Head Pose Estimation Based on Multivariate Label Distribution

Xin Geng* and Yu Xia
School of Computer Science and Engineering
Southeast University, Nanjing, China
{xgeng, xiayu}@seu.edu.cn

Abstract

Accurate ground truth pose is essential to the training of most existing head pose estimation algorithms. However, in many cases, the "ground truth" pose is obtained in rather subjective ways, such as asking the human subjects to stare at different markers on the wall. In such case, it is better to use soft labels rather than explicit hard labels. Therefore, this paper proposes to associate a multivariate label distribution (MLD) to each image. An MLD covers a neighborhood around the original pose. Labeling the images with MLD can not only alleviate the problem of inaccurate pose labels, but also boost the training examples associated to each pose without actually increasing the total amount of training examples. Two algorithms are proposed to learn from the MLD by minimizing the weighted Jeffrey's divergence between the predicted MLD and the ground truth MLD. Experimental results show that the MLD-based methods perform significantly better than the compared state-of-the-art head pose estimation algorithms.

1. Introduction

Head pose plays an important role in the interpersonal communication. Static head pose might indicate a particular direction, the attention of a person, the target of a conversation, etc. The change of head pose as a gesture can also convey rich information, such as agreement, dissent, understanding, confusion, surprise, etc. In addition, head pose is a key information for many other applications, such as face recognition, expression recognition, gaze estimation, etc. As a result, head pose estimation has become an important application of computer vision and pattern recognition. Accordingly, a lot of head pose estimation methods have been proposed in recent years, such as the nonlinear regression methods [17, 8, 19, 9, 11], the subspace embedding methods[18, 12, 4, 14], and the special-feature-based methods [10, 20, 2, 15].

In the context of computer vision, head pose estimation usually means prediction of the head orientation with respect to the image plane. It is a common practice to assume the human head to be a rigid object so that there are three degrees of freedom (DOF) in head pose, i.e., yaw, pitch and roll [16]. For many existing data sets, only a finite number of discrete angles are considered, and only the yaw and pitch angles are available [16]. Thus in this paper, we focus on the problem of head pose estimation with finite discrete yaw and pitch angles. Note that the methods proposed in this paper can be easily extended to three DOF, or specialized to a single DOF.

In practice, the ground truth of head pose is difficult to obtain. Usually, approximate approaches are used to acquire coarse poses. For example, the widely used Pointing'04 head pose database [7] is collected by asking the human subjects sitting at the same position of the room to stare successively at 93 markers attached to different positions of the room. As pointed out in [16], only coarse pose can be obtained in this way because of two reasons. First, the head of the subject is not guaranteed to be at the exactly same location in the 3D space. Second, people can not direct their head toward a marker very accurately. As a result, even when two face images are labeled with the same pose, they might have quite different real poses. Moreover, for most real applications, very accurate head pose, say, precise angle to one degree, is not necessary. This motivates us to regard both the "ground truth" pose and the predicted pose as some kind of *soft* labels, rather than explicit *hard* labels.

Since the head orientation is intrinsically continuous, the face images with close head orientations look quite similar, and such similarity gradually fades away when the angle between the head orientations increases. This observation can be utilized to design the soft label for head pose. While the "ground truth" pose of a face image is considered to be the most relevant label to the image, those poses close to the "ground truth" pose can also be used to describe the image, but with lower relevance. If a real number called *description degree* is used to indicate the relevance of a pose to a

*This research was supported by NSFC (61273300, 61232007) and the Key Lab of Computer Network and Information Integration of Ministry of Education of China.

face image, then in the two-dimensional pose space spanned by the yaw and pitch angles, the description degrees of all the poses constitute a data form similar to a multivariate probability distribution. So, such soft label of head poses is called *multivariate label distribution* (MLD).

The transformation from explicit hard labels to MLD brings at least two advantages. First, the problem of inaccurate "ground truth" poses is alleviated. So long as the "ground truth" pose is roughly correct, i.e., not far away from the real pose, the description degree of the real pose is still high enough to convey positive label information. Second, when learning the model for a particular pose, the face images in the neighboring poses can also help since their poses might also have non-zero description degrees. This means that the training examples for each pose can be boosted without actually expanding the total training set.

The rest of the paper is organized as follows. Section 2 gives the definition of MLD and introduces how to generate an MLD for a given instance. Section 3 proposes two methods to learn from MLD. In Section 4, experimental results are reported. Finally, conclusions are drawn in Section 5.

