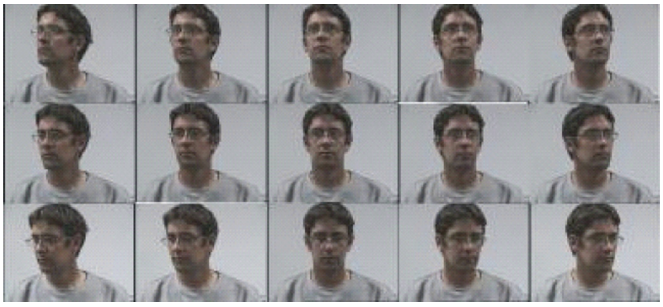

**Fig. 3.** The head images in the Pointinig'04 dataset.

**Fig. 4.** The head images in the MultiPIE dataset.

**Fig. 5.** The head images of one subject in the CAS-PEAL dataset.

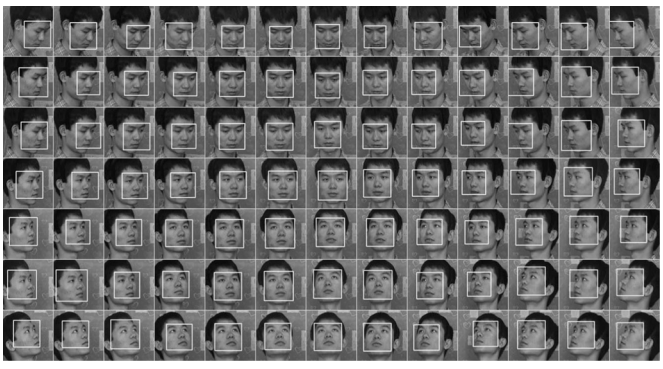

**Fig. 6.** The head images of one subject in the Multi-Pose dataset.

these regions as the input of our method directly. But for the images in the CAS-PEAL, Multi-PIE and Multi-Poses datasets, we use a head detection method [40] to locate the face region from the input images. For all the datasets, the head regions are then normalized to the same size of $32 \times 32$ in the gray-scale space. Histogram equalization is used to reduce the influence of lighting variations. Some images of the results of face detection in the Multi-poses dataset are shown in Fig. 6. From Figs. 4 and 6, we can see that the heads in the Multi-PIE and the Muti-Poses dataset are misaligned. Therefore, from the results of these experiments, we can investigate the robustness of the proposed descriptor to misalignment.

For the FacePix dataset, we estimate the head pose of unknown faces by using a Leave-One-Person-Out (LOPO) strategy. In LOPO, all the images are used for training, except the images of one subject, which will be used for testing. The person to be tested is then changed at each step. In this way, no images of the same person are both in the training and testing parts. The final results are the average of all the tests.

For the other four datasets, 3-fold cross-validation is used to avoid over-training. Specifically, we rank all the images by subjects and divide them into three subsets. Two subsets are taken as the training set and the other is taken as the testing set. In this way, the persons for training and testing are totally different, thus avoiding the over-fitting in identity. Testing is repeated three times by taking each subset as the testing set. The reported results are the average of all the tests.

For the CAS-PEAL and the Multi-PIE dataset, we use the accuracies under the different centroid numbers as the performance measure of the different methods. But for the FacePix, Pointing'04 and Multi-poses, we use the Mean Absolute Error (MAE) as the performance measure. MAE is the mean absolute error between the continuous predicted pose and the ground truth pose. For the Multi-Pose dataset, considering that the sample number is about 40 ($=60/3 \times 2$) for each class (pose) in the training set, the maximal centroid number for each pose is limited to 7, which is different from the experiments on the other datasets.

Besides the results from the references, we also implemented some methods by ourselves and show their results under the NC classifier. We compare the performance of VoD with the following unsupervised methods: PCA, GaFour, Gabor, HoG and SIFT. We also compare the performance of kVoD with the supervised methods of LDA, GFFF, GFC, sHoG and sSIFT. GFFF, GFC, sHoG, and sSIFT are the supervised versions of GaFour, Gabor, HoG, SIFT by using LDA, respectively. As one of the baseline methods in face recognition, PCA [35], is also the baseline method in appearance-based pose estimation, we compare our descriptors with Gabor, HoG and SIFT because these descriptors are the general texture descriptors and have been applied in many areas. Especially, SIFT is often used in the traditional Fisher vector method. For all the methods, PCA is used after feature extraction to reduce the dimension of features and 97% of total energy of eigenvalues is kept. For the supervised methods, we apply PCA first for dimensionality reduction and then LDA for improving the discriminant ability of the representations.

