An Adaptation Framework for Head-Pose Classification in Dynamic Multi-view Scenarios

Anoop K. Rajagopal[1], Ramanathan Subramanian[2], Radu L. Vieriu[3],
Elisa Ricci[4], Oswald Lanz[5], Kalpathi Ramakrishnan[1], and Nicu Sebe[2]

[1] Indian Institute of Science, Bangalore, India
{anoopkr,krr}@ee.iisc.ernet.in
[2] Department of Computer Science and Information Engineering (DISI), Trento, Italy
{subramanian,sebe}@disi.unitn.it
[3] "Gheorghe Asachi" Technical University, Iąsi, Romania
rvieriu@etti.tuiasi.ro
[4] Department of Electrical and Information Engineering, University of Perugia, Italy
elisa.ricci@diei.unipg.it
[5] Fondazione Bruno Kessler, Trento, Italy
lanz@fbk.eu

Abstract. Multi-view head-pose estimation in low-resolution, dynamic scenes is difficult due to blurred facial appearance and perspective changes as targets move around freely in the environment. Under these conditions, acquiring sufficient training examples to learn the dynamic relationship between *position*, *face appearance* and *head-pose* can be very expensive. Instead, a **transfer learning** approach is proposed in this work. Upon learning a weighted-distance function from many examples where the target position is *fixed*, we **adapt** these weights to the scenario where target positions are *varying*. The adaptation framework incorporates reliability of the different face regions for pose estimation under positional variation, by transforming the target appearance to a *canonical appearance* corresponding to a *reference* scene location. Experimental results confirm effectiveness of the proposed approach, which outperforms state-of-the-art by 9.5% under relevant conditions. To aid further research on this topic, we also make DPOSE- a dynamic, multi-view head-pose dataset with ground-truth publicly available with this paper.

1 Introduction

The ability to determine a person's head-pose is critical for video surveillance and human-behavior understanding (HBU). Extensive research has been devoted to head-pose estimation for over a decade [1], and recent research has focused on estimating head-pose from surveillance data [2–6], where faces are captured at low resolution. Employing a single camera view is often insufficient for studying people's behavior in large environments and multi-view images have been exploited to achieve robust pose estimation in [3, 7]. However, most of these methods are designed to work in settings where the target's position is *fixed*.

K.M. Lee et al. (Eds.): ACCV 2012, Part II, LNCS 7725, pp. 652–666, 2013.

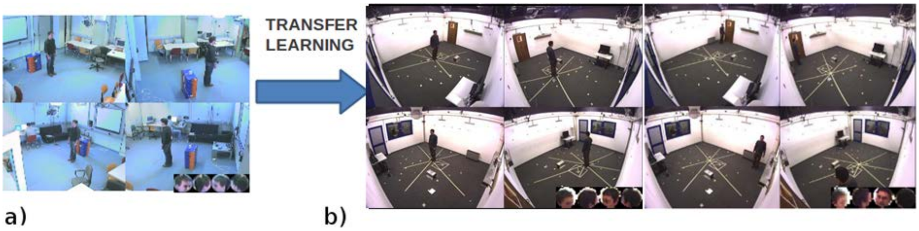


Fig. 1. We deal with scenario (b), where the target is free to move around. For two target positions, the four camera views are shown two-by-two, and facial appearance in these views are shown on the bottom-right. We learn from many labeled examples in (a) where the target's position is fixed, and transfer this knowledge to (b) for enhanced pose classification. (a),(b) are exemplars from the CLEAR [8] and DPOSE datasets.

The objective of this work is to determine the coarse head-orientation[1] from multiple, low-resolution views captured by large field-of-view surveillance cameras, as the target *moves around* freely in the environment. The challenging nature of this scenario can be perceived from Fig.1(b). Even as the target's absolute 3D head orientation is identical for the two instances shown, there are many face appearance differences in the four camera views (bottom-right) due to change in perspective. This phenomenon can severely affect pose classification.

Table 1 presents the effect of appearance variation on pose classification. While the state-of-the-art ARCO algorithm [2] performs very well on CLEAR [8], where target position is ***fixed***, its performance dips sharply on the DPOSE dataset with ***moving*** targets. Learning position-induced appearance variation from examples acquired at multiple locations is a solution. However, acquiring sufficient training data over all scene locations is highly expensive. Therefore, we adopt a **transfer learning** approach.

When the training (*source*) and test (*target*) data have different attributes, knowledge can be transferred from *source* to *target* [9, 10] upon incorporating additional knowledge from a few labeled *target* examples. This obviates the need to compile a large number of *target* samples, and re-synthesize a *target*-specific model. For determining head-pose under positional variation, we first learn a weighted-distance function on the *source* which has many training examples corresponding to a *fixed* position. The weighted-distance function assigns weights to face patches, and the patch weight is indicative of its saliency for pose classification. We then adapt these weights to the *target* data with moving persons. The *source* and *target* data used in our experiments are CLEAR and DPOSE respectively, which differ with respect to (i) scene dimensions, (ii) relative camera configurations (iii) illumination conditions and (iv) target motion (Fig.1).

To compensate for the appearance variation with motion in DPOSE, we warp all target face appearances to a *canonical* appearance, which would be the face

[1] We are mainly interested in determining the absolute 3D head-pan (horizontal rotation) into one of eight classes, each denoting a quantized 45^o (360/8) pan.

Table 1. Head-pose classification accuracy obtained with ARCO algorithm [2] for different training/test datasets. Task is to classify head-pan to one of eight classes.

