Int J Comput Vis (2014) 109:146–167
DOI 10.1007/s11263-013-0692-2

Exploring Transfer Learning Approaches for Head Pose Classification from Multi-view Surveillance Images

Anoop Kolar Rajagopal · Ramanathan Subramanian · Elisa Ricci · Radu L. Vieriu · Oswald Lanz · Ramakrishnan Kalpathi R. · Nicu Sebe

Received: 15 March 2013 / Accepted: 6 December 2013 / Published online: 24 December 2013

Abstract Head pose classification from surveillance images acquired with distant, large field-of-view cameras is difficult as faces are captured at low-resolution and have a blurred appearance. Domain adaptation approaches are useful for transferring knowledge from the training (*source*) to the test (*target*) data when they have different attributes, minimizing *target* data labeling efforts in the process. This paper examines the use of transfer learning for efficient multi-view head pose classification with minimal *target* training data under three challenging situations: (i) where the range of head poses in the *source* and *target* images is different, (ii) where *source* images capture a stationary person while *target* images capture a moving person whose facial appearance varies under motion due to changing perspective, scale and (iii) a combination of (i) and (ii). On the whole, the presented methods represent novel transfer learning solutions employed in the context of multi-view head pose classification. We demonstrate that the proposed solutions considerably outperform the state-of-the-art through extensive experimental validation. Finally, the DPOSE dataset compiled for benchmarking head pose classification performance with moving persons, and to aid behavioral understanding applications is presented in this work.

Keywords Transfer learning · Multi-view head pose classification · Varying acquisition conditions · Moving persons

Electronic supplementary material The online version of this article (doi:10.1007/s11263-013-0692-2) contains supplementary material, which is available to authorized users.

A. Kolar Rajagopal · R. Kalpathi R.
Department of Electrical Engineering, Indian Institute of Science, Bangalore, India
e-mail: anoopkr@ee.iisc.ernet.in

R. Kalpathi R.
e-mail: krr@ee.iisc.ernet.in

R. Subramanian (✉)
Advanced Digital Sciences Center (ADSC), University of Illinois at Urbana-Champaign, Singapore, Singapore
e-mail: Subramanian.R@adsc.com.sg

E. Ricci · O. Lanz
Fondazione Bruno Kessler, Trento, Italy
e-mail: eliricci@fbk.eu; elisa.ricci@diei.unipg.it

O. Lanz
e-mail: lanz@fbk.eu

E. Ricci
Department of Electrical and Information Engineering, University of Perugia, Perugia, Italy

R. L. Vieriu · N. Sebe
Department of Computer Science and Information Engineering (DISI), Trento, Italy
e-mail: vieriu@disi.unitn.it

N. Sebe
e-mail: sebe@disi.unitn.it

1 Introduction

Over the years, extensive research has been devoted to the study of people's head pose due to its relevance in security, human–computer interaction, advertising as well as cognitive, neuro and behavioral psychology. Head pose dynamics have been found to be useful for determining the attentiveness of drivers (Doshi and Trivedi 2012), addressee identification in human–robot interaction (Katzenmaier et al. 2004) and analyzing social behavior in structured and unstructured interactive settings (Subramanian et al. 2010, 2013; Lepri et al. 2012). Even as humans can effortlessly deduce others' head pose from near and far views, most auto-

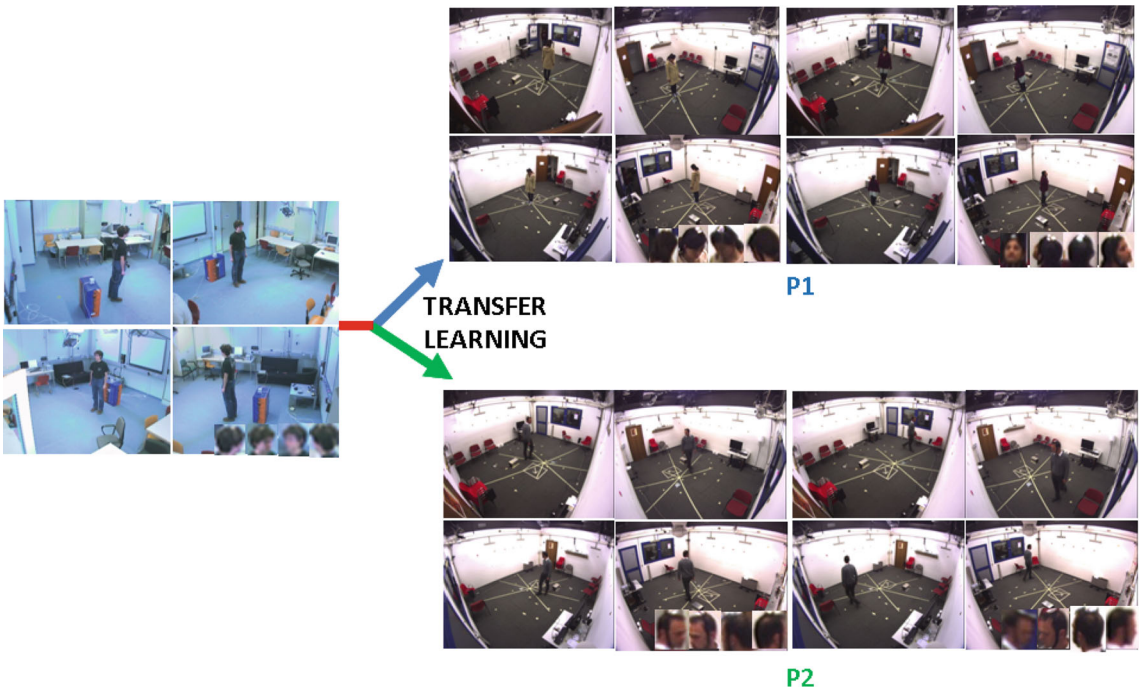

Fig. 1 Problem overview: (*Left*) Exemplar 4-view CLEAR image corresponding to the *source* setting involving a stationary target with frontal head tilt shown two-by-two. The facial appearance in four camera views are shown on the bottom right inset. Our objective is to apply knowledge learnt from many *source* examples onto the *target* setting P3 which combines P1 and P2, where P1 represents the case where targets are stationary, but exhibit a larger head tilt range as compared to the *source*, while P2 involves the same set of head poses as in the *source*, but with moving targets. Figure is best viewed under zoom

mated approaches require detailed facial shape and textural information for reliable head pose estimation (see Murphy-Chutorian and Trivedi 2009 for a review).

Recently though, there has been active interest in determining the head pose from surveillance data (Smith et al. 2008; Tosato et al. 2010; Zabulis et al. 2009; Orozco et al. 2009; Benfold and Reid 2011; Chen and Odobez 2012) where faces are captured by distant, large field-of-view cameras. Under these conditions, estimating head pose is difficult as faces are typically captured at low resolution and appear blurred. Nevertheless, a majority of these techniques are designed for single-camera systems monitoring a relatively small region in space (e.g., train station passageways). Also, employing a single camera view is often insufficient for studying people's behavior in large environments and a handful of approaches (Muñoz-Salinas et al. 2012; Zabulis et al. 2009; Voit and Stiefelhagen 2009) have exploited multi-view images to achieve robust pose estimation. Yet, most of these estimate head pose of a person rotating *in place*.

The larger goal of this work is to estimate people's 3D head orientation as they *freely move* around in naturalistic settings such as parties, museums and supermarkets. Labeling sufficient training data for head pose estimation in such settings is inherently difficult, mainly due to the motion of targets (persons) and the large possible range of head orientations. In contrast, acquiring considerable head pose training data from meeting or group conversational scenarios is much easier due to the involvement of stationary targets and a limited range of head orientations (predominantly frontal head tilt[1]). Therefore, we model head pose estimation in naturalistic settings as a *transfer learning* problem: to learn the relationship between head pose and facial appearance from many labeled examples corresponding to the conversational scenario (*source* data), and employ domain adaptation techniques to transfer this knowledge to the naturalistic setting (*target* data), utilizing only a few *target*-specific training examples. Here, we also assume that the *source* and *target* data are acquired under different conditions, so that models trained on existing and richly annotated datasets can be directly exploited for transfer learning.

Figure 1 illustrates why transfer learning is an effective solution for head pose estimation in the *target* scenario. We use the CLEAR dataset (Stiefelhagen et al. 2007) where targets rotate in place as the *source*, and the DPOSE dataset (Rajagopal et al. 2012), compiled to study head pose estimation under target motion, as the *target*—these datasets evidently differ with respect to (a) scene dimensions, (b) rel-

[1] Head pose estimation involves determination of the *pan* (out-of-plane horizontal head rotation), *tilt* (out-of-plane vertical rotation) and *roll* (in-plane head rotation). In this work, we are mainly concerned about estimating pan and tilt.