### 4.2. Experiments on the FacePix dataset

Table 3 shows a summary of the experimental results and comparisons to other methods. In Table 3, the MAEs of VoD and

**Table 3**
Comparison of the proposed techniques with leading methods on the FacePix dataset.

| Method | Input | Best error (deg) |
|---|---|---|
| Global model (CCA) [28] | Gabor | 7.99 |
| Localized model (CCA) [28] | Gabor | 2.81 |
| Two-layer framework [28] | Gabor | 3.45 |
| Isomap [28] | Laplacian of Gauss | 8.23 |
| LLE [28] | Laplacian of Gauss | 4.31 |
| LE [28] | Laplacian of Gauss | 4.52 |
| NPE [28] | Grayscale imagery | 8.20 |
| LPP [28] | Grayscale imagery | 9.50 |
| Supervised NPE [28] | Grayscale imagery | 4.40 |
| Supervised LPP [28] | Grayscale imagery | 5.00 |
| VoD | 9-d features | 3.69 |
| **kVoD** | **9-d features** | **2.74** |

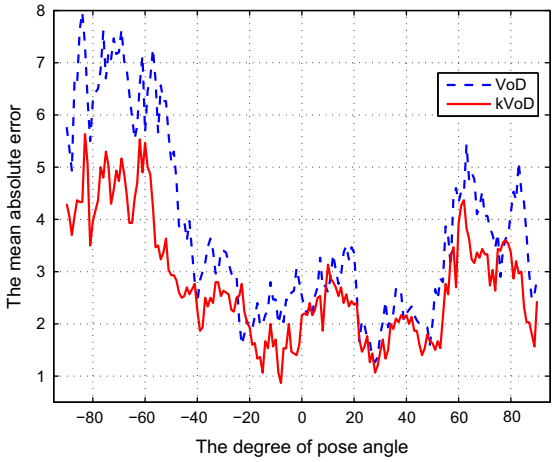


**Fig. 8.** The MAEs under the different poses on the FacePix dataset.

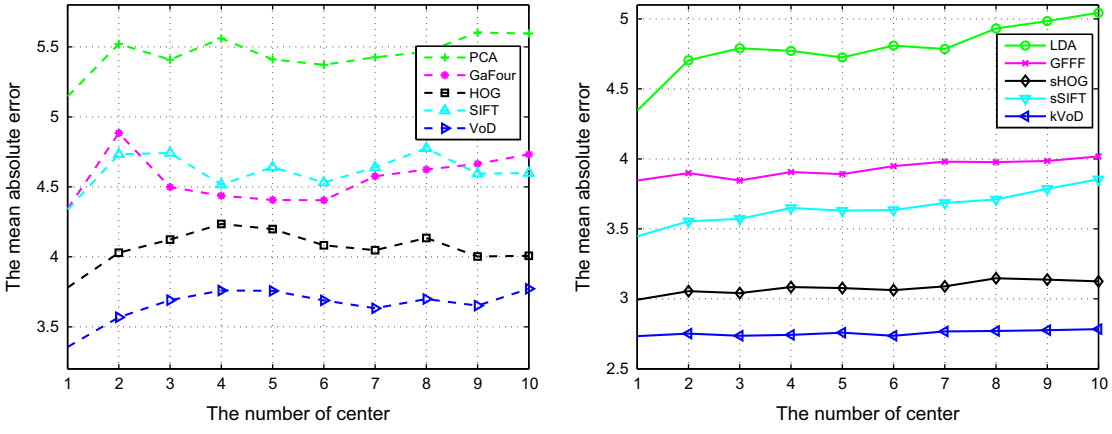


**Fig. 7.** The MAEs of the different methods on the FacePix dataset. The *x*-axis represents the center number of each class and the *y*-axis represents the MAE. The results of different unsupervised methods are shown on the left and supervised on the right.

kVoD are the values when the number of centroid is 6. kVoD achieves the best results among all the methods, which shows the effectiveness of the proposed representations. We can also see that the MAE of the combination of CCA and Gabor representation is 2.81°, very close to the MAE of kVoD's 2.74°. However, compared with Gabor representations with multi-scales and multi-orientations, the advantage of our 9-dimensional representations in efficiency is more obvious.

In Fig. 7, we show the MAEs of different methods with the center numbers ranging from 1 to 10. From the figure we know that for the unsupervised methods, the MAEs of VoD are the lowest under all the center number $k$. And kVoD again gains the best MAEs among all the methods. These results demonstrated the good performance of the proposed representations. Especially, our representations are much better than those of SIFT representations, which shows that the proposed 9-dimensional feature is

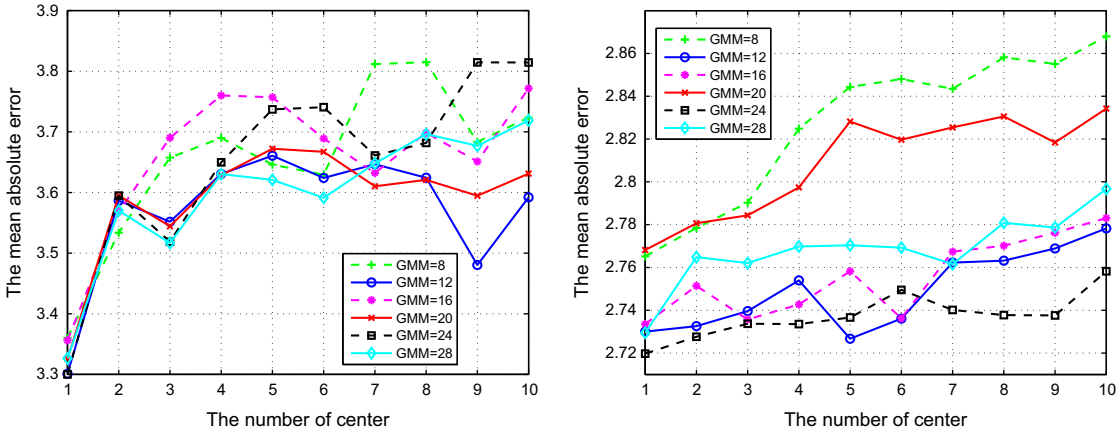


**Fig. 9.** The MAEs of different numbers of GMM components on the FacePix dataset. The $x$-axis represents the center number of each class and the $y$-axis represents the MAE. The results of VoD are shown on the left and kVoD on the right.