Train	# samples	Test	# samples	Accuracy
CLEAR	7490	CLEAR	7485	91.8%
CLEAR	7490	DPOSE	11789	46.2%

appearance if the target was positioned at a *reference* location. Only those face patches still visible at the reference location are reliable for classification. To this end, we assign a ***reliability score*** for each patch as we transform the target appearance to the canonical form. The adaptation framework incorporates this patch reliability information while learning a weighted-distance function on the *target* to achieve enhanced classification. In summary, we make the following contributions:
(i) This is one of the first works to explore multi-view head-pose estimation under positional variation, which is an important problem that needs to be addressed for effective surveillance/HBU in natural settings.
(ii) We compiled a large multi-view dataset with pose ground-truth for dynamic targets known as DPOSE, which we make publicly available with this paper.
(iii) This is also the first work to employ transfer learning for multi-view head-pose estimation in dynamic settings. Transfer learning is an attractive alternative to the highly expensive procedure of compiling training samples at multiple scene locations to learn appearance variation with position.

2 Related Work

To put our contributions in perspective, we review past work in the areas of (a) head-pose estimation from surveillance data, (b) multi-view head-pose estimation and (c) transfer learning in this section.

Many works have addressed the problem of head pose estimation from low resolution images [2, 4–6]. In [4], a Kullback-Leibler (KL) distance-based facial appearance descriptor is found to be effective for pose classification on the i-LIDS dataset [11]. In [2], the array-of-covariance (ARCO) descriptors are shown to be very robust to scale/lighting variations as well as occlusions, and produce 11% better classification on i-LIDS as compared to [4]. In [5], an unsupervised, scene-specific gaze estimator is proposed, while [6] proposes a coupled adaptation framework for joint body and head pose estimation. However, all these works perform head-pose estimation in a single camera set-up.

Among multi-view pose estimation works, a particle filter is combined with two neural networks for head pan and tilt classification in [7]. Also a HOG-based confidence measure is used to determine the relevant views for classification. In [12], multi-class SVMs are employed to compute a probability distribution for head-pose in each view, to produce a more precise estimate upon fusion. Nevertheless, both works attempt to determine head-orientation as a person rotates in place, and position-induced appearance variations are not considered.

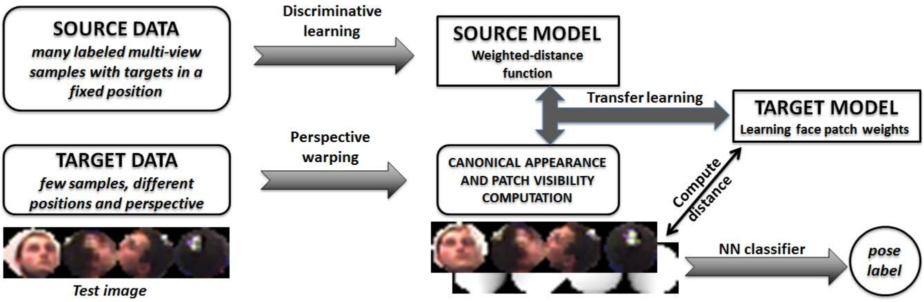

SOURCE DATA
many labeled multi-view samples with targets in a fixed position
Discriminative learning
SOURCE MODEL
Weighted-distance function
Transfer learning
TARGET MODEL
Learning face patch weights
TARGET DATA
few samples, different positions and perspective
Perspective warping
CANONICAL APPEARANCE AND PATCH VISIBILITY COMPUTATION
Compute distance
NN classifier
pose label
Test image

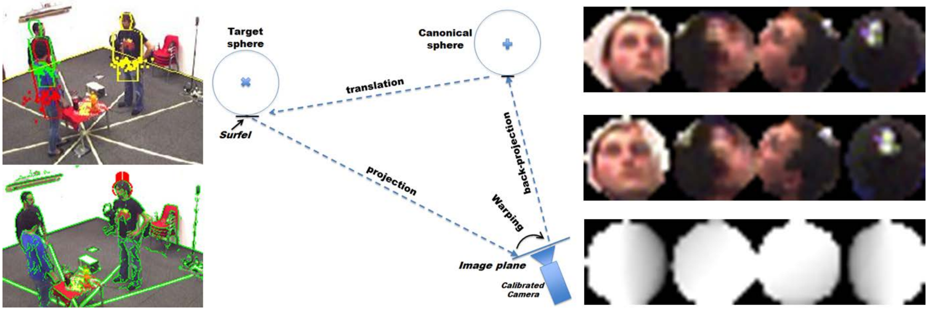
Fig. 2. Overview of the proposed method

One multi-view approach robust to positional variations is discussed in [3], where face texture is mapped on to a spherical head model, and head-pose is determined from the face location on the unfolded spherical texture map. Functionally, [3] is the work closest to ours, but there is a crucial difference between the two. More cameras are needed to produce an accurate texture map ([3] uses eight cameras), while we use only four cameras located at the room corners. As synthesizing a textured 3D model using only four low-resolution views is difficult, we use an image-based approach for classifying head-pose.

Transfer learning approaches have become very popular in computer vision recently [13–15]. However, we are not aware of any these approaches used for multi-view head pose estimation. Our transfer learning framework is inspired by previous works [16, 17], where an effective regularization term for learning relationships between *source* and *target* distributions is proposed. However, our approach is specifically tailored for head-pose classification with positional variation as we integrate information about the patch reliability score into the learning process. Moreover the proposed max-margin learning problem is different from the one considered in [17]. Distance learning methods for head-pose estimation have been previously proposed in [18, 19]. We adopt the method proposed in [19] for learning on the *source*, and extend the same to a transfer learning setting.

3 Head Pose Estimation under Positional Variation

An overview of the proposed method is presented in Fig.2. The proposed system consists of main components: (i) pre-processing-which involves tracking of the moving target(s) and head localization (ii) transfer learning for *target* model creation, where the weights of face patches are learnt to classify pose for the *target* data and (iii) *target* classification. For the *target* data during training/testing, once the four-view target's face appearances are obtained, they are warped to the canonical form, and the visibility of each face patch at a reference position is computed for learning/classification. The detailed description is as follows.