Table 1 Impact of head tilt variations and target motion on ARCO head-pan classification accuracy (expressed as %)

Test	CLEAR	DPOSE *frontal*	DPOSE *up*	DPOSE *down*	DPOSE *all*	DPOSE *motion*
Train						
Clear	91.9	57.2	62.7	34.2	52.1	44

To study P1, we used *target* images with exclusively *frontal*, *upward* and *downward* head tilt, and *all* of these tilts. Task is to assign head-pan to one of eight classes

ative camera positions and (c) illumination conditions. To simulate the meeting scenario, we only learn from CLEAR images corresponding to a frontal head tilt as seen on the left. For simplicity, we divide our original problem (P3) into two sub-problems P1 and P2, which are illustrated on the right- P1 represents the condition where the DPOSE targets are stationed at a particular scene location as in CLEAR, but exhibit a larger head tilt range as in a museum or a supermarket. P2 denotes the case where targets exhibit the same range of head poses as in the *source*, but are *freely moving*. Considering P2 in particular, target facial appearances for an identical head pose at two different scene locations are shown. Significant differences in the target's facial appearance for the four camera views can be seen due to perspective and scale changes- as the target moves closer/away from a camera, the face appears larger/smaller as for the first two views, while face regions can become occluded/visible due to target motion as evident from the third and fourth camera views. Therefore, directly learning pose-appearance relationship on the *target* data will require training examples acquired at many scene locations, which is prohibitively expensive.

To study the impact of facial appearance changes due to varying head tilt (as exemplified by P1) and target motion (denoted by P2) on head pose classification, we performed the following experiments. We trained a state-of-the-art head pose classifier based on array of covariance (ARCO) descriptors (Tosato et al. 2010) with the 4-view *source* images (as in Fig. 1), and tested the classifier with (a) *source* images and (b) *target* images corresponding to conditions P1 and P2. The task was to classify the 3D head pan into one of eight classes, each denoting a quantized 45° pan. Table 1 presents the results. Even though ARCO descriptors are robust to scale and lighting variations, pose classification performance dips sharply when the ARCO classifier is tested with the *target* data instead of *source*. For example, even when the *target* faces correspond to a frontal tilt as in the *source*, varying image acquisition conditions limit the *target* classification accuracy to about 57 %. The accuracy reduces further as the *target* facial appearance becomes more dissimilar with respect to the *source*, as with the downward head tilt, where cameras see more of the target's head instead of the face. A further accuracy difference of 13.2 % between the *frontal* and *motion* cases demonstrates the impact of motion in the *target* dataset.

In this paper, we propose a number of transfer learning solutions to overcome the adverse impact of changing attributes between the *source* and *target* data on head pose classification performance. Transfer learning can be broadly categorized into *instance-based transfer* and *parameter/feature-based transfer*. Instance-based transfer learning involves training a classifier with many *source* and a few *target* instances, under the assumption that the *source* data is still useful in the *target* scenario. In a nutshell, learning is performed assigning different weights to training samples in the *source* domain reflecting their relevance in the *target* domain. On the other hand, parameter/feature-based transfer involves modeling of parameters/features common to both *source* and *target* data, so that *source–target* similarities can be exploited for *target* learning.

To address P1, we propose a domain adaptive version of the ARCO pose classifier based on the instance-based transfer learning technique described in Dai et al. (2007). However, this adaptation is still not effective for determining the head pose of moving targets. For solving P2, we therefore propose a novel parameter transfer learning approach where a set of face patch weights are learnt from *source* data, with each patch weight indicating saliency of the face patch for pose classification. These weights are then adapted to the *target* scenario, incorporating a patch *reliability score* measuring the face patch's appearance distortion under target motion. Note that for problems P1 and P2, we are interested in determining only the head pan, resulting in an equal number of *source* and *target* classes. In P3, we show how transfer learning is applicable in the case where the number of *source* and *target* classes are unequal, by utilizing knowledge learnt from *source* data to determine both head pan and tilt under motion in the *target* dataset. To this end, we employ an adaptation of the transferable distance learning framework proposed in Yang et al. (2010). Overall, the afore described methods represent novel transfer learning solutions in the context of multi-view head pose classification, and considerably outperform competing methods as confirmed by experimental results.

To summarize, the main contributions of this paper are:

- We address head pose estimation from surveillance images acquired with multiple and distant large field-of-view cameras by casting it as a transfer learning problem. To our knowledge, we are the first to adopt domain adaptation to tackle this challenging task.
- Motivated by the interest to study people's behavior in naturalistic settings, we consider a multi-camera framework, as single-camera systems are often insufficient for monitoring large spaces, and monocular head pose estimation approaches do not achieve sufficiently robust esti-

mates. Furthermore, in contrast to most previous works, we deal with the challenge of estimating head pose for freely moving targets. Target motion necessitates development of novel solutions which can effectively cope with change in facial appearance due to varying perspective and scale, which we achieve by efficiently exploiting camera geometry information.
– An extensive experimental evaluation is conducted on the novel DPOSE dataset, which is explicitly compiled for benchmarking head pose classification with moving targets.

The paper is organized as follows. Section 2 reviews related work. Section 3 describes the CLEAR and DPOSE head pose databases which are used as the *source* and *target* datasets in this work, and details the pre-processing steps involved prior to transfer learning. Discussion and evaluation of the proposed transfer learning solutions for problems P1, P2 and P3 are presented in Sects. 4, 5 and 6 respectively. We then conclude in Sect. 7.

2 Related Works

To highlight our research contributions, we now review related work on (a) head pose estimation from surveillance data, (b) multi-view head pose estimation and (c) use of transfer learning for computer vision applications.

2.1 Head-Pose Estimation from Surveillance Data

Many works have addressed the problem of head pose estimation from low resolution images (Smith et al. 2008; Tosato et al. 2010; Orozco et al. 2009; Benfold and Reid 2011; Chen and Odobez 2012). Given a large field-of-view camera capturing a number of moving subjects, Gaussian Mixture and Hidden Markov models incorporating location and head pose information are used to determine the number of persons who attend to an outdoor advertisement in Smith et al. (2008). To determine the coarse head pose of moving persons in crowded scenes as in the i-LIDS (HOSDB 2006) underground scene dataset, a novel Kullback–Leibler (KL) distance-based facial appearance descriptor is proposed in Orozco et al. (2009). However, the classification performance achieved in this work is exceeded through the use of array-of-covariance (ARCO) descriptors robust to scale/lighting variations as well as occlusions in Tosato et al. (2010).

Recent approaches have attempted unsupervised or weakly supervised approaches to pose classification exploiting constraints related to head and body motion. In Benfold and Reid (2011), an unsupervised scene-specific gaze estimator is proposed by feeding the output of a head tracker to a conditional random field (CRF), which models the relationship between head motion, walking direction and appearance and simultaneously trains decision tree classifiers. Alternatively, head pose is determined in Chen and Odobez (2012) employing motion-based cues and constraints imposed by joint modeling of head and body pose. Nevertheless, a primary limitation of the aforementioned works is that they determine head pose in a single camera set-up.

2.2 Multi-view Head Pose Estimation

Among multi-view pose estimation works, a particle filter is combined with two neural networks for head pan and tilt classification in Voit and Stiefelhagen (2009). Also, a HOG-based confidence measure is used to determine the relevant views for classification. In Muñoz-Salinas et al. (2012), multi-class SVMs are employed to compute a probability distribution for head pose in each view, and the view-specific distributions are fused to produce a more precise pose estimate. However, both these works attempt to determine head-orientation of a person who rotates in place, while our objective involves computing the head pose of a moving target. Recently, multi-view head pose classification has been attempted employing active transfer learning in Yan et al. (2012) and multi-task learning in Yan et al. (2013). A robust, multi-view head pose estimation approach that can handle moving targets is discussed in Zabulis et al. (2009). Here, facial texture is mapped on to a spherical head model, and head pose is determined from the face location on the unfolded spherical texture map.

While Zabulis et al. (2009) also attempts to solve aforementioned problem P2, our approach differs from theirs in some respects. Firstly, a large number of cameras are required to synthesize an accurate texture map, (a total of 9 cameras are employed in their work), while our solution can work with much fewer cameras. Also, synthesizing a textured 3D model is computationally expensive. In contrast, our solution is predominantly image-based, requiring minimal use of the 3D camera geometry.