2. Multivariate Label Distribution

Suppose the description degree of a pose $\boldsymbol{y}$ ($\boldsymbol{y}$ is a two-dimensional vector composed by the yaw and pitch angles) to a face image $\boldsymbol{x}$ is represented by $d_{\boldsymbol{x}}^{\boldsymbol{y}}$. If a coarse "ground truth" pose $\hat{\boldsymbol{y}}$ is assigned to $\boldsymbol{x}$, then, $d_{\boldsymbol{x}}^{\hat{\boldsymbol{y}}}$ should be the highest among all possible poses. Due to the appearance similarity of the neighboring poses, a pose $\tilde{\boldsymbol{y}}$ close to $\hat{\boldsymbol{y}}$ should also have a non-zero description degree $d_{\boldsymbol{x}}^{\tilde{\boldsymbol{y}}}$, which is lower than $d_{\boldsymbol{x}}^{\hat{\boldsymbol{y}}}$. The description degree decreases with the increase of the distance from $\tilde{\boldsymbol{y}}$ to $\hat{\boldsymbol{y}}$. Assume that $d_{\boldsymbol{x}}^{\boldsymbol{y}} \in [0,1]$ and $\sum_{\boldsymbol{y}} d_{\boldsymbol{x}}^{\boldsymbol{y}} = 1$. Then, for a particular face image, the description degrees of all possible poses constitute a multivariate label distribution (MLD). Note that although MLD shares the same properties with probability distribution, we still suggest to interpret them differently. This is because that probability distribution implies that only one pose is correct for a face image, while MLD allows using multiple poses to describe a face image. The latter matches the fact better that "pose" itself is not precisely defined.

In order to generate a reasonable MLD for a given face image $\boldsymbol{x}$, we assume that the MLD follows a discretized bivariate Gaussian distribution centered at the coarse "ground truth" pose $\hat{\boldsymbol{y}}$, i.e.,

$$d_{\boldsymbol{x}}^{\boldsymbol{y}} = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}Z} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\hat{\boldsymbol{y}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\hat{\boldsymbol{y}})\right), \quad (1)$$

where $\boldsymbol{\Sigma}$ is a 2×2 covariance matrix, Z is a normalization factor that makes sure $\sum_{\boldsymbol{y}} d_{\boldsymbol{x}}^{\boldsymbol{y}} = 1$.

Fig. 1 shows some typical head poses from the Pointing'04 database [7] together with their MLDs generated by Eq. (1). The "ground truth" pose is given under each face image. The covariance matrix $\boldsymbol{\Sigma}$ is set to $\begin{bmatrix}\tau^2 & 0\\ 0 & \tau^2\end{bmatrix}$, where τ is the finest granularity of the pose angles. In Fig. 1, each MLD is represented by a gray-scale image. Each pixel of the image corresponds to one yaw angle at the horizontal axis and one pitch angle at the vertical axis. For each pixel, higher intensity (lighter) means higher description degree of the corresponding pose. Note that in the Pointing'04 database, when the pitch angle is -90° or 90°, the yaw angle is always 0° due to physical limitations of the human head. Accordingly, in the rows corresponding to the -90° and 90° pitch angles, only the 0° yaw angle can have a non-zero description degree. Also, the intensity is not directly equal to the description degree. Instead, each image undergos a contrast stretching process to increase the image contrast for a better view.

3. Learning from MLD

After the MLD for each image is generated via the approach described in Section 2, the training set becomes $G = \{(\boldsymbol{x}_1, \boldsymbol{P}_1), \cdots, (\boldsymbol{x}_n, \boldsymbol{P}_n)\}$, where $\boldsymbol{P}_i$ is the MLD of $\boldsymbol{x}_i$. Suppose there are n_p different pitch angles and n_y different yaw angles, the pose determined by the j-th pitch angle and the k-th yaw angle is denoted by $\boldsymbol{y}_{jk}$. Then, $\boldsymbol{P}_i$ can be represented by an $n_p \times n_y$ matrix with its element at the j-th row and k-th column to be $d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}}$, i.e., the description degree of $\boldsymbol{y}_{jk}$ to $\boldsymbol{x}_i$.