(a) (b) (c)

Fig. 3. (a) Head-localization procedure: color-based particle filter output (top). Projection of spherical head model used for shape-likelihood estimation. Projected circle in red (bottom). (b) Overview of the perspective warping process. (c) Original 4-view face crops (top), warped crops (middle) and patch visibility at reference location (bottom).

3.1 Pre-processing and Perspective Warping

A multi-view, color-based particle filter [20] is used to compute the 3D body-centroid of target(s). Once the target's body centroid and height are estimated by the tracker (Fig.3(a)), we sample a new set of particles around the estimated 3D head-position using a Gaussian with variance $\sigma_x = \sigma_y = 30\text{cm}, \sigma_z = 10\text{cm}$[2]. Assuming a spherical head model, a head-shape likelihood is computed for each particle by projecting a 3D sphere onto each view employing camera calibration information (Fig.3(a)). Finally, the sample with the highest likelihood sum is determined as the head location. This procedure integrates information from multiple views using a unique 3D geometrical head/body-model with occlusion handling and can be used to jointly locate heads of multiple persons.

As the *target* data contains motion, we always warp the target (or original) face appearance to a canonical appearance corresponding to a *reference* position in the environment that best matches with the *source* imaging conditions[3]. This warping allows for compensation of perspective-based appearance variations and enables effective learning even when only few *target* images are available. The perspective warping procedure is outlined in Fig.3(b). To reconstruct the canonical appearance, each pixel corresponding to the canonical appearance is first back-projected onto a sphere, virtually placed at the *reference* position, to obtain the corresponding 3D surface point. This point is then translated to the target sphere (sphere located at target position), and its image projection is mapped to the originating pixel. During this process, visual information may be lost due to self-occlusions, and pixels could be merged or dilated (due to multiple correspondences between canonical and target pixels). To account for these

[2] These values account for the tracker's variance, the horizontal and vertical offsets of the head from the body centroid due to head pan, tilt and roll.

[3] This procedure is also applicable in the case where the number of cameras/views for the *source* and *target* are different.

inconsistencies, we assign a *pixel reliability score*, $r_p \in [0,1]$ to each canonical pixel. The weight is calculated as the ratio (upper-bounded to 1) of areas of the target and canonical surface patch projections.

Fig.3(c) presents an example of the original and warped appearances along with the computed reliability masks. Significant pose difference induced by the target's displacement from the reference position can be observed for the first and last views. Also, large changes between the original and canonical views are noticeable around the periphery, while central regions are more similar. This is because, when the displacement between the target and canonical positions is large, reliable correspondences can only be computed in the canonical image for target pixels around the center, while multiple peripheral target pixels tend to correspond to the same canonical pixel. Therefore, canonical pixels that arise from peripheral regions in the target image are assigned lower r_p's (occluded pixels indeed have $r_p = 0$), while r_p's for central pixels are closer to 1. Again, as the r_p's will vary depending on the target position, we divide the space into distinct regions and compute the expected r_p for each region from the *target* training set to learn the region-wise *target* patch weights. Finally, the 4-view original/canonical appearances for the *source/target* are resized to 20 × 80 resolution (each view is 20 × 20) prior to feature extraction for transfer learning. From the 4-view appearance image, features are computed for overlapping 8×8 patches (with step size of 4). Next, we describe the transfer learning procedure adopted for learning *target* patch weights.

3.2 Learning a Distance Function under Positional Variation

When the *source* and *target* data have different attributes, so that a model trained on *source* will not usually work well on the *target*, the adaptation framework transfers knowledge learnt from (*source*) to the *target*. For our problem scenario, the *source* (CLEAR) has many exemplars for subjects standing at a ***fixed*** position, while the *target* (DPOSE) has subjects imaged as they are ***moving***. Formally, from the large *source* set $\mathcal{T}_s = \{(\mathbf{x}_1, l_1), (\mathbf{x}_2, l_2), \ldots, (\mathbf{x}_{N_s}, l_{N_s})\}$, we seek to transfer knowledge to the *target* incorporating additional information from a small number of *target* samples $\mathcal{T}_t = \{(\mathtt{x}_1, \mathtt{l}_1), (\mathtt{x}_2, \mathtt{l}_2), \ldots, (\mathtt{x}_{N_t}, \mathtt{l}_{N_t})\}$. Here, $\mathbf{x}_i/\mathtt{x}_i$ and $l_i/\mathtt{l}_i$ respectively denote *source/target* image features and associated class labels.

Overview. The proposed transfer learning framework is a two-step process. First, a discriminative distance function is learned on the *source*. Given that each image consists of Q patches, we learn a weighted-distance on the *source*, $D_{\boldsymbol{W}_s}(\mathbf{x}_i, \mathbf{x}_j)$ as a parameterized linear function, *i.e.* $D_{\boldsymbol{W}_s}(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{W}_s^T \mathbf{d}_{ij}$, where $\mathbf{d}_{ij}$ is the distance (we use euclidean distance) between corresponding patches in images. $\boldsymbol{W}_s$ is the source patch weight vector, which encodes the saliency of each face patch for pose classification.

We propose to learn $D_{\boldsymbol{W}_s}(\mathbf{x}_i, \mathbf{x}_j)$ by imposing that a pair of images $\mathbf{x}_i$ and $\mathbf{x}_j$ corresponding to the same pose should be more similar than two images

$\mathbf{x}_i$ and $\mathbf{x}_k$ corresponding to different poses. Formally, the following quadratic programming problem is considered [19]:

$$\begin{aligned} &\min_{\boldsymbol{W}_s, \xi_i \geq 0} \frac{\lambda}{2} ||\boldsymbol{W}_s||^2 + \frac{1}{N_s} \sum_{i=1}^{N_s} \xi_i \\ &\text{s.t.} \qquad \min_{l_i \neq l_k} \boldsymbol{W}_s^T \mathbf{d}_{ik} - \max_{l_i = l_j} \boldsymbol{W}_s^T \mathbf{d}_{ij} \geq 1 - \xi_i \end{aligned} \tag{1}$$

The constraints $\boldsymbol{W}_s \geq 0$ are introduced to impose that the learned distance function is always positive. To solve this optimization problem, we adopt an efficient iterative algorithm based on stochastic gradient descent (Algorithm 2).