2.3 Transfer Learning

An elaborate categorization and review of various transfer learning solutions proposed in literature is presented in Pan and Yang (2010). When the *source* (training) and *target* (test) data are drawn from different distributions, machine learning methods do not work well requiring statistical models to be trained again with labeled *target* data. In many real-world applications, it is highly expensive to collect *target* training data and rebuild *target*-specific models. In such cases, transfer learning or domain adaptation between the *source* and *target* data/tasks is highly desirable.

There are several approaches to transfer learning. *Instance-based* transfer learning (Dai et al. 2007; Jiang and Zhai

2007) involves the reuse of *source* data in a related *target* domain assuming that certain parts of the *source* data are still useful in the *target* scenario. A transfer learning framework modeled on Adaboost is proposed in Dai et al. (2007), which leverages on extensive labeled *source* data in addition to a few labeled *target* data to train an accurate *target* classifier. In Jiang and Zhai (2007) a method to remove potentially harmful training samples from *source* data is proposed, upon determining the relevance of *source* samples by taking into account the difference between conditional probabilities computed on the *source* and on the *target* data.

Feature-based transfer involves finding a 'good' feature representation for the *source* and *target*. Labeled *source* and *target* data features are copied to synthesize an augmented feature space in Daume (2007), on which supervised learning is employed while jointly optimizing *source* and *target* feature weights to maximize prediction accuracy. Alternatively, *parameter-based* transfer (Williams et al. 2007) involves discovery of shared parameters or priors between the *source* and *target* models which can benefit from transfer learning.

2.4 Transfer Learning in Computer Vision

Transfer learning approaches have become very popular in computer vision recently. A transfer learning approach to overcome limited training data for certain classes in object detection is presented in Lim et al. (2011). To this end, a model learns from training examples of other object classes, and transforms those examples to make them more similar to *target* instances. Another visual domain approach to tackle the varying distribution of object features across image datasets (e.g., high resolution DSLR vs webcam images) is discussed in Kulis et al. (2011). Given labeled *source* and *target* examples, an asymmetric, non-linear transformation is applied to map examples from one domain to another- this transformation can be applied independent of the dimensionality of the *source* and *target* domains. Analogously, an adaptive multiple kernel learning method is proposed in Duan et al. (2012) to recognize visual events in consumer videos upon learning from labeled web (e.g., Youtube) videos.

Transfer learning solutions have been extensively employed for activity recognition. Activity recognition across views through the transfer of splits (arrangement of discriminative hyperplanes) from the *source* to the *target* view is described in Farhadi and Tabrizi (2008). Another methodology for cross-view action recognition employing a transferable and sparse dictionary pair learnt for the *source* and *target* views is described in Zheng et al. (2012). Finally, a transferable distance function is learned for action detection with sparse training data in Yang et al. 2010. Learning salient image patches indicative of human actions from video frames in training sequences, the saliency of each patch in the test video frame is computed following which, a weighted distance is measured between the training and test videos to recover actions similar to the training video.

Examining related works applying transfer learning in computer vision, it is evident that no transfer learning solutions have been proposed to address head pose estimation, and in particular, multi-view head pose estimation for moving targets. We adopt the framework proposed in Dai et al. (2007) to solve problem P1 introduced in Sect. 1. For solving P2, we adopt the method proposed in Ricci and Odobez (2009) for learning face patch weights indicative of their saliency on the *source* dataset, and adapt these weights to the *target* through an online learning procedure. This novel transfer learning approach is inspired by previous works such as Zhang and Yeung (2010), where an effective regularization term for learning the *source–target* relationships is proposed. To solve P3, which involves estimation of both pan and tilt in the *target* upon learning from *source* examples only corresponding to a frontal tilt, we devise a weighted-distance approach employing *hyperfeatures* adapted from Yang et al. (2010). We describe the *source* and *target* datasets used in this work, as well as the steps involved in face cropping and facial feature extraction in the following section.

3 Datasets and Pre-processing Steps

Now, we present details regarding the datasets used followed by a brief discussion of how faces of targets are localized and cropped. Facial appearance for the *source* data is consistent across targets, as they are imaged while stationed at the same spatial location; however, facial appearance varies with target position in the *target* dataset, as evident from Fig. 1. Since we adapt a classifier learnt on the *source* to work on the *target* data, we transform all *target* appearances to a canonical appearance in order to determine which face patches can reliably be used for pose classification—a process termed *perspective warping*. We also describe the perspective warping procedure in this section.

3.1 Datasets

We use the popular and extensively annotated CLEAR dataset (Stiefelhagen et al. 2007) as the *source*. The CLEAR database comprises over 27000 synchronously recorded 4-view images of a person standing at the center of a lecture room (Fig. 1, left panel). The four cameras are placed in the room's upper corners, and the person rotating in-place wears a flock-of-birds magnetic motion sensor through which his/her head movements are measured. Head pose measurements for 15 subjects are available as part of the CLEAR database.

Head rotation measurements are also provided by the UcoHead (Muñoz-Salinas et al. 2012) and Greece (Zabu-

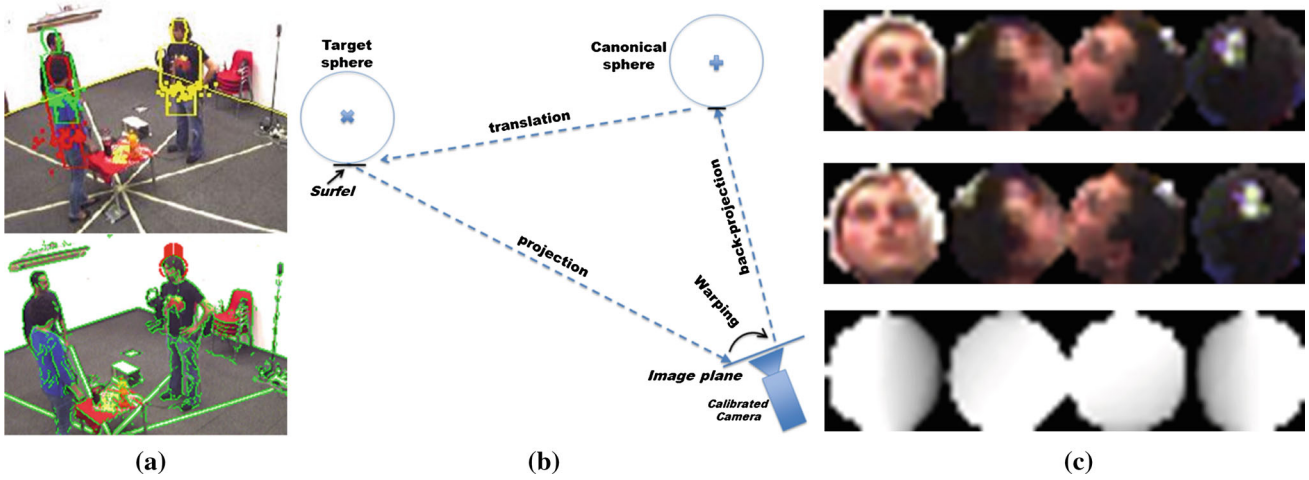

Fig. 2 (**a**) Head-localization procedure: color-based particle filter output (*top*). Projection of spherical head model used for shape-likelihood estimation shown in red (*bottom*). (**b**) Overview of the perspective warping process. (**c**) Original 4-view face crops (*top*), warped crops (*middle*) and patch visibility at reference location (*bottom*)

lis et al. 2009) datasets. Ucohead contains 6-view images capturing head rotations of 10 persons and associated pose measurements—however, here again the subjects rotate in-place and the dataset is much smaller than CLEAR. The Greece dataset contains 9-view images (8 wall-mounted cameras and one ceiling camera) of moving persons. Nevertheless, ground truth head pose readings are available only for one sequence involving a mannequin head mounted on a tripod. Therefore, these datasets were not used in our study.

In order to objectively evaluate head pose classification performance for moving targets, we compiled the dynamic headpose or DPOSE[2] database (Rajagopal et al. 2012), which is used as the *target* dataset in this work. The DPOSE dataset consists of over 50000 4-view synchronized images capturing static as well as moving targets,[3] with ground-truth head pose measurements acquired using an accelerometer, gyrometer, magnetometer platform. Target head movements are captured using cameras mounted at the corners of a 6×4.8 m^2 room. It is important to note here that, apart from target motion, the *source* and *target* datasets also differ with respect to (a) scene dimensions and distance of cameras from the subject, (b) relative camera positions and (c) illumination conditions.