Suppose the instance space is $\mathcal{X} = \mathbb{R}^q$, the label space is $\mathcal{Y} = \{\boldsymbol{y}_{jk}; j = 1, \ldots, n_p, k = 1, \ldots, n_y\}$. Then, the goal is to learn a conditional mass function $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ from G, where $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{y} \in \mathcal{Y}$, and $\boldsymbol{\theta}$ is the parameter vector. The conditional mass function is determined by finding the $\boldsymbol{\theta}$ that can generate an MLD similar to $\boldsymbol{P}_i$ given the instance $\boldsymbol{x}_i$. There are different criteria to measure the distance or similarity between two distributions [1], e.g., the discrete Jeffrey's divergence between two distributions $\boldsymbol{Q}_a$ and $\boldsymbol{Q}_b$ is defined by

$$D_J(\boldsymbol{Q}_a \| \boldsymbol{Q}_b) = \sum\nolimits_i (\boldsymbol{Q}_a^i - \boldsymbol{Q}_b^i) \ln \frac{\boldsymbol{Q}_a^i}{\boldsymbol{Q}_b^i}, \quad (2)$$

where $\boldsymbol{Q}_a^i$ and $\boldsymbol{Q}_b^i$ are the i-th element of $\boldsymbol{Q}_a$ and $\boldsymbol{Q}_b$, respectively. One possible problem of the definition in Eq. (2) is that the relationship among different elements of the distribution is not considered. However, such relationship is extremely important for MLD since it is generated based on the correlation among the neighboring labels. Thus we propose the weighted Jeffrey's divergence defined by

$$D_{wJ}(\boldsymbol{Q}_a \| \boldsymbol{Q}_b) = \sum\nolimits_{i,j} \lambda_{ij} (\boldsymbol{Q}_a^i - \boldsymbol{Q}_b^j) \ln \frac{\boldsymbol{Q}_a^i}{\boldsymbol{Q}_b^j}, \quad (3)$$

where the weight λ_{ij} models the relationship between the i-th element and the j-th element in the distribution.

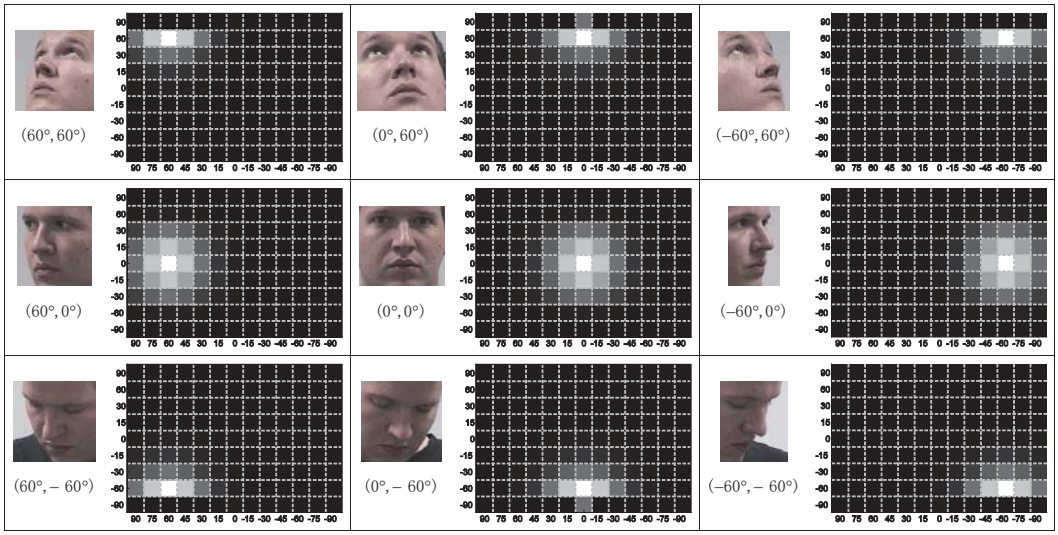


Figure 1. Typical head poses from the Pointing'04 database together with their MLDs generated by Eq. (1).