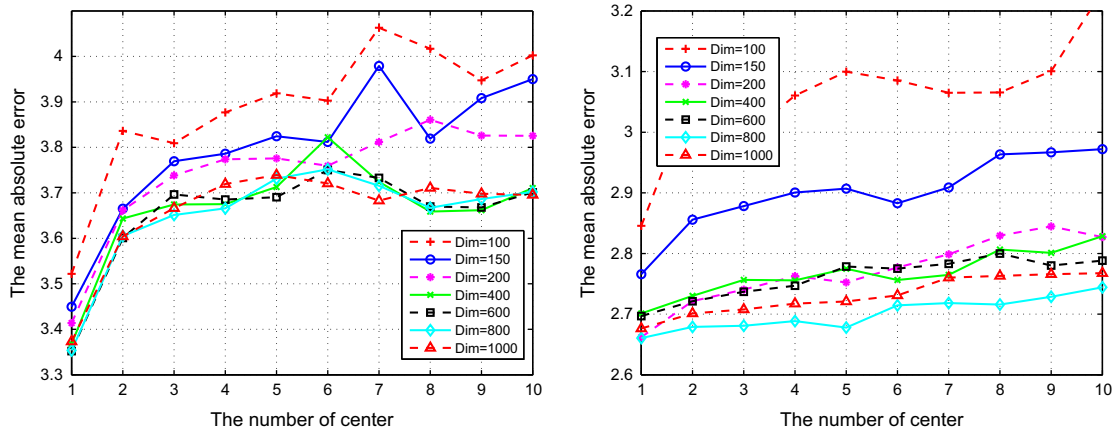


**Fig. 10.** The MAEs of the different dimensions on the FacePix dataset. The $x$-axis represents the center number of each class and the $y$-axis represents the MAE. The results of VoD are shown on the left and kVoD on the right.

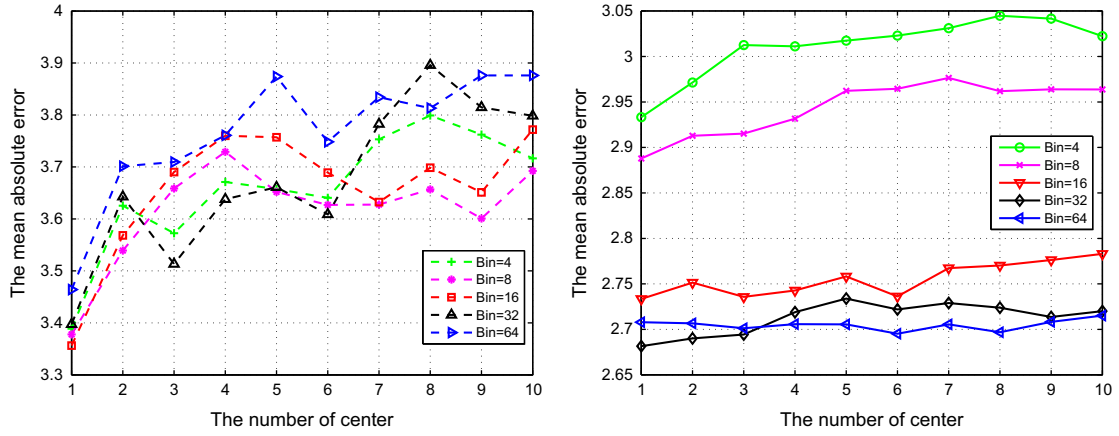


**Fig. 11.** The MAEs of the different bin numbers on the FacePix dataset. The $x$-axis represents the center number of each class and the $y$-axis represents the MAE. The results of VoD are shown on the left and kVoD on the right.

better than the more sophisticated SIFT feature in the task of head pose estimation.

Fig. 8 shows the MAEs of VoD and kVoD with pose variations from $[-90^\circ, 90^\circ]$ at $1^\circ$ intervals. First, the performance of kVoD is consistently better than VoD. We also noticed that within a relative wide range of the frontal view $[-50^\circ, 50^\circ]$, the MAE curves of both kVoD and VoD are relatively flatter, and the MAEs are much smaller than those of the profile poses, which implies that our method is more robust when the head pose is close to frontal.

To investigate the parameters' influence to the performance of VoD and kVoD, we repeat the experiments by using the different numbers of GMM components. The results with respect to the different numbers are shown in Fig. 9. From the figure, we can see that for VoD, the difference of the MAEs is very limited for different numbers of GMM components. But for kVoD, the MAEs with 8 and 20 GMM components are worse than others. Considering that the larger number of GMM components will increase the computational complexity and the storage, in the rest of the experiments, we select the number of GMM components as 16.

