Multimed Tools Appl (2016) 75:9395–9408
DOI 10.1007/s11042-016-3297-2

Transfer useful knowledge for headpose estimation from low resolution images

Ping'an Li[1] · Yufeng Li[1] · Lixin Tan[1]

Received: 29 September 2015 / Revised: 17 January 2016 / Accepted: 22 January 2016 /
Published online:18 February 2016

Abstract The knowledge of where a person is looking is useful in human computer interaction as well as human behavior analysis. Headpose estimation from low resolution images is still a challenge problem due to noisy feature representation for low resolution images. In this paper, we investigate *transfer learning* technique to conquer the weakness of the apperance-based feature of humans head-pose when their relative locations to far-field cameras are different. We evaluate our methods on public datasets which prove the efficiency of our proposed method.

Keywords Headpose estimation · Low resolution images · Transfer learning

1 Introduction

Tracking of people and localizing their head orientation or gaze direction provides additional support to human behavior studies. For example, people nod to indicate that they understand what is being said. Computing the direction of one's head/eye orientation enables the identification of who the intended target of a conversation is. However, most contemporary methods, which estimated a person's focus of attention employing headpose and eye-gaze cues, rely on high resolution images, close-range camera and a highly constrained context.

From the visual analysis viewpoint, focus-of-attention (FOA) by estimating the head pose in a dynamic environment is challenging due to its unstructured behavior, possiblity of occlusion, low resolusion image, not linear relationship between headpose angle and the target location, etc.

✉ Ping'an Li
pingan.li.76@gmail.com

[1] School of Electronic Engineering, Hunan College of Information, Hunan, China

In this work, a multi-cameras system is used to overcome the occulusion problem. Then the robust *Kullback Leibler Divergence* [21] and *Covariance Descriptor* [29] features are proposed for representing low resolusion images. Finally, we combine a tracker-based pose estimator and an appearance-based head pose predictor to do head pose estimation which would eventually lead us to estimate the personality of the participants.

Moreover, under a dynamic and unstructured setting, persons move freely which might cause the changing in their appreance. The feature distribution changes dramatically with respect to the relative location of far-field cameras. Therefore, the appearance-based feature model of human head-pose needs to be updated according to their relative location with the camera. Conventional machine learning algorithms perform poorly under this variant distribution setting. Consequently, we introduce the *transfer learning* concept to conquer this challenging problem. By doing this, we only need to collect a small number of samples in different locations in the room and use *transfer learning* to estimate the head pose in all other locations of the room.

The rest of this proposal is organized as follows. In Section 2, we review the state of the art on head pose estimation, classification and transfer learning. We discuss of our method in details in Section 3. Section 4 illustrates some results. Section 5 draws the concluding remarks of this paper.

2 Related work

During the past decades, researchers modeled the human behaviour using multimodal approaches based on video and audio. Generally speaking, most of the works firstly extract some discriminant features from humans, especially visual features of humans' head/face and audio features of humans' speaking activity. Then they fuse the data to model some specific behaviours such as interest, puzzlement and frustration, etc.

2.1 Head pose estimation

For human head pose estimation, what we need to do is to estimate the three head rotations with respect to the cameras which are represented by the *yaw (or pan) angle, pitch (or tilt) angle and roll angle* (as seen in Fig. 1).

Several popular methods [7], such as *Appearance template* methods, *Tracking* methods, *Geometric* methods are used in the research area of human head pose estimation. *Appearance* template methods compare a new image of a head to a set of exemplars to find the most similar view which is always the shortest distance between the new image and the exemplars. *Tracking* methods combine the tracking of the person and estimating head pose simultaneously from video frames. *Geometric* methods use the location of particular features such as eyes and mouth to determine the pose from their relative configuration.

For high-resolution images, there have been already some effort to do head pose estimation and gaze extraction. Perez and Cordoba [1] investigated gaze recognition through tracking of the eyeballs. Gee and Cipolla's [15] gaze recognition method was based on 3D geometric relationship of facial features. However, all the features extracted for these methods are based on the high-resolution image. Most cameras in public areas are subject to complex condition and low-resolution images which make these methods unreliable in practise.