Since we are more interested in pose classification rather than precise head pose estimation, we segmented the CLEAR and DPOSE data into 24 classes, with eight demarcations denoting a quantized 45° (360/8) head-pan and three demarcations for the head-tilt, namely, *frontal* ($[-20°, 20°]$), *upward* ($[20°, 90°]$) and *downward* ($[-90°, -20°]$). Since our goal is to learn the head pose-facial appearance relationship from group conversation-like scenarios, we used only CLEAR images corresponding to the frontal tilt for training in our experiments.

3.2 Face Cropping and Perspective Warping

Since we rely on the facial appearance of static/moving targets to classify the head pose, the pre-processing steps prior to facial feature extraction include person tracking, face localization and cropping. Also, when the target moves, we transform the face appearance to a canonical form through a perspective warping procedure. These steps are detailed below.

A multi-view, particle filter-based framework to track targets' 3D body centroid positions using a shape-cum-color model (Lanz 2006; Lanz and Brunelli 2008) is used for face localization. Given the body centroid and height of the target as estimated by the tracker (Fig. 2a, top), we sample a new set of particles around the estimated 3D head-position using a Gaussian with variance $\sigma_x = \sigma_y = 30$ cm, $\sigma_z = 10$ cm.[4] Assuming a spherical model of the head, a head-shape likelihood is computed for each particle by projecting a 3D sphere onto each view employing camera calibration information (Fig. 2a, bottom). Finally, the sample with the highest likelihood sum is determined as the head location and the circular face crop is generated as in Fig. 2c. 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 main difference between *source* and *target* datasets is that the *target* data involves moving persons, we always

[2] available at http://tev.fbk.eu/DATABASES/DPOSE.html

[3] 27824 4-view images correspond to static targets rotating in-place at the room center, while 25660 images capture freely moving targets.

[4] 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.

transform a moving target's face appearance to a canonical 4-view appearance corresponding to a *reference* position in the scene that best matches with the *source* imaging conditions.[5] This warping allows for scale and perspective-related changes in facial appearance to be geometrically compensated for, when the camera calibration is known. Learning pose-appearance relations from the *target* data can then be more effective, under our assumption that only a few labeled *target* samples are available.

The perspective warping procedure is outlined in Fig. 2b. Assuming a spherical head model, 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 sphere is then translated to the target position, and the image projections of the translated surface points are computed to determine the canonical-to-target pixel correspondences for warping. During this process, visual information may be lost owing to self-occlusions resulting from sphere translation, or pixels could be merged or dilated (due to multiple correspondences between canonical and target pixels). To account for these inconsistencies, we assign a *pixel reliability score*, $r_p \in [0, 1]$ to each canonical pixel upon warping. The weight is calculated as the ratio (upper-bounded to 1) of the area of target and canonical surface patch (or surfel) projections in the target appearance image.

Figure 2c presents an example of the original and warped facial appearances in the four views along with the computed reliability masks. Significant relative pose difference induced by the target's displacement from the reference position can be observed in 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 have $r_p = 0$), while r_p's for central pixels are closer to 1.

Under target motion, we rely on the reliability masks to determine those facial regions (or patches) that are useful for pose classification. As these masks will vary depending on the target position, we divide the space into distinct regions and compute the typical/expected reliability mask for each region from the *target* training set. In all the following experiments, original (for stationary targets) or canonical (under target motion) appearances from the four views are resized to 20×20 pixel resolution and concatenated to synthesize the 4-view facial appearance image as in Fig. 2c. Thereafter, appearance features are computed for overlapping 8×8 patches (with a step size of 4). The following sections describe the proposed transfer learning solutions for solving aforementioned problems P1, P2 and P3.

[5] This warping can also be applied in the case where the number of cameras/views for the *source* and *target* are different.

4 Head-Pan Classification Under Varying Head-Tilt

Now, we focus on problem P1 illustrated in Fig. 1, where the objective is to predict head-pan in the *target* upon learning from many *source* and a few *target* examples. Apart from varying image acquisition conditions, facial appearance in CLEAR and DPOSE differs due to the range of head poses exhibited by subjects—while all CLEAR training examples correspond to a frontal head-tilt, DPOSE head-tilts are in the range $[-90°, 90°]$. However, we assume that in both CLEAR and DPOSE, targets rotate in-place at a fixed scene location (room center). Therefore, facial appearance for a given head pose remains consistent across targets with respect to perspective and scale.

To begin with, we tested if the array-of-covariance (ARCO) classifier (Tosato et al. 2010) trained on *source* can effectively predict head-pan for the *target* images. ARCO uses powerful covariance features, robust to occlusions as well as scale and lighting variations, for head-pose classification from low-resolution images. Upon dividing the image into a number of overlapping patches, ARCO computes covariance-based patch descriptors. Subsequently, a multi-class Logitboost classifier is learnt for each patch, and the test sample is assigned a label based on majority vote of the patch-based classifiers. Nevertheless, as shown in Table 1, ARCO is still ineffective for predicting head pose when the *source* and *target* data attributes vary considerably, as with the CLEAR and DPOSE datasets.

An effective method for transferring knowledge across datasets through *induction* of a few *target* examples in the learning process is proposed in `Tradaboost` (Dai et al. 2007). `Tradaboost` is modeled on AdaBoost where, given a training set comprising *source* and *target* samples, a set of weak learners are learnt such that misclassified target samples are given priority at each step. In this way, the resulting model is *tuned* to effectively predict *target* samples. Analogously, the ARCO framework also employs a multi-class Logitboost classifier $\{F_l\}$ for each image patch, comprising $l = 1 \ldots L$ weak classifiers. Given a training set $\{x_i\}$ with N samples corresponding to class labels $1 \ldots J$, the Logitboost algorithm iteratively learns training samples most difficult to classify through a set of weights w_i and posterior probabilities, $P_j(x_i)$. Each weak learner solves a weighted-regression problem, whose goodness of fit is measured by the response value vector for the i^{th} training sample, $z_i = \{z_{ij}\}_{j=1}^{J}$.

Algorithm 1 ARCO-Xboost- Transfer learning with ARCO Logitboost

Input: Combined *source* ($x_i, y_i \in \mathscr{T}_s$), *target* ($\mathrm{x}_i, \mathrm{y}_i \in \mathscr{T}_t$) training set (for each facial appearance image patch)
$\mathscr{T} = \{(x_1, y_1), \ldots (x_N, y_N), (\mathrm{x}_{N+1}, \mathrm{y}_{N+1}), \ldots, (\mathrm{x}_{N+M}, \mathrm{y}_{N+M})\}$, where $\{y_i\}, \{\mathrm{y}_i\} = 1..J$, number of learners L.
For $i = 1..N+M$, initialize weights $w_i = \frac{1}{N+M}$ and posterior probabilities $P_j(x_i) = \frac{1}{J}$.
Set $\alpha_s = \frac{1}{2} ln(1 + \sqrt{2ln\frac{N}{L}})$
for $l = 1 \ldots L$
Initialize learner $F_l = 0$.
Compute response values z_i and weights w_i from $P_j(x_i)$
if $L > 1$
Normalize the weight vector $w_1, \ldots, w_{N+M}$
Compute the error on *target*, $\varepsilon_t = \sum_{k=N+1}^{N+M} \frac{w_k[y_k \neq h(\mathrm{x}_k)]}{\sum_{i=1}^{N+M} w_i}$,
where $h(.)$ is the classified label.
Set $\alpha_t = \frac{1}{2} ln(\frac{1-\varepsilon_t}{\varepsilon_t})$, $\varepsilon_t < \frac{1}{2}$
Update weights
$w_i \leftarrow w_i e^{-\alpha_s(y_j \neq h(\mathrm{x}_j))}$ (modify misclassified *source* weights)
$w_i \leftarrow w_i e^{\alpha_t(y_j \neq h(\mathrm{x}_j))}$ (modify misclassified *target* weights)
end if
Compute learner F_l using least-square regression from computed z_{ij}'s and w_i's.
Compute new $P_j(x_i)$'s and $h(x_i)$'s.
end for
Output: Set of learners $\{F_l\}$

Following Dai et al. (2007), we designed ARCO-Xboost- a transfer learning approach for the ARCO Logitboost classifier as follows. Given $N + M$ training data comprising N *source* and M *target* samples, with $N \gg M$, the error on *target* (ε_t) is computed at every step upon normalizing the w_i's. Also, α_s and α_t, which are respectively the attenuating and boosting factors for misclassified *source* and *target* samples, are determined. Finally, weights of misclassified *target* data are boosted by a factor of e^{α_t}, so that the model incorporates more *target*-specific information, while the weights for misclassified *source* weights are attenuated by a factor of $e^{-\alpha_s}$ to discourage learning of these samples. ARCO-Xboost is summarized in Algorithm 1.