In Fig. 10, we also repeat the experiments by using different dimensions of the PCA dimensionality reduction. From the figure, we can see that for both VoD and kVoD, the performances at dimensions 100 and 150 are worse than others. The performances are nearly constant when the dimension is larger than 200. In our experience, keeping 97% of the total energy of eigenvalues work for most of the cases, which is used for the rest of the experiments.

In Fig. 11, we repeat the experiments by using different bin numbers when dividing the original image. In Table 4, we also show the region sizes under different bin numbers. The larger the bin number is, the smaller the region size and the less the number of local descriptors in each bin is. From the figure, we can see that for VoD, the performance does not change much with the increase of the bin number, while for kVoD, using more bins can improve the performance. However, the increment becomes very limited when the bin number is larger than 16. With the increase of the bin numbers, the computational complexity is also increased greatly. Therefore, in the rest of the experiments, we select the bin number to be 16 to balance between the performance and the computational cost.

**Table 4**
The relationship between the number of bins and the region sizes.

| Bin number | 4 | 8 | 16 | 32 | 64 |
|---|---|---|---|---|---|
| Region size | $16\times16$ | $8\times16$ | $8\times8$ | $4\times8$ | $4\times4$ |

**Table 5**
Comparison of the proposed techniques with leading methods on the Pointing'04 dataset.

| Method | Yaw error (deg) |
|---|---|
| Neural network [41] | 9.5 |
| Human performance [42] | 11.8 |
| Associative memories [42] | 10.1 |
| High-order SVD [23] | 12.9 |
| PCA [23] | 14.11 |
| Locally embedded analysis [23] | 15.88 |
| Random forest regression [43] | 9.6 |
| Convex regularized sparse regression [44] | 8.6 |
| VoD | 7.39 |
| **kVoD** | **6.59** |

### 4.3. Experiments on the Pointing'04 dataset

On the Pointing'04 dataset, the comparison between our methods and the other state-of-the-art methods is shown in Table 5. The number of centers in the NC classifier is chosen as 10. From the table, we can see that the MAE of kVoD is only $6.59^\circ$, which is the best in all the methods for head yaw estimation.

We also show the MAEs of the different methods with the center numbers ranging from 1 to 10 in Fig. 12. As the figure shows, the MAEs of kVoD are the lowest under all the center numbers, which shows the good performance of the proposed descriptor. However, the results of VOD are worse than those of Gabor filters when the center number is bigger than 2. We also can find that the performances of VoD and kVoD are much better than the other two texture descriptors sHoG and sSIFT, which shows the advantage of the proposed simple nine-dimensional feature in head pose estimation.

Fig. 13 shows the MAEs of VoD and kVoD under different pose angles on the Pointing'04 database. From the figure, we can see that when poses vary from the front to the profile, the MAEs increase gradually, which is obviously reasonable. For the frontal head image, it is easy to locate the position of the faces for the face detection methods. So, the front faces are near aligned. But for the face under the profile, misalignments increase the difficulty of head pose estimation. And not surprisingly, kVoD performs better than VoD.

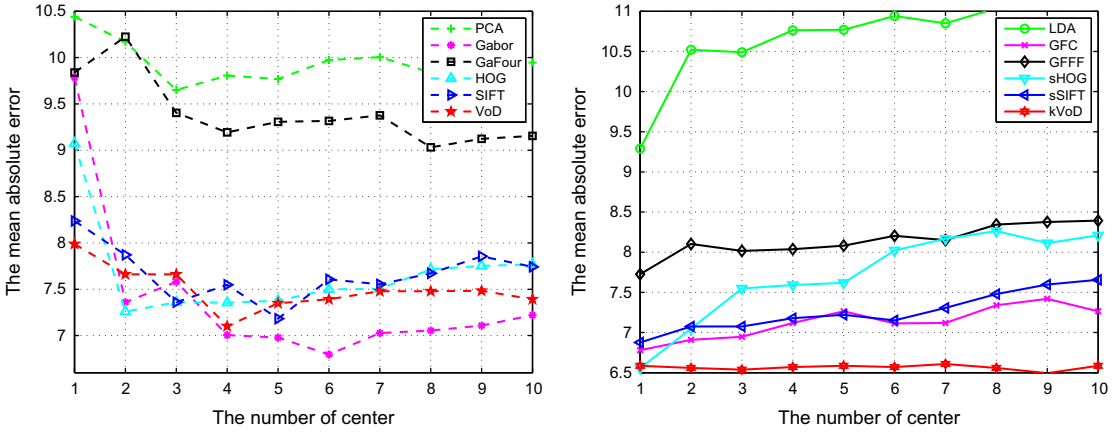


**Fig. 12.** The MAEs of head yaw estimation on the Pointing'04 dataset. The *x*-axis represents the center number of each class and the *y*-axis represents the MAE. The results of different unsupervised methods are shown on the left and supervised on the right.

### 4.4. Experiments on the MultiPIE dataset

Fig. 14 shows the accuracies of different methods on the MultiPIE dataset. From the figure, we can observe that for the unsupervised methods, the accuracy increases with the increase of the number of centers $k$ when $k$ is small. However, for the supervised methods, such as kVoD, the accuracies are nearly equal for different $k$, which actually implies the excellent compactness of each class in the feature space obtained by metric learning.