For low-resolution images, Robertson and Reid [25] proposed a skin and hair color based feature using color histograms for head pose estimation. However, this approach relied critically on good segmentation of skin and non-skin regions of a head image. Ba and Odobez

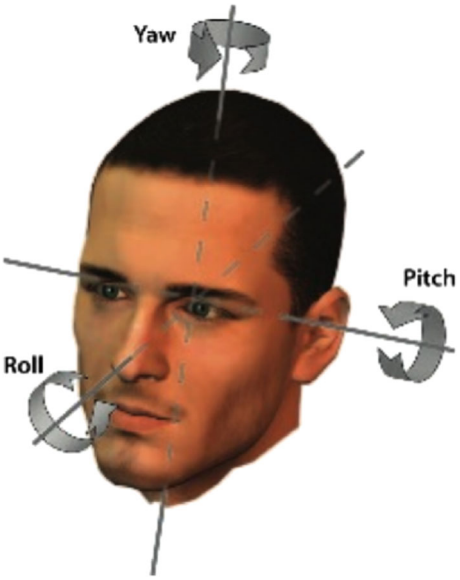


Fig. 1 Three degree-of-freedom of a human head

[3, 4, 26] proposed a method for low-resolution head pose estimation. However, their methods used a fixed camera, which could not solve the occlusion problem. Also, they modeled tracking and pose classification as two paired tasks in a single framework which has the problem that the head pose estimation accuracy is affected by the accuracy of the tracking results. Tosato *et al.* [28] proposed the array of variances feature and classified the feature on Riemannian manifolds. The array of variance feature could describe visual object at low resolution better than other methods. However, the classification on Riemannian manifold rather than on Euclidean manifold is time consuming and probably could not be used in real-time systems. Yan *et al.* [32, 33, 37] proposed multi-task learning and transfer learning for headpose estimation.

2.2 Classification

Appearance-based head pose estimation can be translated into a machine learning problem. There are several basic learning algorithms that are widely used in machine learning area, such as *K-Nearest-Neighbor (KNN), Boosting and Support Vector Machine (SVMs)*.

K-Nearest-Neighbor (KNN) algorithm is a method for classifying objects based on closest training examples in the feature space. The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is the most frequent among the k training samples nearest to that query point.

$Boosting$ refers to an effective method of producing an accurate prediction rule by combining rough and moderate inaccurate rules of thumb. Freund and Schapire [14] proposed $AdaBoost$ which solved many practical difficulties of earlier boosting algorithms. $AdaBoost$ calls a given weak or base learning algorithm repeatedly in a series of rounds

$t = 1, ..., T$. Once the weak hypothesis h_t has been received. AdaBoost chooses a parameter α_t according to the error. The final hypothesis H is a weighted majority vote of T weak hypotheses where α_t is the weight assigned to h_t.

Support Vector Machines (SVMs) [13] consider a d-dimensional feature space F which is a subset of R^d and is spanned by a mapping φ. In a support vector (SV) setting, any φ corresponds to a Mercer Kernel $k(x, x') =< \varphi(x) \cdot \varphi(x') >$ implicitly computing the dot product in F. The goal of SVMs is to find some separating hyperplane described by a vector w in the feature space F. Given training set pairs $(x_i, y_i), i = 1, \ldots, l, x_i \in R^d$ and $y_i \in \{-1, 1\}$, the classification requires the solution of the following optimization problem:

$$\begin{aligned} &= \min_{w,b,\zeta} \tfrac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ &s.t. y_i (w^T \varphi(x_i) + b) \geq 1 - \xi_i \\ &\xi_i \geq 0 \end{aligned} \tag{1}$$

where ξ_i represents the slack variable and $\sum_{i=1}^{l} \xi_i$ measures the total classification error. The objective function seeks a decision boundary that achieves a small classification error and meanwhile creates a large margin, with two goals balanced by the scalar cost factor C.

2.3 Transfer learning

Traditional machine learning approches already achieve significant success in computer vision area including classification, regression and clustering. However, traditional machine learning algorithms are based on the assumption that training and testing data share the same feature space and the same distribution. When the training and testing data distributions are different, the accuracy of classification drops significantly. In this case, transfer learning between different domains is desirable. Transfer learning assumes that training and testing data could be from different domains and distributions. It is motivated by the fact that people can intelligently apply knowledge learning previously to solve new problems faster. The target of transfer learning is to find some commom property which is shared between the training (or source) and test (or target) domain. Some representative work used for event detection, egocentric activity recognition and multiview action recognition are [34–36]

In transfer learning, we have three main research issues: (1) what to transfer, (2) how to transfer, and (3) when to transfer [22]. "What to transfer" solves the problem of which part of knowledge can be transferred across domains or tasks. After discovering which knowledge can be transferred, learning algorithms are developed to do "how to transfer". "When to transfer" asks under which situation the knowledge could be transfered in case some negative transfer could even hurt the performance of the target domain.