If the weighted Jeffrey's divergence defined in Eq. (3) is used to measure the distance between two bivariate MLDs, then the best parameter vector $\boldsymbol{\theta}^*$ is determined by

$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_i D_{wJ}(\boldsymbol{P}_i \| p(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{\theta})) \\ &= \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{\substack{i,j,k,\\ l,m}} \lambda_{jklm}(d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}} - p(\boldsymbol{y}_{lm}|\boldsymbol{x}_i;\boldsymbol{\theta})) \\ &\quad (\ln d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}} - \ln p(\boldsymbol{y}_{lm}|\boldsymbol{x}_i;\boldsymbol{\theta})) \\ &= \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{\substack{i,j,k,\\ l,m}} \lambda_{jklm}(p(\boldsymbol{y}_{lm}|\boldsymbol{x}_i;\boldsymbol{\theta}) \ln p(\boldsymbol{y}_{lm}|\boldsymbol{x}_i;\boldsymbol{\theta}) - \\ &\quad d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}} \ln p(\boldsymbol{y}_{lm}|\boldsymbol{x}_i;\boldsymbol{\theta}) - p(\boldsymbol{y}_{lm}|\boldsymbol{x}_i;\boldsymbol{\theta}) \ln d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}}), \end{aligned} \tag{4}$$

where $\boldsymbol{y}_{jk} \in \mathcal{Y}$ is a pose in the ground truth MLD $\boldsymbol{P}_i$, $\boldsymbol{y}_{lm} \in \mathcal{Y}$ is a pose in the MLD predicted by $p(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{\theta})$, and λ_{jklm} is the weight between $\boldsymbol{y}_{jk}$ and $\boldsymbol{y}_{lm}$. Exploring all possible combinations over the indices j, k, l and m is both computationally inefficient and unnecessary. It is reasonable to assume that only the neighboring poses are correlated. Accordingly, only the weights for the neighboring poses are non-zero. The value of λ_{jklm} relies on the definition of "neighborhood" among the poses. For instance, if λ_{jklm} is defined as

$$\lambda_{jklm} = \begin{cases} 1, & \text{if } j = l \wedge k = m; \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

i.e., the neighborhood of a pose $\boldsymbol{y}_{jk}$ only includes itself, then the weighted Jeffrey's divergence degenerates to the standard Jeffrey's divergence defined by Eq. (2).

More elaborately, λ_{jklm} can be defined as

$$\lambda_{jklm} = \begin{cases} \frac{1}{\Lambda_{jk}} \exp\left(\frac{\|\boldsymbol{y}_{jk}-\boldsymbol{y}_{lm}\|^2}{-\delta}\right), & \text{if } |j-l| \le 1 \wedge \\ & |k-m| \le 1; \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where $\Lambda_{jk} = \sum_{\substack{|j-l|\le 1\\ |k-m|\le 1}} \exp\left(\frac{\|\boldsymbol{y}_{jk}-\boldsymbol{y}_{lm}\|^2}{-\delta}\right)$ is a normalization factor that makes sure $\sum_{l,m} \lambda_{jklm} = 1$, and δ is a parameter that controls the influence of the distance between $\boldsymbol{y}_{jk}$ and $\boldsymbol{y}_{lm}$ to λ_{jklm}. This means that the original Jeffrey's divergence is locally smoothed within the nine-point neighborhood. The contribution of each point (l, m) in the neighborhood to the divergence is determined by the distance from $\boldsymbol{y}_{jk}$ to $\boldsymbol{y}_{lm}$.

As to the form of $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, similar to the work of Geng et al. [5, 6], we assume it to be a maximum entropy model, i.e.,

$$p(\boldsymbol{y}_{lm}|\boldsymbol{x}_i;\boldsymbol{\theta}) = \frac{1}{\Gamma_i} \exp\left(\sum_r \theta_{lmr} \boldsymbol{x}_i^r\right), \tag{7}$$

where $\Gamma_i = \sum_{l,m} \exp\left(\sum_r \theta_{lmr} \boldsymbol{x}_i^r\right)$ is the normalization factor, $\boldsymbol{x}_i^r$ is the r-th feature of $\boldsymbol{x}_i$, and θ_{lmr} is an element in $\boldsymbol{\theta}$ corresponding to the pose $\boldsymbol{y}_{lm}$ and the r-th feature. Substituting Eq. (7) into Eq. (4) yields the target function

$$\begin{aligned} T(\boldsymbol{\theta}) &= \sum_i \ln \Gamma_i + \sum_{\substack{i,j,k,\\ l,m}} \lambda_{jklm}[\frac{1}{\Gamma_i} \exp(\sum_r \theta_{lmr} \boldsymbol{x}_i^r) \\ &\quad (\sum_r \theta_{lmr} \boldsymbol{x}_i^r - \ln \Gamma_i - \ln d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}}) - \\ &\quad d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}} \sum_r \theta_{lmr} \boldsymbol{x}_i^r]. \end{aligned} \tag{8}$$