On the MultiPIE dataset, compared with other unsupervised method, the advantage of VoD is more obvious. For example, the accuracies of VoD are higher than 96% while the accuracies of other unsupervised method are less than 94% when the center number is larger than 4. The accuracies of VoD are even better than some of the supervised methods' accuracies. All these cases show the advantage of VoD.

After using the metric learning method, the results of kVoD are the best of all methods. For different numbers of centers, the accuracies of kVoD are constantly 99.2% while the results of GFFF and GFC are about 96.8% and 98.0%, respectively. The robustness of kVoD to the number of centers also shows the discriminative ability of kVoD. Based on the robustness of kVoD to the center number, in the real system, we can just select 5 or 6 centers for each pose, which can decrease the computational cost of matching the gallery samples and the probe sample.

Fig. 15 shows the accuracies of VoD and kVoD under different pose angles on the MultiPIE database. Again, kVoD performs better than VoD. And VoD's performance is not as stable as kVoD with respect to different poses.

### 4.5. Experiments on the CAS-PEAL dataset

In Fig. 16, we show the accuracies of different methods on the CAS-PEAL dataset. From the figure, we can see that compared with other unsupervised method, the performance of VoD is the best and the advantage is very obvious. After combining with the metric learning method, the accuracies of kVoD are about 94.2% and it is the best of all the methods. But, the accuracies of kVoD are very near to those of GFC.

In Fig. 17, we show the accuracies of VoD and kVoD under different pose angles on the CAS-PEAL database. Similar to the previous observations, kVoD performs better than VoD, and the frontal head images are much easier to be estimated than those of the profile ones.

### 4.6. Experiments on the Multi-Pose dataset

In Fig. 18, we show the MAEs of the different methods on the Mutli-Poses dataset. Similar to the scenes on the other datasets, the results of VoD are the best in all the unsupervised methods and the

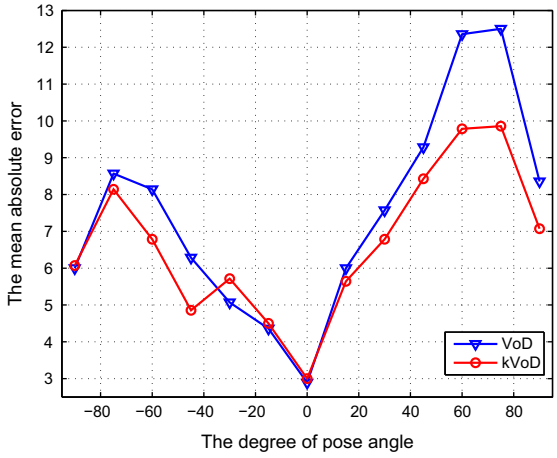


**Fig. 13.** The MAEs under the different poses on the Pointing'04 dataset.

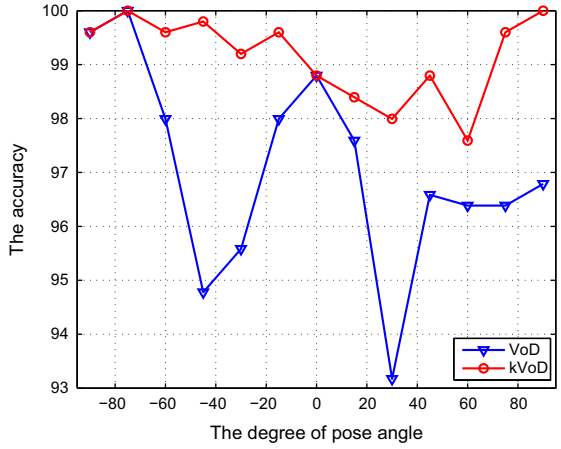


**Fig. 15.** The accuracies under the different poses on the MultiPIE dataset.

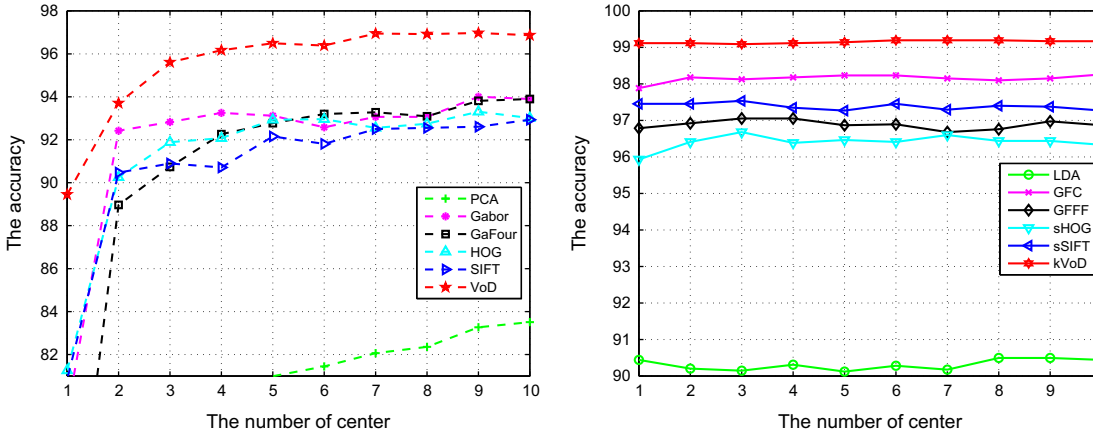


**Fig. 14.** The accuracies of different methods on the MultiPIE dataset. The $x$-axis represents the center number of each class and the $y$-axis represents the accuracy. The results of different unsupervised methods are shown on the left and supervised on the right.