Learning Distance Function on the *Target*. In the second step, a distance function $D_{\boldsymbol{W}_t}(\cdot)$ is learned on target data $\mathcal{T}_t$. $\boldsymbol{W}_s$ is used in this phase, in order to transfer the *source* knowledge onto the *target*. The ***reliability score*** for each *target* patch as computed from the *canonical* transformation (Fig.3(c)) is also considered.

We first discuss the adaptation of the *source* weights to the *target*, assuming that all *target* images correspond to a *specific* position (for simplicity, we can assume the *reference* position associated to the canonical image here). We formulate the adaptation problem as:

$$\begin{aligned} &\min_{\boldsymbol{W}_t \geq 0,\ \xi_i \geq 0,\ \boldsymbol{\Sigma} \succeq 0} \quad \lambda_1 ||\boldsymbol{W}_t||^2 + \lambda_2 \text{tr}(\boldsymbol{W^T} \boldsymbol{\Sigma}^{-1} \boldsymbol{W}) + \frac{1}{N_t} \sum_{i=1}^{N_t} \xi_i \\ &\text{s.t.} \qquad \min_{\mathtt{l}_i \neq \mathtt{l}_k} \boldsymbol{W}_t^T \mathbf{d}_{ik} - \max_{\mathtt{l}_i \neq \mathtt{l}_j} \boldsymbol{W}_t^T \mathbf{d}_{ij} \geq 1 - \xi_i \\ &\qquad\qquad \text{tr}(\boldsymbol{\Sigma}) = 1 \end{aligned} \tag{2}$$

where tr$(\cdot)$ denotes trace of matrix, $\boldsymbol{W} = [\boldsymbol{W}_s \quad \boldsymbol{W}_t]^T$ and $\boldsymbol{\Sigma} \in I\!R^{2 \times 2}$ is a symmetric adaptation matrix defining the dependencies between the *source* and the *target* weight vectors. The transfer learning is realized by the term tr$(\boldsymbol{W^T} \boldsymbol{\Sigma}^{-1} \boldsymbol{W})$, and specifically by learning the *source-target* dependency matrix $\boldsymbol{\Sigma}$. This adaptation term, previously proposed in [16], allows for both negative and positive transfer, and, being a convex function on the optimization parameters, makes our approach convex. Defining $\boldsymbol{\Sigma} = [\alpha \ \ \beta; \beta \ \ 1-\alpha]$[4], (2) can be rewritten as follows:

$$\begin{aligned} &\min_{\boldsymbol{W}_t, \alpha, \beta} \gamma_1(\alpha, \beta) ||\boldsymbol{W}_t||^2 - \gamma_2(\alpha, \beta) \boldsymbol{W}_s^T \boldsymbol{W}_t - \gamma_3(\alpha, \beta) ||\boldsymbol{W}_s||^2 + \frac{1}{N_t} \sum_{i=1}^{N_t} \xi_i \\ &\text{s.t.} \qquad \min_{\mathtt{l}_i \neq \mathtt{l}_k} \boldsymbol{W}_t^T \mathbf{d}_{ik} - \max_{\mathtt{l}_i = \mathtt{l}_j} \boldsymbol{W}_t^T \mathbf{d}_{ij} \geq 1 - \xi_i, \\ &\qquad\quad \boldsymbol{W}_t \geq 0, \quad \xi_i \geq 0, \quad \alpha(1-\alpha) - \beta^2 > 0 \end{aligned} \tag{3}$$

where we define $\mathit{\Delta}(\alpha, \beta) = \alpha(1-\alpha) - \beta^2$,

$$\gamma_1(\alpha, \beta) = \lambda_1 + \frac{\lambda_2 \alpha}{\mathit{\Delta}(\alpha, \beta)}, \quad \gamma_2(\alpha, \beta) = \frac{2\lambda_2 \beta}{\mathit{\Delta}(\alpha, \beta)}, \quad \gamma_3(\alpha, \beta) = \frac{\lambda_2 (1-\alpha)}{\mathit{\Delta}(\alpha, \beta)} \tag{4}$$

[4] $\mathit{\Sigma}$ is chosen in this form in order to be positive semi-definite and have a trace equal to 1 as proposed in [15].

Finally, we integrate information regarding appearance variation in the multiple views due to position changes. As previously stated, when the target appearance is transformed to the canonical form, the reliability of a face patch for pose classification depends on the target position. We assume that the room is divided into R distinctive regions, and to effectively learn appearance variation with position, we have K_r *target* training samples for each region $r \in R$. The patch reliability score vector, $\hat{\rho} = [\rho_q], q = 1..Q$, is determined from the mean reliability score of the P patch pixels, *i.e.* $\rho_q = \frac{1}{P}\sum_{p=1}^{P} r_p$ and the ***expected patch reliability*** for region $r, r = 1..R$, is computed as $\hat{\rho}_r = \frac{1}{K_r}\sum_{i=1}^{K_r} \hat{\rho}_i$. Given $\hat{\rho}_r$, a diagonal matrix $\mathbf{B} \in I\!R^{Q \times Q}$ for region r is defined such that $B_{pq} = e^{-(1-\frac{\eta}{\hat{\rho}_r})}$ if $p = q$, while $B_{pq} = 0$ otherwise. Then the optimization problem (3) can be reformulated accounting for patch reliability as follows:

$$\min_{\boldsymbol{W}_t,\xi_i,\alpha,\beta} \gamma_1(\alpha,\beta)||\mathbf{B}\boldsymbol{W}_t||^2 - \gamma_2(\alpha,\beta)\boldsymbol{W}_s^T\boldsymbol{W}_t - \gamma_3(\alpha,\beta)||\boldsymbol{W}_s||^2 + \frac{1}{N_t}\sum_{i=1}^{N_t}\xi_i \quad (5)$$

$$\text{s.t.} \quad \min_{\mathtt{l}_i \neq \mathtt{l}_k} \boldsymbol{W}_t^T\mathbf{d}_{ik} - \max_{\mathtt{l}_i = \mathtt{l}_j} \boldsymbol{W}_t^T\mathbf{d}_{ij} \geq 1 - \xi_i$$

$$\boldsymbol{W}_t \geq 0, \quad \xi_i \geq 0, \quad \alpha(1-\alpha) - \beta^2 > 0$$