The minimization of the target function $T(\boldsymbol{\theta})$ can be effectively solved by the limited-memory quasi-Newton method

L-BFGS [13]. The basic idea of L-BFGS is to avoid explicit calculation of the inverse Hessian matrix used in the Newton method. Instead, L-BFGS approximates the inverse Hessian matrix by an iteratively updated matrix without actually storing the full matrix. The computation of L-BFGS is mainly related to the first-order gradient of the target function, which can be obtained through

$$\begin{aligned}\frac{\partial T(\boldsymbol{\theta})}{\partial \theta_{lmr}} &= \sum_{\substack{i,j,k,\\ l,m}} \lambda_{jklm}[p_{ilm}\boldsymbol{x}_i^r(1-p_{ilm}) \\ &\quad (\sum_r \theta_{lmr}\boldsymbol{x}_i^r - \ln\Gamma_i - \ln d_{\boldsymbol{x}_i}^{\boldsymbol{y}_{jk}} + 1)] \\ &\quad - \sum_i \boldsymbol{x}_i^r(1-p_{ilm}), \end{aligned} \tag{9}$$

where $p_{ilm} = \frac{1}{\Gamma_i}\exp\left(\sum_r \theta_{lmr}\boldsymbol{x}_i^r\right)$.

After the optimal parameter vector $\boldsymbol{\theta}^*$ is finally learned, given a test image $\boldsymbol{x}'$, its MLD is first predicted by $p(\boldsymbol{y}|\boldsymbol{x}';\boldsymbol{\theta}^*), \boldsymbol{y} \in \mathcal{Y}$. Then, the pose corresponding to the maximum description degree in the MLD is output as the predicted pose for $\boldsymbol{x}'$.

4. Experiments

4.1. Methodology

The data set used in the experiments is the Pointing'04 database [7]. The head poses in this database are discretized into 13 yaw angles $\{-90°, -75°, -60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°, 75°, 90°\}$ and 9 pitch angles $\{-90°, -60°, -30°, -15°, 0°, 15°, 30°, 60°, 90°\}$. A pose is represented by the combination of a yaw angle and a pitch angle. Specially, when the pitch angle is $-90°$ or $90°$, the yaw angle is always $0°$. Thus, there are in total $13 \times 7 + 2 = 93$ poses involved in the data set. The images are taken from 15 different human subjects in two different times, resulting in $93 \times 15 \times 2 = 2790$ face images. For each image, the bounding box of the face region is provided in the database. The face images are normalized to the same size of 32×32 pixels, and the features are extracted by the Histogram of Oriented Gradients (HOG) [3] with the 3×3 pixel cells.

Two versions of the MLD methods are tested in the experiments. The first is denoted by MLD-J, which uses the inter-pose weights defined by Eq. (5), i.e., using the standard Jeffrey's divergence in the target function. The second is denoted by MLD-wJ, which uses the inter-pose weights defined by Eq. (6), i.e., using the weighted Jeffrey's divergence in the target function. The comparison between MLD-J and MLD-wJ may reveal the effects of utilizing the correlation among the neighboring poses. The MLD of each training image is generated by Eq. (1), and if not explicitly stated, $\boldsymbol{\Sigma} = \begin{bmatrix}\tau^2 & 0\\ 0 & \tau^2\end{bmatrix}$, where τ is the finest granularity of the pose angles (in the Pointing'04 case, $\tau = 15$).

The MLD methods are compared via five-fold cross-validation with several state-of-the-art head pose estimation methods including linear/kernel PLS [11], linear/kernel SVM and linear/kernel SVR [9]. For each compared method, several parameter configurations are tested and the best performance is reported. For MLD-wJ, the δ in Eq. (6) is set to $0.4\tau^2$. The number of factors is set to 40 for kernel PLS and 25 for linear PLS. Kernel PLS uses the RBF kernel with the width 0.05. Both kernel SVM and kernel SVR uses the RBF kernel with the width 0.01. Moreover, some previously reported head pose estimation results [16, 4] on the Pointing'04 database are also provided as references.