There are several approaches to transfer learning. *Instance-transfer* [8, 17, 23, 27, 39, 40] is to re-weight some labeled data in the source domain for using in the target domain due to the assumption that certain parts of the data in the source domain can be reused for the target domain. *Feature-reperesentation-transfer* [2, 9] is to find a "good" feature representation that reduces the difference between the source target domains and the error of classification and regression models. *Parameter-transfer* [5, 12] is to discover shared parameters or priors between the source domain and target domain models which can benefit from transfer learning. *Relational-knowledge-transfer* [20] is to build a mapping of relational knowledge between the source domain and the target domain.

For our situation, we wish to minimize the difference among the changes of appearance features when people stand in different locations relative to the cameras. We use an adaptive multiple kernel learning method which belongs to the *instance-transfer learning* category. Specifically, for each type of local features, we train a set of SVM classifiers based on a combined training set from the two domains by using multiple base kernels of different kernel types and parameters, which are further fused with equal weights to obtain an average classifier. The objective function minimizes the structural risk functional and the mismatch of data distributions between the source and the target domain simultaneously. The next section presents our solution in detail.

3 Transfer learning for head pose estimation

The head-pose estimation process in the party scenario involves four steps. (i) Head feature representation. (ii) Head pose classification.

3.1 Head feature representation

Face crops for the four camera views, obtained from the head localization procedure are used for head pose prediction. We focus more attention on where are the persons looking at, especially for the humans' head horizontal rotation. Therefore, we discretized the space of possible head rotations into 24 classes, 8 classes for pan (horizontal rotation) and 3 classes for tilt (vertical rotation). We resize the head pose image to 20×20 pixels for one cropping and make four image as one panorama image. A template matching method is used for the head pose estimation. In order to solve the problem of occlusion, we combine four camera images as a panorama image to extract features and feed into the classifier to do the prediction. This method improves the accuracy of estimation compared with using only one camera output.

For low-level feature representation, there are two kinds of representation methods. The first category is a sparse representation which consists of a set of representative local regions obtained by an interest point detection algorithm. Reliable interest points should contain valuable information about the local image content and should remain stable under changes, such as viewpoint and illumination changes. Histogram-based representations of gradients, such as scale-invariant feature transform descriptors (SIFT) [19] and shape contexts belong to this category.

The second category is a dense representation which consists of a set of representative regions obtained inside a detection window. The entire image is scanned densely and a learned classifier of object model is evaluated. Intesity templates and principal component analysis (PCA) coeffients belong to this category.

We present two kinds of low-level features for human head representation which are *Kullback Leibler Divergence* [21] and *Covariance Descriptor* [29].

It is critical to represent the head pose based on the good separation of background, hair and skin/non-skin pixels. The idea is to compute each input image pixel to a set of mean appearance regardless pose. We compute the *Kullback Leibler divergence* (KL) distance [21] between the input test image and the mean template image for every pose. We choose the maximum value of KL distance for each pixel in RGB channel as the feature.

$$\theta_{i,j} = \max_{c} \left\{ \max_{RGB} \left\{ p^{c}_{i,j} * \log \frac{p^{c}_{i,j}}{q^{c}_{i,j}} \right\} \right\} \tag{2}$$

where i, j represents each pixel, $p^c_{i,j}$ and $q^c_{i,j}$ are the mean image and test image pixel intensity value. $\theta_{i,j}$ is the maximum coefficient from all 24 classes and RGB channels for each pixel.

We also investigate another feature representation called *covariance descriptor* [29]. For low resolution images, the number of features that can be extracted are relatively small and quite unreliable.

Let I be a digital image and F be the $W \times H \times d$ dimensional feature image extracted from I

$$F(x, y) = \varphi(I, x, y) \tag{3}$$

where φ can be any mapping such as intensity, color, gradients, filter responses, etc. The region R in the image can be represented with a $d \times d$ convariance matrix of the feature points

$$C_R = \frac{1}{S-1}\sum_{i=1}^{S}(z_i - \mu)(z_i - \mu)^T \tag{4}$$

$\{z_i\}_{i=1..S}$ are the d-dimensional feature points inside a region, μ is the mean of the points. Figure 2 shows the covariance descriptor of a region in the image.