Solving the Transfer Learning Optimization Problem. To solve (5), we consider the auxiliary vector, $\hat{\boldsymbol{W}}_t = \mathbf{B}\boldsymbol{W}_t$ and re-define accordingly $\hat{\boldsymbol{W}}_s = \mathbf{B}^{-1}\boldsymbol{W}_s$ and $\hat{\mathbf{d}}_{ik}^{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{d}_{ik}$. We adopt an efficient alternate optimization approach. In particular, we first solve with respect to $\hat{\boldsymbol{W}}_t$ with α, β fixed, and then, given a certain distance function we compute the optimal adaptation weights α, β. The optimization problems to be solved are:

$$\min_{\hat{\boldsymbol{W}}_t,\xi_i \geq 0} \gamma_1(\alpha,\beta)||\hat{\boldsymbol{W}}_t||^2 - \gamma_2(\alpha,\beta)\hat{\boldsymbol{W}}_s^T\hat{\boldsymbol{W}}_t + \frac{1}{N_t}\sum_{i=1}^{N_t}\xi_i \quad (6)$$

$$\text{s.t.} \quad \min_{\mathtt{l}_i = \mathtt{l}_k} \hat{\boldsymbol{W}}_t^T\hat{\mathbf{d}}_{ik}^{\mathbf{B}} - \max_{\mathtt{l}_i \neq \mathtt{l}_j} \hat{\boldsymbol{W}}_t^T\hat{\mathbf{d}}_{ij}^{\mathbf{B}} \geq 1 - \xi_i$$

and:

$$\min_{\boldsymbol{\theta}} \mathbf{a}^T\boldsymbol{\theta} \quad (7)$$

$$\text{s.t.} \quad \boldsymbol{\theta}^T\mathbf{I}\boldsymbol{\theta} - \mathbf{e}^T\boldsymbol{\theta} \leq 0$$

where $\boldsymbol{\theta} = [\alpha \quad \beta]^T$, $\mathbf{e} = [1 \quad 0]^T$, $\mathbf{a} = [\hat{\boldsymbol{W}}_t^T\hat{\boldsymbol{W}}_t - \hat{\boldsymbol{W}}_s^T\hat{\boldsymbol{W}}_s \quad -2\hat{\boldsymbol{W}}_s^T\hat{\boldsymbol{W}}_t]$.

To solve (6), we adopt an efficient online learning approach. The objective function of the quadratic program (6) is a sum of two terms: a strongly convex function, *i.e.* the square norm of the weights, and a convex function which is represented by the sum of the differences of the similarity scores and the contribution of *source* weights. For solving this, we again employ Algorithm 2. Problem in (7) can be reduced to a Second Order Cone Programming (SOCP) problem and it is solved efficiently using SEDUMI[5]. The overall alternate optimization approach terminates upon convergence and the learned *target* weights are $\boldsymbol{W}_t = \mathbf{B}^{-1}\hat{\boldsymbol{W}}_t$. The entire process is outlined in Algorithm 1.

[5] `http://sedumi.ie.lehigh.edu/`

4 Experimental Results

This section presents a brief description of the CLEAR [8] and DPOSE datasets, followed by a detailed discussion of the experiments and results.

Algorithm 1. Algorithm for Learning a Transfer Distance Function

```
Input: The source and target training data $\mathcal{T}_s$, $\mathcal{T}_t$.
Learning on Source Data
  Set $\lambda_1$ to a fixed value ($\lambda_1 = 1$ in our experiments).
  $\boldsymbol{W}_s$=ComputeDistance($\mathcal{T}_s$, $\lambda_1$, 0, $\bar{\mathbf{0}}$, $\mathbf{I}$);

Learning on Target Data
  Compute patch reliability matrix $\mathbf{B}$.
  Set $\lambda_1$ and $\lambda_2$ to fixed values ($\lambda_1 = 100, \lambda_2 = 10$ in our experiments).
  Set $\hat{\boldsymbol{W}}_s = \mathbf{B}^{-1}\boldsymbol{W}_s$.
  repeat until convergence
    Compute $\gamma_1(\alpha, \beta)$, $\gamma_2(\alpha, \beta)$ with (4).
    $\hat{\boldsymbol{W}}_t$=ComputeDistance($\mathcal{T}_t$, $\gamma_1$, $\gamma_2$, $\hat{\boldsymbol{W}}_s$, $\mathbf{B}$);
    Given $\hat{\boldsymbol{W}}_s$, $\hat{\boldsymbol{W}}_t$ compute $\alpha, \beta$ solving (7).
  end
  Compute $\boldsymbol{W}_t = \mathbf{B}^{-1}\hat{\boldsymbol{W}}_t$.
Output: $\boldsymbol{W}_t$
```

Datasets: The CLEAR dataset [8], is a popular multi-view dataset used for evaluating multi-view pose estimation algorithms. Acquired from 15 targets rotating in-place in the middle of a room, the dataset comprises over 30000 synchronously acquired images from four cameras with head-pose measurements. Another publicly multi-view dataset, with moving targets, is provided by [3]. However, this dataset only contains ground-truth measurements for a mannequin's head mounted on a tripod, as against human subjects. So, for the purpose of studying the head-pose estimation problem with moving targets, we compiled the DPOSE (dynamic, multi-view head-pose) dataset.