The algorithms are compared by two groups of evaluation measures. One group are the regression measures, i.e., the mean absolute error (MAE) between the predicted pose and the "ground truth" pose. The other group are the classification measures, i.e., the accuracy of the predicted pose with respect to the "ground truth" pose. Each group includes three measures, one for the yaw angle, one for the pitch angle, and the last for both of them (denoted by "yaw+pitch"). The MAE of yaw+pitch is calculated by the Euclidean distance between the predicted (yaw, pitch) pair and the "ground truth" (yaw, pitch) pair. For the classification measures, the regression methods (linear PLS, kernel PLS, linear SVR, and kernel SVR) predict the class closest to their output pose. The accuracy of yaw+pitch is calculated by regarding each (yaw, pitch) pair as a class.

4.2. Results

The head pose estimation results on the Pointing'04 database are compared in Table 1. Each result is represented by the mean value $\pm$ standard deviation of the five-fold cross-validation. The best mean performance on each measure is highlighted by boldface. In addition to the eight methods compared in the exactly same experimental settings, some previously reported head pose estimation results [16, 4] on the Pointing'04 database are also provided in the lower part of Table 1 as references. Note that some of the previous results might be obtained with different experimental settings, such as differences in feature extractor, validation protocol and embedding dimensionality, which are detailed in the footnotes of the table. As can be seen from Table 1, both MLD-wJ and MLD-J perform significantly better than all other compared methods on all evaluation measures. Moreover, by more effectively utilizing the relationship among the neighboring poses, MLD-wJ performs significantly better than MLD-J on all evaluation measures.

In greater detail, the confusion matrices (in %) of MLD-wJ on the yaw and pitch angles are shown in Fig. 2. As can

Table 1. Head Pose Estimation Results on the Pointing'04 Database.

Method	MAE			Accuracy		
	Yaw	Pitch	Yaw+Pitch	Yaw	Pitch	Yaw+Pitch
MLD-wJ	$\mathbf{4.24^\circ \pm 0.17^\circ}$	$\mathbf{2.69^\circ \pm 0.15^\circ}$	$\mathbf{6.45^\circ \pm 0.29^\circ}$	$\mathbf{73.30\% \pm 1.36\%}$	$\mathbf{86.24\% \pm 0.97\%}$	$\mathbf{64.27\% \pm 1.82\%}$
MLD-J	$5.02^\circ \pm 0.31^\circ$	$3.54^\circ \pm 0.30^\circ$	$7.94^\circ \pm 0.53^\circ$	$67.96\% \pm 2.21\%$	$81.51\% \pm 1.67\%$	$55.66\% \pm 3.28\%$
Kernel PLS	$5.79^\circ \pm 0.32^\circ$	$4.83^\circ \pm 0.29^\circ$	$9.66^\circ \pm 0.33^\circ$	$64.48\% \pm 1.79\%$	$78.35\% \pm 1.11\%$	$51.47\% \pm 1.64\%$
Linear PLS	$9.28^\circ \pm 0.48^\circ$	$8.92^\circ \pm 0.56^\circ$	$15.88^\circ \pm 0.79^\circ$	$46.49\% \pm 2.80\%$	$60.54\% \pm 2.52\%$	$28.10\% \pm 3.28\%$
Kernel SVM	$6.83^\circ \pm 0.36^\circ$	$5.91^\circ \pm 0.31^\circ$	$11.87^\circ \pm 0.39^\circ$	$57.17\% \pm 2.12\%$	$68.24\% \pm 1.71\%$	$34.23\% \pm 2.05\%$
Linear SVM	$8.30^\circ \pm 0.57^\circ$	$8.16^\circ \pm 0.48^\circ$	$14.91^\circ \pm 0.54^\circ$	$50.54\% \pm 2.81\%$	$57.67\% \pm 2.23\%$	$23.80\% \pm 1.75\%$
Kernel SVR	$6.89^\circ \pm 0.47^\circ$	$6.59^\circ \pm 0.62^\circ$	$11.99^\circ \pm 0.76^\circ$	$60.22\% \pm 3.11\%$	$71.72\% \pm 2.22\%$	$44.73\% \pm 3.46\%$
Linear SVR	$8.33^\circ \pm 0.55^\circ$	$8.27^\circ \pm 0.35^\circ$	$14.50^\circ \pm 0.68^\circ$	$52.08\% \pm 3.16\%$	$64.70\% \pm 1.38\%$	$35.16\% \pm 3.08\%$
Stiefelhagen [17][1]	9.5°	9.7°	—	52.0%	66.3%	—
Human Performance [8][2]	11.8°	9.4°	—	40.7%	59.0%	—
Gourier (Associative Memories) [8][3]	10.1°	15.9°	—	50.0%	43.9%	—
Tu (High-order SVD) [18][4]	12.9°	17.97°	—	49.25%	54.84%	—
Tu (PCA) [18][4]	14.11°	14.98°	—	55.20%	57.99%	—
Tu (LEA) [18][4]	15.88°	17.44°	—	45.16%	50.61%	—
Voit [19]	12.3°	12.77°	—	—	—	—
Li (PCA) [12][5]	26.9°	35.1°	—	—	—	—
Li (LDA) [12][5]	25.8°	26.9°	—	—	—	—
Li (LPP) [12][5]	24.7°	22.6°	—	—	—	—
Li (Local-PCA) [12][5]	24.5°	37.6°	—	—	—	—
Li (Local-LDA) [12][5]	19.1°	30.7°	—	—	—	—
Li (Local-LPP) [12][5]	29.2°	40.2°	—	—	—	—
Foytik (Two-layer Phase Cong.) [4][6]	13.0°	—	—	—	—	—