Integral images are intermediate image representations used for the fast calculation of region sums [30]. Let P be the $W \times H \times d$ tensor of integral images

$$P(x', y', i) = \sum_{x \le x', y \le y'} F(x, y, i) \qquad i = 1...d \tag{5}$$

and Q be the $W \times H \times d \times d$ tensor of second-order of integral images

$$Q(x', y', i, j) = \sum_{x \le x', y \le y'} F(x, y, i)F(x, y, j) \qquad i, j = 1...d \tag{6}$$

Then $P_{x,y}$ is the d-dimensional vector and $Q_{x,y}$ is the $d \times d$ dimensional matrix

$$P_{x,y} = [P(x, y, 1)...P(x, y, d)]^T \tag{7}$$

$$Q_{x,y} = \begin{bmatrix} Q(x, y, 1, 1)...Q(x, y, 1, d) \\ ... \\ Q(x, y, d, 1)...Q(x, y, d, d) \end{bmatrix} \tag{8}$$

for a region R. We could fast calculate the covariance of a region $R(x', y'; x'', y'')$ using integral image as

$$\begin{aligned} &C_{R(x',y';x'',y'')} \\ &= \tfrac{1}{S-1}[Q_{x'',y''} + Q_{x'-1,y'-1} - Q_{x'',y'-1} - Q_{x'-1,y''} \\ &\quad -\tfrac{1}{S}(P_{x'',y''} + P_{x'-1,y'-1} - P_{x'',y'-1} - P_{x'-1,y''}) \\ &\quad (P_{x'',y''} + P_{x'-1,y'-1} - P_{x'',y'-1} - P_{x'-1,y''})^T] \end{aligned} \tag{9}$$

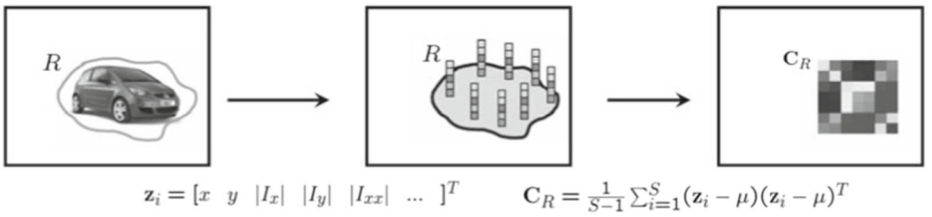


Fig. 2 Covariance descriptor

where (x', y') and (x'', y'') are the upper-left and bottom-right coordinator of a region in an image and $S = (x'' - x' + 1) \cdot (y'' - y' + 1)$. Therefore, after constructing integral images, the covariance of any rectangular region can be computed in $O(d^2)$ time.

The benefit of *covariance descriptor* is that it could combine several different features such as color, texture in a single descriptor.

3.2 Transfer learning on head-pose estimation

Assume $Pr(x, y)$ and $Pr'(x, y)$ are two different distributions, the objective of learning method is to minimize the *expected risk*

$$R[\Pr, \theta, l(x, y, \theta)] = E_{(x,y)\sim \Pr}[l(x, y, \theta)] \tag{10}$$

of a loss function $l(x, y, \theta)$ which depends on a parameter θ. Here, the notation $(x, y) \sim Pr$ means (x, y) belongs to the distribution $Pr(x, y)$.

In practice, we only observe examples (x, y) drawn from $Pr(x, y)$ which means we use empirical average

$$R_{emp}[Pr, \theta, l(x, y, \theta)] = \frac{1}{m} \sum_{i=1}^{m} l(x_i, y_i, \theta) \tag{11}$$

To avoid overfitting, we add a regularizer $\Omega(\theta)$ and minimize the following equation

$$R_{reg}[Pr, \theta, l(x, y, \theta)] = R_{emp}[Pr, \theta, l(x, y, \theta)] + \lambda \Omega(\theta) \tag{12}$$

where λ is the trade-off coefficient between loss function and regularizer.

Importance sampling is a general technique for estimating properties of a particular distribution, while only having samples generated from a different distribution rather than the distribution of interest. If we use *importance sampling*,

$$\begin{aligned} R[\Pr', \theta, l(x, y, \theta)] &= E_{(x,y)\sim \Pr'}[l(x, y, \theta)] \\ &= E_{(x,y)\sim \Pr'}[\Pr(x, y) \tfrac{\Pr'(x,y)}{\Pr(x,y)} l(x, y, \theta)] \\ &= E_{(x,y)\sim \Pr}[\tfrac{\Pr'(x,y)}{\Pr(x,y)} l(x, y, \theta)] \\ &= R[\Pr, \theta, \beta(x, y) l(x, y, \theta)] \end{aligned} \tag{13}$$

where $\beta(x, y) = \frac{\Pr'(x,y)}{\Pr(x,y)}$ is a reweighting factor for the training example. However, coefficients $\beta(x, y)$ are usually unknown, which means we need to estimate $\beta(x, y)$.