4.1 Experimental Results and Discussion

For all our experiments, the *source* training set comprised 300 CLEAR images for each of the eight frontal tilt classes. Also, all classification accuracies reported in this paper correspond to the mean value obtained from four independent trials involving randomly chosen *target* training sets. For the sake of evaluating how ARCO-Xboost improves classification performance over ARCO, we used covariance features derived from the 12-dimensional feature set $\phi = [x, y, R, G, B, I_x, I_y, OG, Gabor_{\{0,\pi/6,\pi/3,4\pi/3\}}, KL]$. Here, x, y and R, G, B denote spatial positions and color values, while Ix, Iy and OG respectively denote intensity gradients and gradient orientation of pixels. $Gabor$ is the set of coefficients obtained from Gabor filtering at aforementioned orientations (frequency = 16 Hz), while KL denotes maximal divergence between corresponding patches in the target face image and each of the pose-class templates computed as described in Orozco et al. (2009). The presented results correspond to two covariance features, namely, $Cov(d = 12)$, which denotes covariance descriptors computed from all features in ϕ, and $Cov(d = 7)$, where covariances are computed only for color and Gabor features.

As such, the ARCO Logitboost classifier learns until all training data are correctly classified and therefore, ARCO classification accuracies are significantly improved by simply including a few *target* examples in the training process. For example, when the *target* images correspond to *downward* tilt, the head pan classification accuracy improves from 34.2[6] to 61 % when 5 *target* samples/class are added to the *source* data prior to model training. However, preferentially learning misclassified *target* samples over *source* examples as in ARCO-Xboost provides a benefit when (a) weaker features are employed for learning (b) fewer learners are used and (c) very few *target* examples are inducted for transfer learning.

Figure 3a–d shows variation in classification accuracies for ARCO and ARCO-Xboost upon changing the number of learners L employed in the boosting framework. Plots are shown for the model trained with $Cov(d = 12)$ (blue) and $Cov(d = 7)$ (red) features extracted from the 4-view face appearances, with 5 *target* samples/class added to the *source*. Higher classification accuracies are achieved with $Cov(d = 12)$ features, implying that they are superior appearance descriptors for pose classification. However, larger gains in pose classification performance observed with $Cov(d = 7)$ features suggest that preferential learning of *target* examples is more beneficial with weaker features.

Maximum gains achieved with ARCO-Xboost using $Cov(d = 7)$ and $Cov(d = 12)$ features are 8.1 % (40.9 vs 37.8, $L = 12$) and 2.8 % (70.6 vs 68.7, $L = 8$) for the DPOSE *down* and *all* test sets respectively. We observe from Table 1 that these test sets are most dissimilar to the *source*, resulting in worse classification performance with a *source*-only model. Thus, the ARCO-Xboost framework achieves most effective transfer learning for more dissimilar *source* and *target* data. Also, classification performance with ARCO-Xboost more or less saturates for $L \geq 12$.

Classification accuracy trends upon increasing the number of inducted *target* samples from 5 to 30 samples/class, with $L = 12$, are shown in Fig. 4a–d. Here, we also compare the classification performance achieved with 4-view and single-view features—mean value of the accuracies achieved with each of the four views is considered for the single-

[6] as seen from Table 1, which presents accuracies achieved with *source*-only $Cov(d = 12)$ features

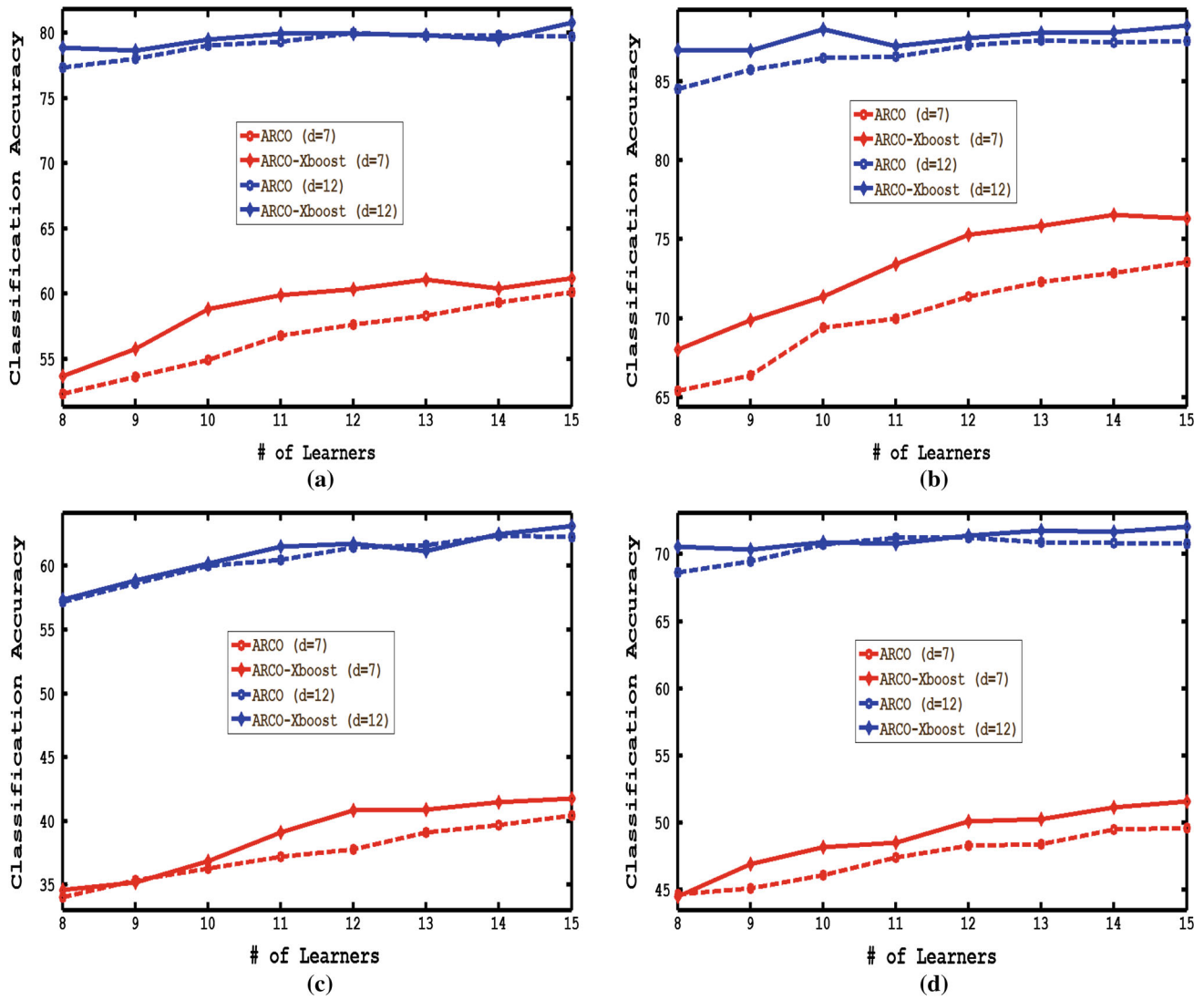

Fig. 3 Classification accuracies achieved with ARCO and ARCO-Xboost upon varying number of weak learners L with 5 *target* samples/class added to the *source* dataset. 4-view $Cov(d = 12)$ and $Cov(d = 7)$ features are used. Plots (**a**–**d**) correspond to DPOSE images with *frontal* (#test = 12406), *upward* (#test = 5141), *downward* (#test = 6277) and *all* tilts (#test = 23824)

view case. Considerably higher accuracies are obtained when features extracted from all four views are employed for pose classification, implying that multi-view information improves robustness of pose classification on low-resolution images. Both ARCO and ARCO-Xboost classification accuracies increase sharply as more *target* examples are added to the *source* training data. While little difference is observed between ARCO and ARCO-Xboost classification performance employing $Cov(d = 12)$ features, ARCO-Xboost performs better than ARCO for both single and 4-views with $Cov(d = 7)$ features. Also, larger gains with ARCO-Xboost are obtained with single-view features and when fewer *target* examples are inducted in the training set.