[1] Used 80% of Pointing'04 images for training and 10% for evaluation;
[2] Human performance with training;
[3] Best results over different reported methods;
[4] Better results have been obtained with manual localization;
[5] Results for 32-dim embedding;
[6] Trained on the FacePix(30) database and tested with Pointing'04 yaw angles.

be seen, for both yaw and pitch, MLD-wJ tends to produce more incorrect estimations at larger angles, e.g., $\pm 90^\circ$. The reason might be that the human subjects often pose differently at those extreme yaw or pitch angles. One additional reason for the relatively poor performance at the -90° pitch angle is that there are much fewer training examples for the $\pm 90^\circ$ pitch angles than for other angles. Fortunately, for both yaw and pitch, most incorrect predictions are adjacent to the ground truth angles.

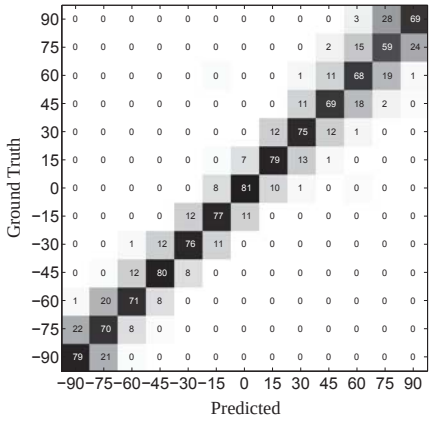
When generating the MLDs for the training images by Eq. (1), the covariance matrix $\mathbf{\Sigma}$ determines how the neighboring poses are related to the ground truth pose. Suppose $\mathbf{\Sigma} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$, then the standard deviation σ becomes an indicator of the relationship between the neighboring poses and the ground truth pose. The larger σ is, the more neighboring poses are related to the ground truth pose. In order to discover the usefulness of the neighboring poses, MLD-wJ is tested on the MLDs generated with different values of σ changing from 0 to 4τ with 0.5τ as the interval. Note that $\sigma = 0$ corresponds to the special case where the description degree of the ground truth pose is 1 while the description degrees of all other poses are 0. The result is shown in Fig. 3. As can be seen, too small or too large σ may both lead to performance deterioration. Thus, a proper choice of σ is important for good performance of the MLD methods. Roughly speaking, $1.0\tau \leq \sigma \leq 1.5\tau$ is a good choice.

5. Conclusion

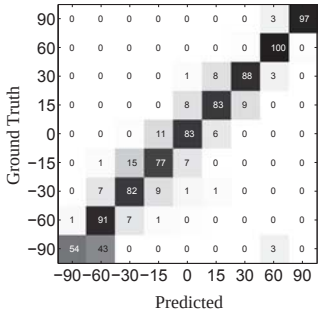
This paper is motivated by the inaccurate pose labels in the training set for head pose estimation algorithms. In order to solve this problem, we propose to associate a soft pose label called multivariate label distribution (MLD) to each image. The MLD covers a neighborhood centered at the original (perhaps inaccurate) pose label. Using MLD instead of the explicit hard pose label can not only alleviate the negative influence of inaccurate pose labels, but also boost the training examples for each pose without actually increasing the total amount of training examples. Learning from MLD is implemented by minimizing the weighted Jeffrey's divergence between the predicted MLD and the ground truth MLD. The proposed MLD-based methods are compared with several state-of-the-art head pose estimation algorithms on the Pointing'04 database and achieve significantly better results.