Sugiyama et. al [27] propose a *least-squares approach* to directly estimate this importance coefficients $\beta(x, y)$. They model the importance coefficients $\beta(x, y)$ by the linear model

$$\hat{\beta}(x) = \sum_{i=1}^{m} \alpha_i \phi_i(x) \tag{14}$$

where $\alpha = (\alpha_1 ... \alpha_m)$ are the parameters learned from data samples and $\phi_i(x)$ are the basis functions we need to choose. They use a least-squares approach to minimize $J(\alpha) = \frac{1}{2} \int \left(\hat{\beta}(x, y) - \beta(x, y)\right)^2 \Pr(x) dx$.

Then they formulate the problem as an optimization problem

$$\min_{\alpha} \frac{1}{2} \alpha^T \hat{H} \alpha - \hat{h}^T \alpha + \lambda \alpha, \quad s.t.\ \alpha \geq 0 \tag{15}$$

where $\hat{H}_{i,j} = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \phi_i(x_i^{tr})\phi_j(x_i^{tr})$ and $\hat{h}_l = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \phi_l(x_i^{tr})$, and n_{tr} and n_{te} represent the number of training and test samples. This method gives a closed-form solution.

Huang et. al [17] propose the *Kernel Mean Matching* method which does not need to estimate the density of the function directly. Let $\Phi : X \rightarrow F$ be a mapping into a feature space F and $\mu : P \rightarrow F$ be the expectation operateor

$$\mu(\Pr) = E_{x\sim \Pr(x)}[\Phi(x)] \tag{16}$$

Then we can infer a suitable β by solving the following optimization problem

$$\begin{aligned} &\min_{\beta} \left\| \mu(Pr') - E_{x\sim \Pr(x)}[\beta(x)\Phi(x)] \right\| \\ &s.t. \quad \beta(x) \geq 0 \\ &\qquad E_{x\sim \Pr(x)}[\beta(x)] = 1 \end{aligned} \tag{17}$$

In practice, we use empirical means instead of density distribution, then we have

$$\begin{aligned} &\left\| \frac{1}{m} \sum_{i=1}^{m} \beta_i \Phi(x_i) - \frac{1}{m'} \sum_{i=1}^{m'} \Phi(x_i') \right\|^2 \\ &= \frac{1}{m^2} \beta^T K \beta - \frac{2}{m^2} \kappa^T \beta + C \end{aligned} \tag{18}$$

Here $K_{ij} = k(x_i, x_j)$ and $\kappa_i = \frac{m}{m'} \sum_{j=1}^{m'} k(x_i, x_j')$, and C is a const factor.

The optimatization problem can be reformulated as a quadratic problem as following

$$\begin{aligned} &\min_{\beta} \frac{1}{2} \beta^T K \beta - \kappa^T \beta \\ &s.t. \quad \beta_i \in [0, B], \\ &\qquad \left| \sum_{i=1}^{m} \beta_i - m \right| \leq m\xi_i \end{aligned} \tag{19}$$

where B is the upbound of β and ξ_i is a slack variable.

Due to the hardness of kernel parameter choosen of SVM model, Rakotomamonjy et. al [24] proposed a multiple kernel method to simultaneously learn a kernel and the associated predictor in a supervised learning setting. They address the multiple kernel learning problem through a weighted 2-norm regularization formulation with an additional constraint on the weights that encourages sparse kernel combinations.

They define the kernel as a linear combination of M base kernels as

$$\begin{aligned} &K(x', x) = \sum_{m=1}^{M} d_m K_m(x, x') \\ &s.t. \quad d_m \geq 0, \sum_{m=1}^{M} d_m = 1 \end{aligned} \tag{20}$$

Then they formulate the optimization problem as

$$\begin{aligned} &\min_{d} \; T(d) \\ &s.t. \quad d_m \geq 0, \sum_{m=1}^{M} d_m = 1 \end{aligned} \tag{21}$$

where

$$T(d) = \begin{cases} \min\limits_{\{f\},b,\xi} \frac{1}{2}\sum\limits_{m}\frac{1}{d_m}\|f_m\|^2 + C\sum\limits_{i}\xi_i \\ s.t. \quad y_i\sum\limits_{m} f_m(x_i) + y_i b \geq 1-\xi_i \\ \xi_i \geq 0 \end{cases} \tag{22}$$

Here f_m is the m-th decision function, $\sum\limits_i \xi_i$ measures the total classification error, b is a constant factor, and *T(d)* is a traditional SVM format which can be solved by the optimization problem using a gradient method.