Furthermore, in order to demonstrate the benefit of transfer learning employing extensively labeled *source* data as against training a classifier only using few *target* data, we compared classification accuracies obtained with ARCO-Xboost (trained with *source*+*target*) against ARCO trained with only *target* data using $Cov(d = 12)$ features. The benefit of transfer learning is evident from Table 2, when either single or 4-view features are employed for classification. ARCO-Xboost consistently produces higher accuracies, and the performance improvements are more pronounced for smaller *target* training data sizes.

Another set of experiments were conducted to compare ARCO-Xboost with other state-of-the-art transfer learning methods. More specifically, we consider the Feature Replication (FR) method proposed in Daume (2007), the Adaptive Support Vector Machine (A-SVM) approach presented in Yang et al. (2007), the Domain Adaptation Machine (DAM)

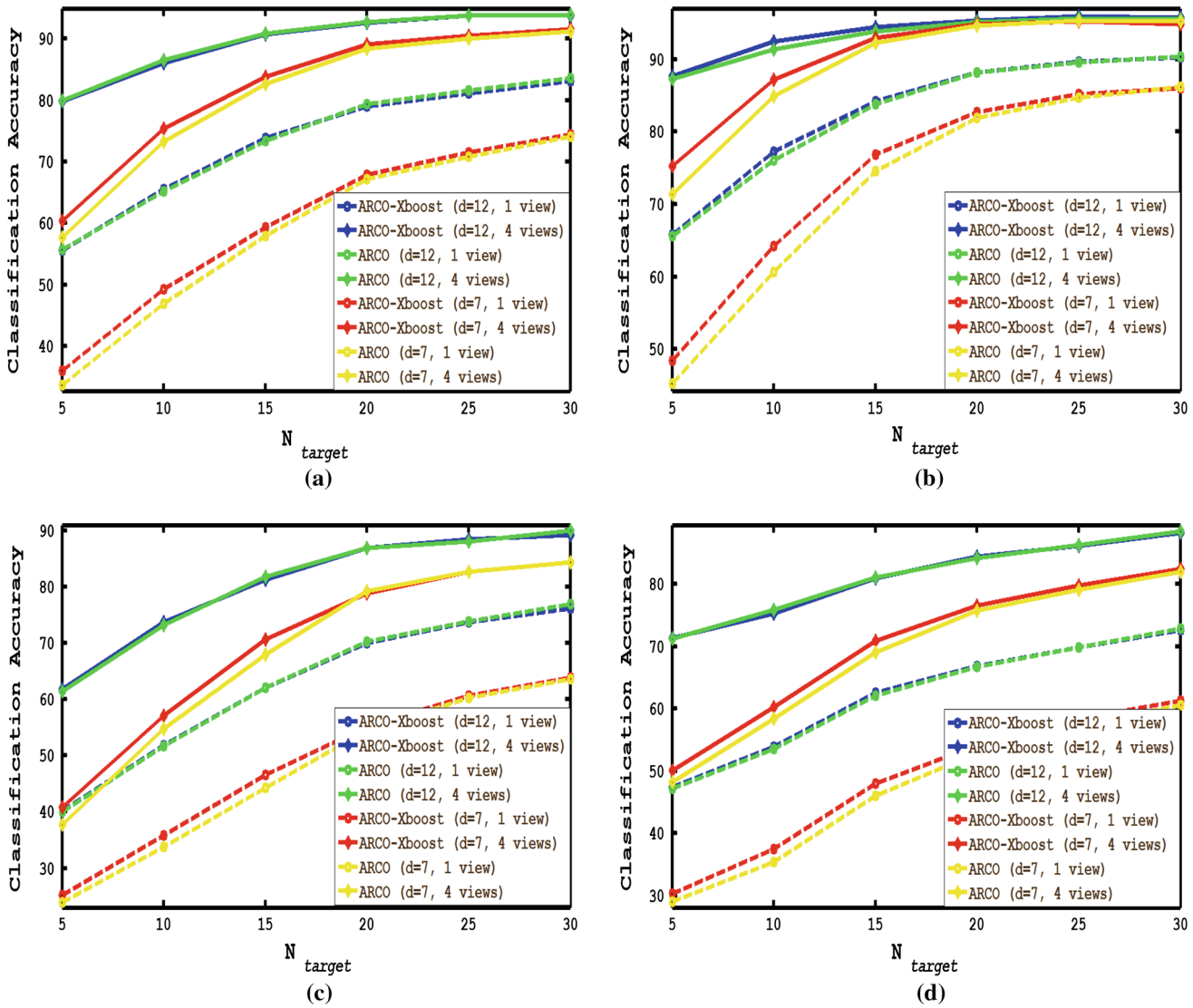

Fig. 4 Classification accuracies with ARCO and ARCO-Xboost (*red*) upon varying number of *target* training samples with $L = 12$. Results are plotted for models trained with 4-view and single-view $Cov(d = 12)$, $Cov(d = 7)$ features. Plots (**a**–**d**) show accuracies for the DPOSE *frontal*, *up*, *down* and *all* test sets

Table 2 Classification accuracies with ARCO (only *target*) and ARCO-Xboost (*source+target*) upon varying number of *target* training samples/class with $L = 12$

# Target samples	5	10	15	20	25	30
Method						
$ARCO_{(t)}$ (1-view)	42.8 ± 0.8	51.9 ± 0.5	61 ± 0.7	65.9 ± 0.4	68.7 ± 0.8	71.6 ± 0.6
$ARCO\text{-}Xboost_{(s+t)}$(1-view)	47.4 ± 0.5	54 ± 0.6	62.6 ± 0.6	66.9 ± 0.6	69.9 ± 0.7	72.6 ± 0.5
$ARCO_{(t)}$ (4-view)	60.6 ± 0.9	71.3 ± 0.5	79 ± 0.4	83.2 ± 0.3	85.1 ± 0.6	87.5 ± 0.7
$ARCO\text{-}Xboost_{(s+t)}$(4-view)	71.4 ± 0.8	$75.9.8 \pm 0.6$	80.9 ± 0.6	84.5 ± 0.6	86.2 ± 0.5	88.2 ± 0.6

Results correspond to models trained with 4-view and single-view $Cov(d = 12)$ features for the *all* test set

algorithm (Duan et al. 2009), the Domain Adaptive Metric Learning (DAML) (Kulis et al. 2011) and the Domain transfer multiple kernel learning (DTMKL) described in Duan et al. (2012). Figure 5 shows the results of our evaluation when $Cov(d = 12)$ features are used. For SVM-like methods, we considered a Gaussian kernel. The regularization parameters of all considered methods were tuned upon cross-validation. Three auxiliary classifiers are used in A-SVM and DAM,