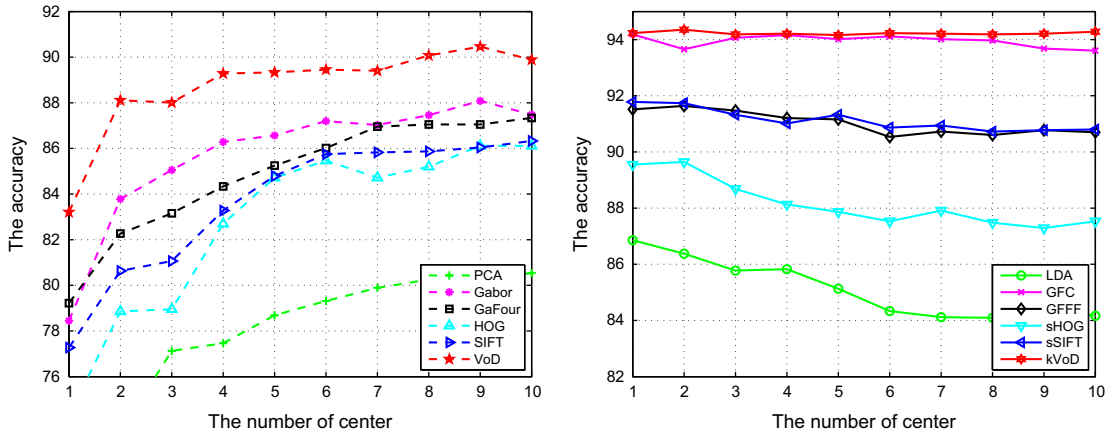


**Fig. 16.** The accuracies of different methods on the CAS-PEAL dataset. The *x*-axis represents the center number of each class and the *y*-axis represents the accuracy. The results of different unsupervised methods are shown on the left and supervised on the right.

results of kVoD are the best of all the methods. Especially, for different numbers of centers, the MAEs of VoD are close to 4.6° while those of kVoD are close to 4.1°. The good performance of the proposed method again shows that the proposed method can improve the state-of-the-art performance for head yaw estimation.

In Fig. 19, we show the accuracies of VoD and kVoD under different pose angles on the Multi-Poses dataset. From the figure, we can see that by using metric learning, kVoD can improve the performance of VoD, and the most significant improvement appears when the pose angles are close to 0° (e.g., [−30°, 30°]).

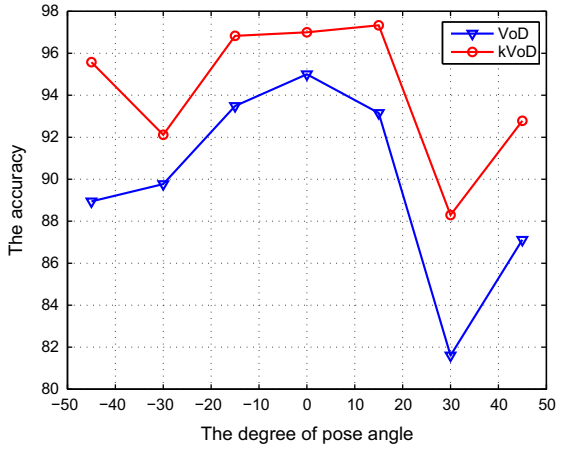


**Fig. 17.** The accuracies under the different poses on the CAS-PEAL dataset.

## 5. Conclusions

In this paper, we propose a novel image representation for the problem of head pose estimation. The proposed representation encodes a new type of concise 9-dimensional local descriptors into a globe descriptor as Fisher vectors. The performance of the proposed representation can be improved further by using metric learning. We test our method on five challenging datasets, outperforming the current state-of-the-art on all these datasets.

There are several aspects to be further studied in the future. For example, in our method, the background and the noise in the image are computed as a part of the image representation. Inevitably, this decreases the performance of the head pose estimation. How to eliminate the effect of the background should

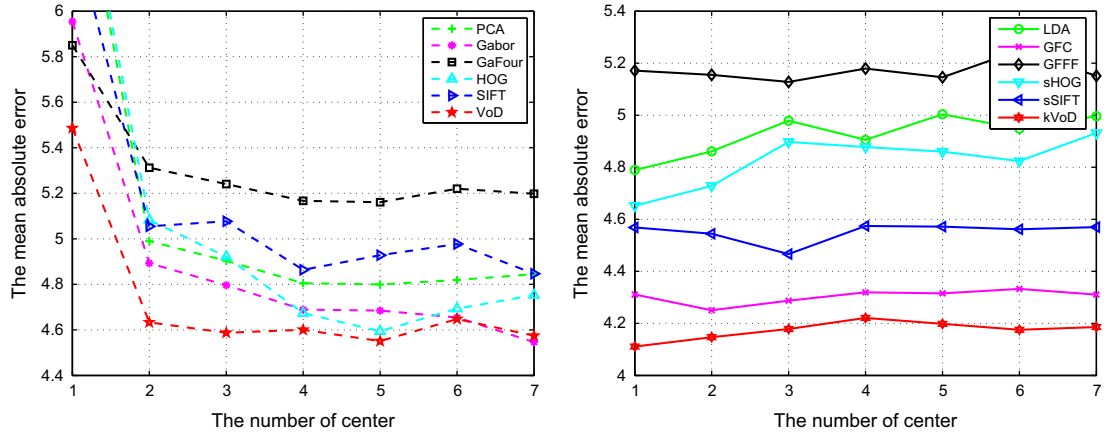


**Fig. 18.** The MAEs of the different methods on the Multi-Poses dataset. The *x*-axis represents the center number of each class and the *y*-axis represents the MAE. The results of different unsupervised methods are shown on the left and supervised on the right.