Recently, several adaptation methods for support vector machine (SVM) classifier were proposed in the video retrieval literature [18, 31, 38]. In order to make the SVM classifier adaptive to new domain, we formulate the target decision function for any sample x as

$$f^T(x) = \sum_{p=1}^{P}\gamma_p f_p(x) + \sum_{m=1}^{M} d_m w_m' \phi_m(x) + b \tag{23}$$

Where $f_p(x)$ are the prelearned classifiers trained based on the labeled data from both domains. γ_p and d_m are coefficients of prelearned classifiers and multiple kernels, respectively.

For transfer learning [10, 11], the first objective is to reduce the mismatch between the source and the target domain. Gretton et al. [16] propose a measurement method of two different distributions. The mismatch is measured by Maximum Mean Discrepancy(MMD) based on the distance between the means of sample from source domain and target domain in the Reproducing Kernel Hilbert Space(RKHS) namely:

$$DIST(D^S, D^T) = \Omega(d) = \| \frac{1}{n_S}\sum_{i=1}^{n_S}\phi(x_i^S) - \frac{1}{n_T}\sum_{i=1}^{n_T}\phi\left(x_i^T\right) \|_H \tag{24}$$

where x_i^S and x_i^T are the samples from the source and target domains, respectively.

The second objective is to minimize the structural risk functional. If we combine these two objectives, the optimization problem is given by

$$\min_d G(d) = \frac{1}{2}\Omega^2(d) + \theta J(d) \tag{25}$$

where

$$J(d) = \min_{w_m,\gamma,b,\xi_i} \frac{1}{2}\left(\sum_{m=1}^{M} d_m \|w_m\|^2 + \lambda\|\gamma\|^2\right) + C\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad y_i f^T(x_i) \geq 1-\xi_i,\ \xi_i \geq 0 \tag{26}$$

Here, $\gamma = [\gamma_1, ..., \gamma_P]'$ and $\lambda, C \geq 0$ are the regularization parameters. If we define $\widetilde{w_m} = \left[w_m', \sqrt{\lambda}\gamma'\right]'$, $\widetilde{v_m} = d_m\widetilde{w_m}$ and $\widetilde{\Phi_m}(x_i) = \left[\Phi_m(x_i)', \frac{1}{\sqrt{\lambda}}f(x_i)'\right]'$, where $f(x_i) = [f_1(x_i), ..., f_P(x_i)]$. Then we can derive the following equation

$$J(d) = \min_{\widetilde{v_m},b,\xi_i} \frac{1}{2}\sum_{m=1}^{M}\frac{\|\widetilde{v_m}\|^2}{d_m} + C\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad y_i\left(\sum_{m=1}^{M}\widetilde{v_m}'\widetilde{\Phi}(x_i) + b\right) \geq 1-\xi_i,\ \xi_i \geq 0 \tag{27}$$

Table 1 Adaptive Multiple Kernel Learning Algorithm

Initialization: $d = 1/M$

for $t = 1, ..., T_{max}$ do

1 Solve the dual variable α_t by the dual of SVM

using LIBSVM with the kernel matrix $\sum_{m=1}^{M} d_m \widetilde{K_m}$.

2 Update the base kernel coefficients d_t by

$d_{t+1} = d_t - \eta_t g_t$.

end for

By introducing the Lagrangian multipliers α, the dual form of the optimazition is:

$$J(d) = \max_{\alpha} \alpha' - \frac{1}{2}(\alpha \cdot y)' \left(\sum_{m=1}^{M} d_m \widetilde{K_m} \right) (\alpha \cdot y) \tag{28}$$

This is the same form as the dual form of primary SVM with kernel matrix $\sum_{m=1}^{M} d_m \widetilde{K_m}$. Then the optimization problem can be solved by an existing SVM solver [6] .

It was proven in [24] that this optimization problem is jointly convex with respect to d, $\widetilde{v_m}$, b and ξ_i. For the multiple kernel learning parameter d could be updated by $d_{t+1} = d_t - \eta_t g_t$, where $g_t = (\nabla_t^2 G)^{-1} \nabla_t G$ according alternative coordinate descent method shown in Table 1.

4 Results

In this section, we evaluate our proposed method by testing head-pose estimation accuracy and transfer learning between two datasets.