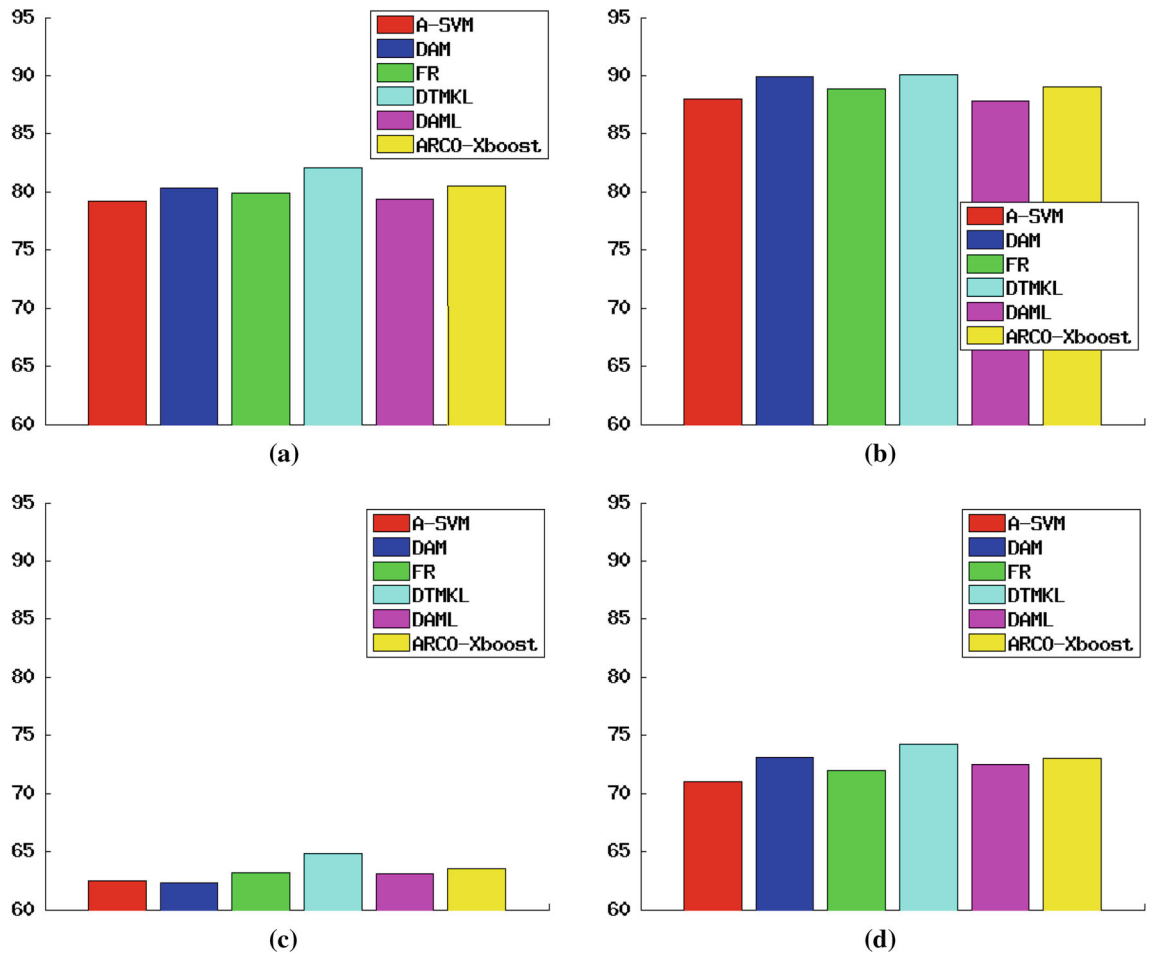

Fig. 5 Comparison with state-of-the-art transfer learning approaches. Results are plotted for models trained with *source*+5 *target* samples/class, and with 4-view $Cov(d = 12)$ features. Plots (**a**–**d**) show accuracies for the DPOSE *frontal*, *up*, *down* and *all* test sets

while 20 pre-learned base kernels are adopted in DTMKL. From Fig. 5, we observe that all transfer learning approaches achieve very similar performance, with DTMKL achieving a slightly superior accuracy. The improved performance of DTMKL can be attributed to the use of multiple kernels in the learning framework.

In all subsequent experiments, ARCO-Xboost classification accuracies obtained with $L = 12$ will be used for benchmarking. The next section discusses a second transfer learning approach for determining the head pan of a freely moving target, and why an instance-based transfer learning framework like ARCO-Xboost is unsuitable in that situation.

5 Head Pan Classification Under Target Motion

In this section, we address problem P2 introduced in Sect. 1, where the objective is to employ knowledge from *source* images capturing stationary targets to determine head pan in *target* images involving *freely moving* persons, but exhibiting the same range of head poses as in the *source*. As shown in Fig. 1, the challenge in this scenario is that facial appearance for a given pose changes with the target's position due to varying camera perspective and scale.

To this end, we propose a two-step, adaptive weights learning technique outlined in Fig. 6. First, upon dividing of the multi-view facial appearance image into a number of overlapping patches as described in Sect. 3.2, the *weight* of each patch denoting its saliency for pose classification is learnt from *source* images. These patch weights can be directly applied to the *target* dataset if it also involves stationary targets. However, since the *target* dataset involves moving persons, visibility of face patches and their reliability for pose classification would vary based on the target's position. Therefore in the second step, we transform the person's appearance in the *target* dataset to a *canonical* appearance corresponding to a *reference* spatial position,[7] and then ***adapt***

[7] In our implementation, we consider the room-center as the reference position.

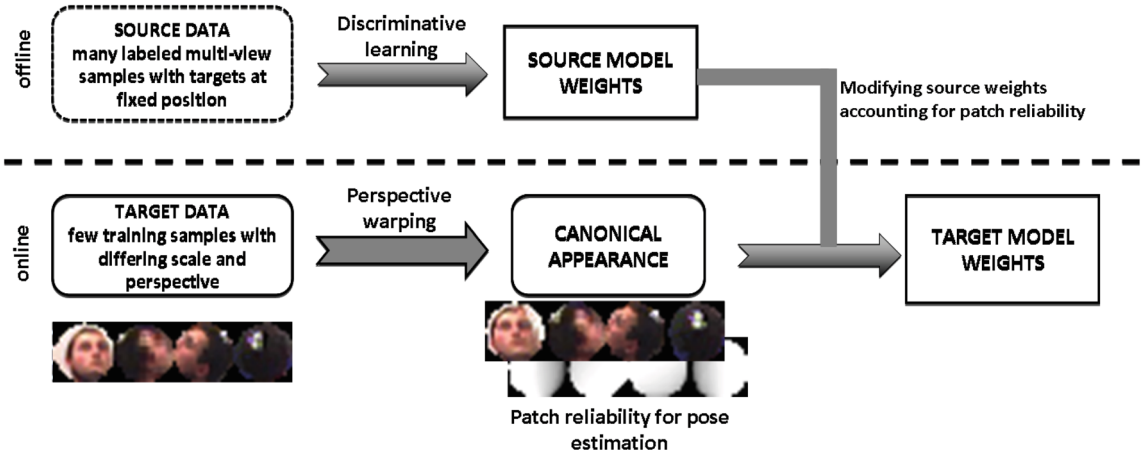

Fig. 6 Overview of the adaptive weights learning approach for head-pan classification under target motion

source patch weights to the *target* based on the visibility differences between the current and reference target positions. A notable aspect of the proposed transfer learning approach is that the target adaptation can be performed virtually online, upon acquiring very few training examples for each pose class corresponding to different room partitions. Finally, the pose class of a *target* test image is assigned using its nearest training example, computed using a weighted distance measure. The proposed transfer learning framework is formally described in the following section.

5.1 Learning a Distance Function Under Target Motion

For our problem scenario, the *source* (CLEAR) has many exemplars with persons standing at a *fixed* position, while the *target* (DPOSE) has persons imaged as they are *moving*. Formally, from the large *source* set $\mathscr{T}_s = \{(\mathbf{x}_1, l_1), (\mathbf{x}_2, l_2), \ldots, (\mathbf{x}_{N_s}, l_{N_s})\}$, we seek to transfer knowledge to the *target* incorporating additional information from a small number of *target* samples $\mathscr{T}_t = \{(\mathtt{x}_1, \mathtt{l}_1), (\mathtt{x}_2, \mathtt{l}_2), \ldots, (\mathtt{x}_{N_t}, \mathtt{l}_{N_t})\}$. 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 (Ricci and Odobez 2009):

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

In practice, the weight vector $\boldsymbol{W}_s$ with minimum norm is obtained imposing that the minimum inter-class distance exceeds the maximum intra-class distance by a margin. ξ_i's are slack variables and the parameter λ controls the trade-off between regularization and constraints violation. 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 $\mathscr{T}_t$. $\boldsymbol{W}_s$ 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. 2c) is also considered.

We first discuss the adaptation of the *source* weights to the *target*, assuming that all *target* images correspond to a *reference* position associated to the *canonical* image. We formulate the adaptation problem as:

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

where $\text{tr}(\cdot)$ denotes trace of matrix, $\boldsymbol{W} = [\boldsymbol{W}_s\ \boldsymbol{W}_t]^T$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$ is a symmetric adaptation matrix defining the dependencies between the *source* and the *target* weight vectors. The transfer learning is realized by the term $\text{tr}(\boldsymbol{W}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{W})$, and specifically by learning the *source–target* dependency matrix $\boldsymbol{\Sigma}$. This adaptation term, previously proposed in Zhang and

Yeung (2010), allows for both negative and positive transfer, and, being a convex function on the optimization parameters, makes our approach convex. Defining $\boldsymbol{\Sigma} = [\alpha \ \beta; \beta \ 1-\alpha]$[8], (2) can be rewritten as follows:

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

where we define

$$\begin{aligned}&\Delta(\alpha,\beta) = \alpha(1-\alpha)-\beta^2, \gamma_1(\alpha,\beta) = \lambda_1 + \frac{\lambda_2\alpha}{\Delta(\alpha,\beta)}, \\ &\gamma_2(\alpha,\beta) = \frac{2\lambda_2\beta}{\Delta(\alpha,\beta)}, \gamma_3(\alpha,\beta) = \frac{\lambda_2(1-\alpha)}{\Delta(\alpha,\beta)}\end{aligned} \tag{4}$$