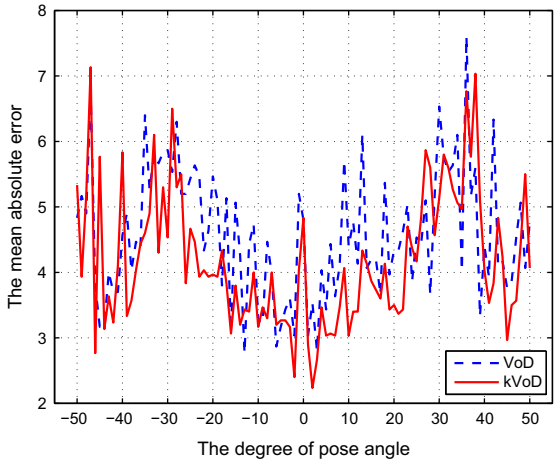


**Fig. 19.** The MAEs under the different poses on the Multi-Poses dataset.

be considered. We are also investigating the other applications of the new image representation.

## Acknowledgments

This paper is partially supported by National Natural Science Foundation of China under Contract nos. 61003103, 61173065, 61105014, 61332016, and the President Fund of UCAS.

## References

[1] L. Chen, L. Zhang, Y. Hu, M. Li, H. Zhang, Head pose estimation using fisher manifold learning, in: Proceedings of IEEE International Workshop on Analysis and Modeling of Faces and Gestures, 2003, pp. 203–207.
[2] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: Proceedings of IEEE International Conference on Computer Vision, 2003.
[3] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[4] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher kernel for large-scale image classification, in: Proceedings of European Conference on Computer Vision, 2010, pp. 143–156.
[5] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed Fisher vectors, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[6] M. Köstinger, M. Hirzer, P. Wohlhart, P.M. Roth, H. Bischof, Large scale metric learning from equivalence constraints, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012.
[7] E. Murphy-Chutorian, M.M. Trivedi, Head pose estimation in computer vision: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 31 (4) (2009) 607–626.
[8] A. Nikolaidis, I. Pitas, Facial feature extraction and determination of pose, in: Proceedings of NOBLESSE Workshop on Nonlinear Model Based Image Analysis, 1998, pp. 257–262.
[9] F. Fleuret, D. Geman, Fast face detection with precise pose estimation, in: Proceedings of IEEE International Conference on Pattern Recognition, vol. 1, 2002, pp. 235–238.
[10] R. Stiefelhagen, J. Yang, A. Waibel, A model-based gaze tracking system, in: IEEE International Joint Symposia on intelligence and Systems, 1996, pp. 304–310.
[11] J. Xiao, T. Moriyama, T. Kanade, J.F. Cohn, Robust full-motion recovery of head by dynamic templates and re-registration techniques, Int. J. Imaging Syst. Technol. 13 (2003) 85–94.
[12] Q. Ji, R. Hu, 3D face pose estimation and tracking from a monocular camera, Image Vis. Comput. 20 (7) (2002) 499–511.
[13] Y. Wei, L. Fradet, T. Tan, Head pose estimation using Gabor eigenspace modeling, in: Proceedings of IEEE International Conference on Image Processing, 2002, pp. 281–284.
[14] Y. Li, S. Gong, H. Liddel, Support vector regression and classification based multi-view face detection and recognition, in: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, 2000, pp. 300–305.
[15] J.N.S. Kwong, S. Gong, Learning support vector machines for a multi-view face model, in: Proceedings of British Machine Vision Conference, 1999, pp. 300–305.
[16] T. Darrell, B. Moghaddam, A.P. Pentland, Active face tracking and pose estimation in an interactive room, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1996.
[17] Stan.Z. Li, X.G. Lu, X.W. Hou, X.H. Peng, Q.S. Cheng, Learning multiview face subspaces and facial pose estimation using independent component analysis, IEEE Trans. Image Process. 14 (6) (2005) 705–712.
[18] M.A. Haj, J. Gonzlez, L.S. Davis, On partial least squares in head pose estimation: how to simultaneously deal with misalignment, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2602–2609.
[19] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[20] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[21] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Adv. Neural Inf. Process. Syst. 15 (2001) 585–591.
[22] Y. Fu, T.S. Huang, Graph embedded analysis for head pose estimation, in: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, 2006, pp. 3–8.
[23] J. Tu, Y. Fu, Y. Hu, T.S. Huang, Evaluation of head pose estimation for studio data, in: TheFirst International Evaluation Workshop on Classification of Events, Activities and Relationships, 2007, pp. 281–290.
[24] N. Hu, W. Huang, S. Ranganath, Head pose estimation by non-linear embedding and mapping, in: Proceedings of IEEE International Conference on Image Processing, vol. 2, 2005, pp. 342–345.
[25] B. Raytchev, I. Yoda, K. Sakaue, Head pose estimation by nonlinear manifold learning, in: Proceedings of IEEE International Conference on Pattern Recognition, vol. 4, 2004, pp. 462–466.
[26] V.N. Balasubramanian, J. Ye, S. Panchanathan, Biased manifold embedding: a framework for person-independent head pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, June 2007, pp. 1–7.
[27] D. Huang, M. Storer, F.D.l Torre, H. Bischof, Supervised local subspace learning for continuous head pose estimation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011.