4.1 Head pose estimation accuracy

We evaluated the performance of the pose estimation framework on the UcoHead and DPOSE dataset, for which *pan*, *tilt* and *roll* head rotation measurements are available with the datasets. Upon resizing the face crops to 20×20 pixels, and computing the appearance templates (of size 80×20) for each class, we performed three-fold cross validation, with two parts used for training and one for testing. The mean accuracy obtained from the three

Table 2 Head-pose classification accuracy using SVM.

	KL distance feature	Covariance descriptor feature
Ucohead	71.2 %	79.4 %
DPOSE	83.4 %	87.5 %

Table 3 Head-pose classification results of using transfer learning technique. (A - UcoHead, B - DPOSE)

	Source A, Target B Test B	Source B, Target A Test A
Transfer	71.2 %	54.4 %
Not-transfer	43.4 %	36.5 %

runs is presented in Table 2. We can observe that covariance descriptor feature is better than KL-distance feature for these two datasets.

4.2 Transfer learning between two datasets

We evaluate transfer learning benefit between the UcoHead dataset and the DPOSE dataset. We discretized the space of possible head horizontal rotations (*Pan*) into 8 classes as we propose above. For each pan class, we randomly select images from the dataset and repeat the experiments 5 times to calculate mean results. We evaluate our methods by two experiments. One is using UcoHead dataset as source domain (471 images) and DPOSE dataset (60 images). Then we use 610 images from DPOSE dataset, which is not included in the training set, as testing set. The other experiment is using party data (610 images) we record as source domain and UcoHead (80 images) as target domain for training dataset. Then we use 629 images from UcoHead dataset, which is not included in the training set, as testing set. Here the image number for source domain is nearly 8 times larger than target domain in training set which is reasonable for transfer learning. We use 5 base Gaussian kernels ($i.e.,\ K(x_i, x_j) = exp(-\gamma D^2(x_i, x_j))$) with different kernel parameters $\gamma = \{-2, -1, 0, 1, 2\}$ respectively. Table 3 illustrates the classification results for head-pose estimation between two different dataset with transfer learning and without transfer learning.

From Table 3, we could observe that the classification accuracy is relatively low if we do not use transfer learning technique between two datasets. With transfer learning, we could actually extract some useful information from the source dataset to help classify on target dataset which increases the classification accuracy a lot.

4.3 Comparison

At last, we also compare with other low-resolution headpose estimation methods. Table 4 shows the comparison with other state-of-the-art methods.

Table 4 Comparison with other low-resolution headpose estimation methods

Methods	UcoHead,	DPOSE
Ours	71.2 %	83.4 %
[32]	63.4 %	73.5 %
[28]	67.4 %	76.5 %

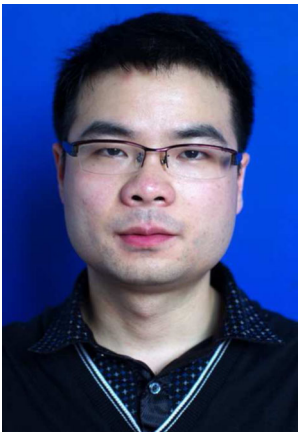
5 Conclusion

Human head pose is the first step in understanding the behaviors of human. We propose a framework to do the head pose estimation in low resolution images. We propose an adaptive multiple kernel transfer learning technique to overcome the weakness of apperance-based feature representation. Experimental results on public dataset prove the efficiency of our proposed method.