The DPOSE dataset contains sequences acquired from 16 targets, where the target is either (i) rotating in-place at the room center, or (ii) moving around freely in a room, and moving their head in all possible directions. The dataset consists of over 50000 images. Head pan, tilt and roll measurements for various poses are recorded using an accelerometer, gyro, magnetometer platform (visible in Fig.3(c)) strapped onto the head using an elastic band running down from the back of the head to the chin. As mentioned earlier, the CLEAR and DPOSE datasets differ with respect to (i) scene dimensions, (ii) relative camera configurations (iii) illumination conditions and (iv) moving targets.

To demonstrate the validity of the proposed adaptation framework, the performance of our algorithm is evaluated on the DPOSE data when (i) the target rotates in-place and (ii) when the target freely moves around. We will discuss the experimental results for these scenarios as follows:

Algorithm 2. Online algorithm to solve (1) and (6)

```
$\boldsymbol{w}$=**ComputeDistance**($\mathcal{T}$, $\theta_1$, $\theta_2$, $\boldsymbol{w}_o$, $\mathbf{M}$)
{
    Set the number of iteration $T$ and the sample size $k$.
    $\boldsymbol{w} = 0$.
    **for** $t = 1, \dots, T$ **do**
      Choose $\mathcal{T}_k \subseteq \mathcal{T}$ s.t. $|\mathcal{T}| = k$
      Set $\mathcal{T}^+ = \{(\mathbf{x}_i, l_i) \in \mathcal{T}_k : \max_{l_i \neq l_k, l_i = l_j} \left[1 - \boldsymbol{w}^T \hat{\mathbf{d}}_{ij}^{\mathbf{M}} + \boldsymbol{w}^T \hat{\mathbf{d}}_{ik}^{\mathbf{M}}\right] \geq 0\}$
      $\forall$ $(\mathbf{x}_i, l_i) \in \mathcal{T}^+$ compute constraints violators
        $\{(\hat{\mathbf{x}}_j, l_j), (\hat{\mathbf{x}}_k, l_k) \in \mathcal{T} : \hat{\mathbf{x}}_k, \hat{\mathbf{x}}_j := \arg\max_{l_i \neq l_k, l_i = l_j,} \left[1 - \boldsymbol{w}^T \hat{\mathbf{d}}_{ij}^{\mathbf{M}} + \boldsymbol{w}^T \hat{\mathbf{d}}_{ik}^{\mathbf{M}}\right]\}$
      $\boldsymbol{w}^{t+\frac{1}{3}} = \left(1 - \frac{1}{t}\right) \boldsymbol{w}^t + \frac{1}{k\theta_1 t} \sum_{(\mathbf{x}_i, l_i) \in \mathcal{T}^+} \left[\hat{\mathbf{d}}^{\mathbf{M}}(\mathbf{x}_i, \hat{\mathbf{x}}_k) - \hat{\mathbf{d}}^{\mathbf{M}}(\mathbf{x}_i, \hat{\mathbf{x}}_j)\right] - \frac{\theta_2}{\theta_1 t} \boldsymbol{w}_o$
      $\boldsymbol{w}^{t+\frac{2}{3}} = \max\{0, \boldsymbol{w}^{t+\frac{1}{3}}\}$
      $\boldsymbol{w}^{t+1} = \min\{1, \frac{1}{\sqrt{\theta_1} \|\boldsymbol{w}^{t+\frac{2}{3}}\|}\} \boldsymbol{w}^{t+\frac{2}{3}}$
    **endfor**
}
```

Transfer Learning for *Stationary* Target- We compare our results with those obtained with two recent state-of-the-art methods for head-pose classification from low-resolution images: array of covariance (ARCO) descriptors [2], and the multi-camera head-pose estimation framework proposed in [12]. ARCO is a powerful framework for pose classification from low-resolution images which employs covariance descriptors, and has been shown to be robust to scale/lighting variations and occlusions. Also, since ARCO is inherently not a transfer learning approach, we adapted boosting-based transfer learning [10], which adapts to the *target* upon re-creation of the *source+target* model, from many *source* and few *target* samples. We term this algorithm as ARCO-Xboost.

In this first series of experiments, we assume that the target is rotating in-place in the *target* dataset, and we consider the cases where the learning model are trained with CLEAR data and tested on DPOSE images. Upon dividing the (original/canonical) four-view appearance image into a number of overlapping patches (we use 8 × 8 patches in our experiments and a patch step-size of 4 pixels), we computed the patch descriptors employing the following features:

1) 7D covariance descriptors $\phi = R, G, B, Gabor_{\{0,\pi/6,\pi/3,4\pi/3\}}$, comprising pixel colors and Gabor coefficients upon filtering at the specified orientations (termed $Cov(d = 7)$).
2) 12D covariance descriptors $\phi = [x, y, R, G, B, I_x, I_y, Gabor_{\{0,\pi/6,\pi/3,4\pi/3\}}, KL]$, which additionally include the pixel positions (x, y), image gradients (I_x, I_y) and the KL divergence descriptor proposed in [4] (termed Cov $(d = 12)$).
3) HOG [21] and LBP [22] descriptors, which are also popular for low-resolution head-pose estimation. 64 bin histograms are used as LBP/HOG descriptors in our experiments.