[28] J. Foytik, V.K. Asari, A two-layer framework for piecewise linear manifold-based head pose estimation, Int. J. Comput. Vis. 101 (2) (2013) 270–287.
[29] R. Stiefelhagen, J. Yang, A. Waibel, Modeling focus of attention for meeting indexing based on multiple cues, IEEE Trans. Neural Netw. 13 (July (4)) (2002) 928–938.
[30] S.O. Ba, J.M. Odobez, A probabilistic framework for joint head tracking and pose estimation, in: Proceedings of IEEE International Conference on Pattern Recognition, 2004.
[31] S.O. Ba, J.M. Odobez, Evaluation of multiple cue head pose estimation algorithms in natural environments, in: Proceedings of IEEE International Conference on Multimedia and Expo, 2005, pp. 1330–1333.
[32] Z. Yang, H. Ai, B. Wu, S. Lao, L. Cai, Face pose estimation and its application in video shot selection, in: Proceedings of IEEE International Conference on Pattern Recognition, 2004.
[33] G. Fanelli, J. Gall, L. Van Gool, Real time head pose estimation with random regression forests, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011.
[34] G. Fanelli, J. Gall, L. Van Gool, Random forests for real time 3D face analysis, Int. J. Comput. Vis. (2012).
[35] M.A. Turk, A.P. Pentland, Eigenfaces for recognition, J. Cogn.Neurosci. 3 (1) (1991) 71–86.
[36] J. Black, M. Gargesha, K. Kahol, P. Kuchi, S. Panchanathan, A framework for performance evaluation of face recognition algorithms, in: ITCOM, Internet Multimedia Systems II, 2002.
[37] N. Gourier, D. Hall, J.L. Crowley, Estimating face orientation from robust detection of salient facial features, in: Proceedings of IEEE International Conference on Pattern Recognition International Workshop on Visual Observation of Deictic Gestures, 2004, pp. 183–191.
[38] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Guide to the CMU Multi-PIE Database, Technical Report, Carnegie Mellon University, 2007.
[39] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, D. Zhao, The CAS-PEAL large-scale chinese face database and baseline evaluations, IEEE Trans. Syst. Man Cybern. Part A 38 (1) (2008) 149–161.
[40] S. Yan, S. Shan, X. Chen, W. Gao, J. Chen, Matrix-structural learning (MSL) of cascaded classifier from enormous training set, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[41] R. Stiefelhagen, Estimating head pose with neural networks results on the Pointing04 ICPR workshop evaluation data, in: Proceedings of IEEE International Conference on Pattern Recognition Workshop, 2004.
[42] N. Gourier, J. Maisonnasse, D. Hall, J.L. Crowley, Head pose estimation on low resolution images, in: The First International Evaluation Workshop on Classification of Events, Activities and Relationships, 2006, pp. 270–280.
[43] Y. Li, S. Wang, X. Ding, Person-Independent Head Pose Estimation Based on Random Forest Regression, September 2010, pp. 1521–1524.
[44] H. Ji, R. Liu, F. Su, Z. Su, Y. Tian, Robust head pose estimation via convex regularized sparse regression, in: Proceedings of IEEE International Conference on Image Processing, 2011, pp. 3617–3620.

**Bingpeng Ma** received the B.S. degree in mechanics, in 1998, and the M.S. degree in mathematics, in 2003, from Huazhong University of Science and Technology. He received Ph.D. degree in computer science at the Institute of Computing Technology, Chinese Academy of Sciences, PR China, in 2009. He was a post-doctorial researcher in University of Caen, France, from 2011 to 2012. He joined the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, in March 2013, and now he is an assistant professor. His research interests cover image analysis, pattern recognition, and computer vision.

**Rui Huang** received his B.Sc. degree in Peking University (1999) and M.E. degree in Chinese Academy of Sciences (2002). In 2008, he received his Ph.D. degree in Rutgers University, and had since worked there as a research associate for two years. He was an assistant professor at Huazhong University of Science and Technology, and is currently a research staff member at NEC Laboratories China. His research interests include graphical models and their applications in computer vision, pattern recognition and medical imaging.

**Lei Qin** received the B.S. and M.S. degrees in mathematics from the Dalian University of Technology, Dalian, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008. He is currently an associate professor with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include image/video processing, computer vision, and pattern recognition. He has authored or coauthored over 30 technical papers in the area of computer vision. He is a reviewer for IEEE Trans. on Multimedia, IEEE Trans. on Circuits and Systems for Video Technology, and IEEE Transactions on Cybernetics. He has served as TPC member for various conferences, including ECCV, ICPR, ICME, PSIVT, ICIMCS, PCM.