References

1. APerez, Cordoba M (2003) A precise eye-gaze detection and tracking system. In: Proc Intl Conf. on Computer Graphics, Visualization and Computer Vision
2. Argyrious A, Evegenious T (2007) Multi-task feature learning. In: NIPS
3. Ba O, Odobez J (2005) Evaluation of multiple cues head pose estimation algorithms in natural environments. In: Proc. of the Intl. Conf. on Multi-media and Expo
4. Ba O, Odobez J (2011) Multiperson visual focus of attention from head pose and meeting contextual cues. In: IEEE Transactions on Pattern Analysis and Machine Intelligence
5. Bonilla E, Chai K, Williams C (2008) Multi-task gaussian process prediction. In: NIPS
6. Chang CC, Lin CJ (2001) Libsvm: a library for support vector machines
7. Chutorian EM, Trivedi MM (2009) Head pose estimation in computer vision: a survey. In: IEEE Transations on Pattern Analysis and Machine Intelligence
8. Dai W, Yang Q, Yu Y (2007) Boosting for transfer learning. In: ICML
9. Daume H (2007) Frustratingly easy domain adaptation. In: Proc. of the Assoc. Computational Linguistics
10. Duan L, Tsang IW, Xu D, Maybank SJ (2009) Domain transfer svm for video concept detection. In: CVPR
11. Duan L, Xu D, Tsang IW, Luo J (2010) Visual event recognition in videos by learning from web data. In: CVPR
12. Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proc. 10th ACM SIGKDD Int Conf. Knowledge Discovery and Data Mining
13. Evgenious T, Poggio T, Verri A (2002) Regularization and statistical learning theory for data analysis. Computational Statistics and Data Analysis 38:421–432
14. Freund Y, Schapire R (1999) A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 14(5):771–780
15. Gee A, Cipolla R (1994) Determining the gaze of faces in images, vol 12
16. Gretton A, Borgwardt K, Scholkopt B (2006) A kernel method for the two-sample-problem. In: NIPS
17. Huang J, Smola A, Scholkopf B (2007) Correcting sample selection bias by unlabeled data. In: NIPS
18. Jiang W, Zavesky E, Chang SF, Loui A (2008) Cross-domain learning methods for high-level visual concept classification. In: Proc. IEEE Int Conf. Image Processing, pp. 161164
19. Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
20. Mihalkova L, Huynh T, Mooney R (2007) Mapping and revising markov logic networks for transfer learning. In: Proc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence
21. Orozco J, Gong S, Xiang T (2009) Head pose classification in crowded scenes. In: British Machine Vision Conference
22. Pan SJ, Yang Q (2010) A survey on transfer learning. In: IEEE Transactions on Knowledge and Data Engineering, Vol 22. NO 10
23. Pan SJ, Kwok JT, Yang Q (2008) Transfer learning via dimensionality reduction. In: Proc. Assoc. for the Advancement of Artificial Intelligence
24. Rakotomamonjy A, Bach FR (2008) Simplemkl. In: Journal of Machine Learning Research
25. Robertson N, Reid I (2006) Estimating gaze direction from low-resolution faces in video. In: Proc. European Conf. Computer Vision
26. Smith K, Ba O, Odobez J (2008) Tracking the visual focus of attention for a varying number of wandering people. In: IEEE Transactions on Pattern Analysis and Machine Intelligence
27. Sugiyama M, Nakajima S, Kawanabe M (2008) Direct importance estimation with model selection and its application to covariate shift adaptation. In: NIPS
28. Tosato D, Farenzena M, Cristani M (2010) Multi-class classification on riemannian manifolds for video surveillance. In: Proc. European Conf. Computer Vision
29. Tuzal O, Porikli F, Meer P (2008) Pedestrian detection via classification on riemannian manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(10):1713–1727

30. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: CVPR
31. Wu P (2004) Improving svm accuracy by training on auxiliary data sources. In: ICML
32. Yan Y, Subramanian R, Lanz O, Sebe N (2012) Active transfer learning for multi-view head-pose classification. In: ICPR
33. Yan Y, Ricci E, Subramanian R, Lanz O, Sebe N (2013) No matter where you are: Flexible graph-guided multi-task learning for multi-view head pose classification under target motion. In: ICCV
34. Yan Y, Ricci E, Subramanian R, Liu G, Sebe N (2014) Multi-task linear discriminant analysis for multi-view action recognition. IEEE Trans Image Process 23(12):5599–5611
35. Yan Y, Ricci E, Liu G, Sebe N (2015) Egocentric daily activity recognition via multitask clustering. IEEE Transactions on Image Processing 24(10):2984–2995
36. Yan Y, Yang Y, Meng D, Liu G, Tong W, Hauptmann A, Sebe N (2015) Event oriented dictionary learning for complex event detection. IEEE Transactions on Image Processing 24(6):1867–1878
37. Yan Y, Ricci E, Subramanian R, Liu G (2016) A multi-task learning framework for head pose estimation under target motion. IEEE Transactions on Pattern Analysis and Machine Intelligence
38. Yang J, Yan R, Hauptmann AG (2007) Cross-domain video concept detection using adaptive svms. In: Proc. ACM Int Conf. Multimedia
39. Yao Y, Dorretto G (2010) Boosting for transfer learning with multiple sources. In: CVPR
40. Zheng VW, Pan SJ, Yang Q (2008) Transferring multi-device localization models using latent multi-task learning. In: Proc. Assoc. for the Advancement of Artificial Intelligence



Ping'an Li is currently a lecturer in Hunan College of Information, China. His research interests include signal and information processing, Pattern recognition and intelligent control.

Yufeng Li is currently an assistant Professor in Hunan College of Information. His research interests include embedded developing, intelligent control.

Lixin Tan is currently a professor in Hunan College of Information, China. His research interests include intelligent control and signal processing