Table 2 presents the pose classification accuracies with the different features for the stationery target scenario. 10 *target* samples/class are used in addition to 300 *source* samples/class for model creation. Here, ARCO$_s$ denotes an ARCO model trained only with *source* data, ARCO-Xboost$_{(s+t)}$ denotes boosting-based adaptation with ARCO, NWD$_s$ denotes a single nearest neighbor (NN) classifier where the test image is assigned the label of the nearest *source* sample based on (unweighted) Euclidean distance, and WD$_t$ denotes our adaptation framework, where patch weights are learnt for the *target*, and based on the learnt weighted-distance measure, the test sample is assigned the class of the nearest *target* sample. ARCO$_s$ and NWD$_s$ denote baselines, as they correspond to the classification performance when a model trained purely on the *source* is tested on the *target*. To verify if multi-view pose classification performance is better than using only one view, we also computed the accuracies using the 4-view appearance image and the individual images for the 4-views. The average accuracy of 4-views, measured independently, is reported in the single-view case. Each result reported is the mean of 4 independent trials where each trial uses a randomly selected *target* training set. From the table, we make the following observations:

- Per feature, the 4-view accuracies are much higher than the mean single view accuracies, which confirms that pose classification utilization from multiple views is more powerful and robust.
- Also, employing 1-view/4-views, higher accuracies are obtained with 12D *Cov* as compared to 7D *Cov*. This implies that as more and more image statistics are employed for computing covariance features, the pose classification performance improves.
- Accuracies with the weighted distance measure are much higher for LBP, as compared to HOG. Therefore, LBP appears a more suitable feature for pose-classification from surveillance data, as compared to HOG.
- Even as similar accuracies are obtained ARCO-Xboost and WD, there exists one important difference between the two approaches. ARCO-Xboost requires model synthesis from both (large number of) *source* and (few) *target* samples- when *target* attributes change, re-training a model can be highly expensive. In contrast, our approach requires the *source* model to be trained ***exactly once*** and adapts to the *target* through online learning, which is much faster.

Transfer Learning for *Moving* Target- Now, we analyze pose classification results for the case where the target is *moving*. In this scenario, it is highly expensive to acquire many labeled target samples at multiple locations for learning the appearance variation for the same pose with position. To tackle this problem, we divide the space (room) into a finite number of non-overlapping regions and learn the target patch weights for each region as detailed in Section 3.2. In this work, we divide the room into four quadrants, denoted as R1-R4 in anti-cyclic order, and assume that 5 *target* training samples are available per pose-class per region. This amounts to a total of $5 \times 4 \times 8 = 160$ *target* training samples. Again, experiments are performed upon learning from a *source* training set with 300 samples/class.

Mean reliability masks computed from 40 *target* training samples acquired for each of the quadrants R1-R4 are presented in Fig.4. These masks demonstrate why we opt for region-based *target* patch weight learning. Notice that the masks for diagonally opposite regions R1, R3 and R2,R4 are antisymmetric, *i.e.* darker regions in the R1 mask are brighter for the R3 mask and vice-versa. This is again due to the perspective problem- as the target position varies, the face patches visible in the canonical view also vary, and patch visibility modulates its saliency for pose-classification.

Table 3 presents the region-wise classification accuracies obtained with the various methods and features for the moving target scenario. An identical procedure as for the stationary target case was employed to compute the results. For brevity, we only compare the performance of transfer learning approaches (WD, ARCO-Xboost), with the baselines indicated in braces. From Table 3, we again make the following observations:

- For the same feature, WD performs better than ARCO for the moving target case. In fact, the difference in performance is higher for weaker features ($Cov(d = 7)$), as compared to stronger features ($Cov(d = 12)$). Therefore, our transfer learning framework appears to be more effective when the representational features are less robust.
- For ARCO, the best performance is achieved with ($Cov(d = 12)$), while LBP produces the best results with WD. Comparing these best results, WD produces a 9.5% increase (70.93 vs 64.75) in classification performance as compared to ARCO.
- The accuracies obtained with the multi-view SVM [12] are much lower. Even though this method uses multi-view fusion to compute pose, the low accuracies may be attributed to (a) weak features (only gradient features are used in this work) and (b) non-consideration of the dynamic target position scenario.
- It needs to be noted that our approach explicitly considers the variation appearance due to target dynamics. which is not the case with ARCO-Xboost. However, if target motion only affected the appearance of a few face patches, ARCO should still be effective as it learns a classifier *per patch.* The fact that this is not the case reinforces our claim that head-pose computation in dynamic settings is a non-trivial and important research problem.

Qualitative Results. We also show some qualitative results obtained with our approach in Fig.5. Fig.5(a,b) correspond to a single moving target, while Fig.5(c) shows computed pose labels for 2 of 6 moving targets (as in a real party, which is our application scenario). Fig.5(a) corresponds to a correct result, while Fig.5(b) shows an incorrect result, because the face localization and ensuing face crops (on the top-right) are erroneous. Fig.5(c) demonstrates that the proposed approach can work well even with multiple targets. However, no pose ground-truth was available for this sequence, but the computed pose labels can be observed to be correct from visual inspection (videos provided in supplementary material).

Table 2. Performance comparison assuming ***stationary*** target. {# Train (*source*)=2400 (300 samples/class), (*target*)= 80 (10 samples/class)}, {# test (*target*) = 12406.}

	Cov	*Cov*	HOG	LBP	*Cov*	*Cov*	*HOG*	*LBP*
	$(d = 7)$	$(d = 12)$			$(d = 7)$	$(d = 12)$		
	(1-view)	(1-view)	(1-view)	(1-view)	(4-views)	(4-views)	(4-views)	(4-views)
$\text{ARCO}_{(s)}$	21.2	35	-	-	31.3	58.5	-	-
NWD_s	19	20.6	22.2	39.5	47.7	61.3	32.3	75.1
$\text{ARCO-Xboost}_{(s+t)}$	41.8	62.2	-	-	68.8	85.4	-	-
WD_t	57.7	59.9	32	63.3	78.3	83.3	52.1	**85.6**