References

[1] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1:300–307, 2007. 2



(a)



(b)

Figure 2. The confusion matrices (in %) of MLD-wJ on (a) the yaw angles and (b) the pitch angles.

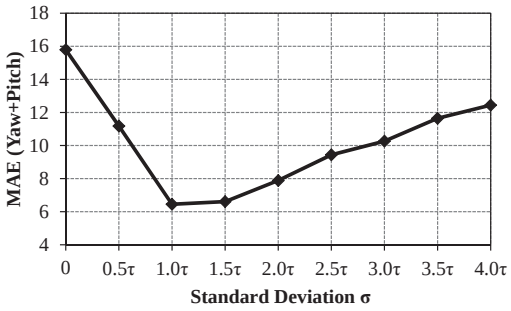


Figure 3. The MAE (Yaw+Pitch) of MLD-wJ with different standard deviations which are used to generate the MLDs.

[2] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. J. V. Gool. Random forests for real time 3D face analysis. *International Journal of Computer Vision*, 101(3):437–458, 2013. 1

[3] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010. 4

[4] J. Foytik and V. K. Asari. A two-layer framework for piecewise linear manifold-based head pose estimation. *International Journal of Computer Vision*, 101(2):270–287, 2013. 1, 4, 5

[5] X. Geng, K. Smith-Miles, and Z.-H. Zhou. Facial age estimation by learning from label distributions. In *Proc. 24th AAAI Conf. Artificial Intelligence*, pages 451–456, Atlanta, GA, 2010. 3

[6] X. Geng, C. Yin, and Z.-H. Zhou. Facial age estimation by learning from label distributions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(10):2401–2412, 2013. 3

[7] N. Gourier, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial features. In *Proc. Pointing 2004, ICPR, Int'l Workshop on Visual Observation of Deictic Gestures*, Cambridge, UK, 2004. 1, 2, 4

[8] N. Gourier, J. Maisonnasse, D. Hall, and J. L. Crowley. Head pose estimation on low resolution images. In *Proc. 1st Int'l Workshop Classification of Events, Activities and Relationships*, pages 270–280, Southampton, UK, 2006. 1, 5

[9] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang. Head pose estimation: Classification or regression? In *Proc. 19th Int'l Conf. Pattern Recognition*, pages 1–4, Tampa, FL, 2008. 1, 4

[10] S. Gurbuz, E. Oztop, and N. Inoue. Model free head pose estimation using stereovision. *Pattern Recognition*, 45(1):33–42, 2012. 1

[11] M. A. Haj, J. Gonzàlez, and L. S. Davis. On partial least squares in head pose estimation: How to simultaneously deal with misalignment. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2602–2609, Providence, RI, 2012. 1, 4

[12] Z. Li, Y. Fu, J. Yuan, T. S. Huang, and Y. Wu. Query driven localized linear discriminant models for head pose estimation. In *Proc. IEEE Int'l Conf. Multimedia and Expo*, pages 1810–1813, Beijing, China, 2007. 1, 5

[13] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(3):503–528, 1989. 4

[14] J. Lu and Y.-P. Tan. Ordinary preserving manifold analysis for human age and head pose estimation. *IEEE Trans. Human-Machine Systems*, 43(2):249–258, 2013. 1

[15] B. Ma, X. Chai, and T. Wang. A novel feature descriptor based on biologically inspired feature for head pose estimation. *Neurocomputing*, 115:1–10, 2013. 1

[16] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4):607–626, 2009. 1, 4

[17] R. Stiefelhagen. Estimating head pose with neural networks - results on the Pointing04 ICPR workshop evaluation data. In *Proc. Pointing 2004 Workshop: Visual Observation of Deictic Gestures*, Cambridge, UK, 2004. 1, 5

[18] J. Tu, Y. Fu, Y. Hu, and T. S. Huang. Evaluation of head pose estimation for studio data. In *Proc. 1st Int'l Workshop Classification of Events, Activities and Relationships*, pages 281–290, Southampton, UK, 2006. 1, 5

[19] M. Voit, K. Nickel, and R. Stiefelhagen. Neural network-based head pose estimation and multi-view fusion. In *Proc. 1st Int'l Workshop Classification of Events, Activities and Relationships*, pages 291–298, Southampton, UK, 2006. 1, 5

[20] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2879–2886, Providence, RI, 2012. 1