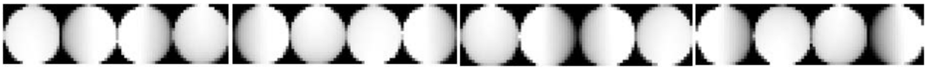
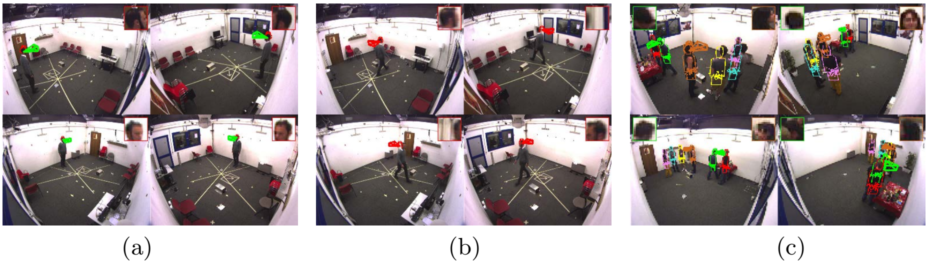


Fig. 4. The mean reliability masks computed from 40 *target* training samples for R1-R4, which are respectively the room quadrants traced in anti-cyclic order beginning from top-left.

Table 3. Performance comparison for the *moving* target scenario. The room is divided into 4 quadrants (regions R1-R4). {# Train (*source*)=2400 (300 samples/class), (*target*)= 160 (5 samples/class/region)}, {# Test (*target*) = 2399 (R1), 3185 (R2), 3048 (R3), 2996 (R4).} Baseline (ARCO, NWD) accuracies are within braces. Only 4-view accuracies are reported.

	ARCO-Xboost	ARCO-Xboost	WD	WD	WD	Multi-view
	Cov $(d = 7)$	*Cov* $(d = 12)$	*Cov* $(d = 7)$	*Cov* $(d = 12)$	*LBP*	SVM
R1	41.1 (27.2)	66.1 (45.4)	65.8 (33.1))	69.8 (45)	**74.7** (60.9)	47.6
R2	43.6 (28.3)	67.6 (45.5)	67.4 (41.5)	72.4 (51.6)	**77.6** (61.3)	51.3
R3	45.9 (29.3)	66.2 (44.4)	59.6 (51.2)	63 (59.6)	**66.9** (58.7)	41
R4	41.7 (28.1)	59.1 (41)	60.6 (37.8)	62.4 (42.3)	**64.5** (58.3)	41.6



(a) (b) (c)

Fig. 5. Head pose estimation results with target moving (a,b). Green cone indicates accurate pan estimation while the red cone denotes wrongly predicted pose label. (c) Results with the proposed approach for a *party* scenario involving multiple targets.

5 Conclusion and Future Work

We introduce a transfer learning framework for multi-view head-pose classification when the target is moving. Experimental results confirm effectiveness of the proposed approach. We also make the multi-view image dataset with pose ground-truth publicly available for further research on this topic. In our experiments we consider a four camera set-up but we want to stress that our method is general and can be applied also to different scenarios (*e.g.* different number of cameras/views in *source* and *target*). Future work involves development of a single transfer function for the environment, instead of the current region-based learning method, integrating information from multiple sources.

References

1. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE PAMI 31, 607–626 (2009)
2. Tosato, D., Farenzena, M., Cristani, M., Spera, M., Murino, V.: Multi-class Classification on Riemannian Manifolds for Video Surveillance. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 378–391. Springer, Heidelberg (2010)
3. Zabulis, X., Sarmis, T., Argyros, A.A.: 3d head pose estimation from multiple distant views. In: BMVC (2009)
4. Orozco, J., Gong, S., Xiang, T.: Head pose classification in crowded scenes. In: BMVC, pp. 1–11 (2009)
5. Benfold, B., Reid, I.: Unsupervised learning of a scene-specific coarse gaze estimator. In: ICCV (2012)
6. Chen, C., Odobez, J.M.: We are not contortionists: coupled adaptive learning for head and body orientation estimation in surveillance video. In: CVPR (2012)
7. Voit, M., Stiefelhagen, R.: A system for probabilistic joint 3d head tracking and pose estimation in low-resolution, multi-view environments. In: ICVS, pp. 415–424 (2009)
8. Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.): RT 2007 and CLEAR 2007. LNCS, vol. 4625. Springer, Heidelberg (2008)
9. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22, 1345–1359 (2010)
10. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: ICML, pp. 193–200 (2007)
11. HOSDB: Imagery library for intelligent detection systems (i-lids). IEEE Crime and Security (2006)
12. Muñoz-Salinas, R., Yeguas-Bolivar, E., Saffiotti, A., Carnicer, R.M.: Multi-camera head pose estimation. Mach. Vis. Appl. 23, 479–490 (2012)
13. Yang, W., Wang, Y., Mori, G.: Efficient Human Action Detection Using a Transferable Distance Function. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009, Part II. LNCS, vol. 5995, pp. 417–426. Springer, Heidelberg (2010)
14. Lim, J.J., Salakhutdinov, R., Torralba, A.: Transfer learning by borrowing examples for multiclass object detection. In: NIPS, pp. 118–126 (2011)
15. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR (2011)

16. Zhang, Y., Yeung, D.Y.: A convex formulation for learning task relationships in multi-task learning. In: UAI, pp. 733–742 (2010)
17. Zhang, Y., Yeung, D.Y.: Transfer metric learning by learning task relationships. In: KDD (2010)
18. Wang, X., Huang, X., Gao, J., Yang, R.: Illumination and Person-Insensitive Head Pose Estimation Using Distance Metric Learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 624–637. Springer, Heidelberg (2008)
19. Ricci, E., Odobez, J.M.: Learning large margin likelihoods for realtime head pose tracking. In: ICIP (2009)
20. Lanz, O.: Approximate bayesian multibody tracking. IEEE PAMI (2006)
21. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
22. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: ICCV, pp. 32–39 (2009)