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, $\hat{\rho} = [\rho_q], q = 1 \ldots Q$, 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 $r, r = 1 \ldots R$, 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 $p = q$ and 0 otherwise. Then the optimization problem (3) can be reformulated accounting for patch reliability as follows:

$$\begin{aligned}&\min_{\boldsymbol{W}_t,\xi_i,\alpha,\beta} \gamma_1(\alpha,\beta)||\mathbf{B}\boldsymbol{W}_t||^2 - \gamma_2(\alpha,\beta)\boldsymbol{W}_s^T\boldsymbol{W}_t \\ &\quad - \gamma_3(\alpha,\beta)||\boldsymbol{W}_s||^2 + \frac{1}{N_t}\sum_{i=1}^{N_t}\xi_i \\ &\text{s.t.} \min_{1_i\neq 1_k} \boldsymbol{W}_t^T\mathbf{d}_{ik} - \max_{1_i=1_j} \boldsymbol{W}_t^T\mathbf{d}_{ij} \\ &\geq 1-\xi_i, \quad \boldsymbol{W}_t \geq 0, \quad \xi_i \geq 0, \quad \alpha(1-\alpha)-\beta^2 > 0 \end{aligned} \tag{5}$$

Solving the Transfer Learning Optimization Problem. To solve the optimization problem (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, where we first solve with respect to $\hat{\boldsymbol{W}}_t$ keeping α, β fixed, and then, given a certain distance function we compute the optimal adaptation weights α, β. The optimization problems that must be solved are:

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

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

$$\min_{\boldsymbol{\theta}} \mathbf{a}^T\boldsymbol{\theta} \quad \text{s.t.} \quad \boldsymbol{\theta}^T\mathbf{I}\boldsymbol{\theta} - \mathbf{e}^T\boldsymbol{\theta} \leq 0 \tag{7}$$

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

As for the *source* data, 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. The optimization problem (7) can be reduced to a Second Order Cone Programming (SOCP) problem and it is solved efficiently using SEDUMI[9]. 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 3.

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 iterations T and the sample size k ($T = 100$ and $k = 5$ in our experiments).
$\boldsymbol{w} = 0$.
for $t = 1, \ldots, T$ **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
$\{(\hat{\mathbf{x}}_j, l_j), (\hat{\mathbf{x}}_k, l_k) \in \mathcal{T} :$
$\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_j := \arg\max_{l_i\neq l_k, l_i=l_j,}\left[1 - \boldsymbol{w}^T\hat{\mathbf{d}}_{ij}^{\mathbf{M}} + \boldsymbol{w}^T\hat{\mathbf{d}}_{ik}^{\mathbf{M}}\right]\}$
$\boldsymbol{w}^t = \left(1 - \frac{1}{t}\right)\boldsymbol{w}^t +$
$\frac{1}{k\theta_1 t}\sum_{(\mathbf{x}_i,l_i)\in\mathcal{T}^+}\left[\hat{\mathbf{d}}^{\mathbf{M}}(\mathbf{x}_i, \hat{\mathbf{x}}_k) - \hat{\mathbf{d}}^{\mathbf{M}}(\mathbf{x}_i, \hat{\mathbf{x}}_j)\right] - \frac{\theta_2}{\theta_1 t}\boldsymbol{w}_o$
$\boldsymbol{w}^t = \max\{0, \boldsymbol{w}^t\}$
$\boldsymbol{w}^t = \min\{1, \frac{1}{\sqrt{\theta_1}\|\boldsymbol{w}^t\|}\}\boldsymbol{w}^t$
endfor
}

5.2 Experimental Results and Discussion

We now evaluate the adaptive weighted distance learning framework for pose classification under target motion against: (i) ARCO-Xboost, described in Sect. 4 and (ii) the

[8] $\boldsymbol{\Sigma}$ is chosen to be positive semi-definite and have a trace equal to 1 as proposed in Kulis et al. (2011)

[9] http://sedumi.ie.lehigh.edu/

(R1) (R2) (R3) (R4)

Fig. 7 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 8-class head-pan classification under target motion

	ARCO-Xboost $Cov\ (d = 7)$	ARCO-Xboost $Cov\ (d = 12)$	WD $Cov\ (d = 7)$	WD $Cov\ (d = 12)$	WD LBP	Multi-view SVM
R1	41.1 ± 0.9	66.1 ± 1.2	65.8 ± 1.2 (33.1)	69.8 ± 1.1 (45)	**74.7** ± 1.1 (60.9)	47.6 ± 1.2
R2	43.6 ± 1.2	67.6 ± 1.3	67.4 ± 1.1 (41.5)	72.4 ± 1.2 (51.6)	**77.6** ± 1.2 (61.3)	51.3 ± 1
R3	45.9 ± 1	66.2 ± 1	59.6 ± 1.3 (51.2)	63 ± 1.1 (59.6)	**66.9** ± 0.9 (58.7)	41 ± 0.9
R4	41.7 ± 1.2	59.1 ± 1.3	60.6 ± 1.2 (37.8)	62.4 ± 1.4 (42.3)	**64.5** ± 1.1 (58.3)	41.6 ± 1
Regions average	43.1 ± 1	64.8 ± 1.2	63.4 ± 1.2(40.9)	66.9 ± 1.1(49.7)	**70.9** ± 1 (59.8)	45.4 ± 1

The room is divided into 4 quadrants (R1–R4). Classification accuracies are computed using a training set comprising 2400 *Source* training examples (300 samples/class) and 160 *target* examples (5 samples/class/region). # Test = 2399 (R1), 3185 (R2), 3048 (R3), 2996 (R4). NWD accuracies are reported within braces

Best performances are shown in Bold

Algorithm 3 Adaptive Weights Learning

Input: The *source* and *target* training data $\mathscr{T}_s$, $\mathscr{T}_t$.

Learning on Source Data

Set λ_1 to a fixed value ($\lambda_1 = 1$ in our experiments).

$\boldsymbol{W}_s$=**ComputeDistance**($\mathscr{T}_s$, λ_1, 0, $\bar{\mathbf{0}}$, $\mathbf{I}$);

Learning on Target Data

Compute patch reliability matrix $\mathbf{B}$.

Set λ_1 and λ_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**($\mathscr{T}_t$, γ_1, γ_2, $\hat{\boldsymbol{W}}_s$, $\mathbf{B}$);

Given $\hat{\boldsymbol{W}}_s$, $\hat{\boldsymbol{W}}_t$ compute α, β solving (7).

end

Compute $\boldsymbol{W}_t = \mathbf{B}^{-1}\hat{\boldsymbol{W}}_t$.

Output: $\boldsymbol{W}_t$

multi-view SVM (MSVM)-based pose estimation proposed in Muñoz-Salinas et al. (2012). MSVM-based pose estimation feeds gradient features from the target appearance image in each camera view to a multi-class SVM classifier, the output of which is used to compute a probability distribution over all pose classes. Then, a combined distribution fusing the multi-view information is computed for determining the pose class.

Since a region-specific $\mathbf{B}$ matrix is used in the adaptation framework, we divide the scene of interest into $R = 4$ non-overlapping regions and assume that a few *target* training examples are available per quadrant. Region-wise classification accuracies achieved using only 5 *target* samples/class/quadrant are presented. Figure 7 shows the mean reliability masks computed through the perspective warping procedure (Sect. 3.2) from *target* training examples in each quadrant. These masks demonstrate why we opt for region-based patch weight learning for the *target*. 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 moves, the face patches visible in the canonical view also vary, and visibility of a face patch modulates its saliency for pose classification.

Apart from $Cov(d = 7)$ and $Cov(d = 12)$ features, we also employ 64 bin-indexed local binary pattern (LBP) descriptors (Wang et al. 2009) to learn face patch weights using the proposed framework. Furthermore, we analyze how learning of patch weights is beneficial by comparing classification accuracies achieved with a nearest-neighbor (NN) classifier employing the weighted (WD) and unweighted (NWD) distance measures.[10] Table 3 presents the region-wise classification results. NWD classification accuracies obtained with the different features are indicated in braces. Optimal values of the regularization parameters λ_1 and λ_2 (whose values are reported in Algorithm 3), are set using a separate validation set.

Considering the mean classification accuracy over all quadrants, we make the following observations. Evidently, learning of face patch weights through the proposed adaptive framework is immensely beneficial under target motion. For all the features used, WD accuracies are much higher than NWD accuracies. In Sect. 4.1, we observed that transfer learning is more beneficial when weaker features are employed for learning. Results obtained with weighted distance learning are consistent with that observation. Best WD

[10] The NN classifier assigns the class label of the nearest *target* training example to the test image.

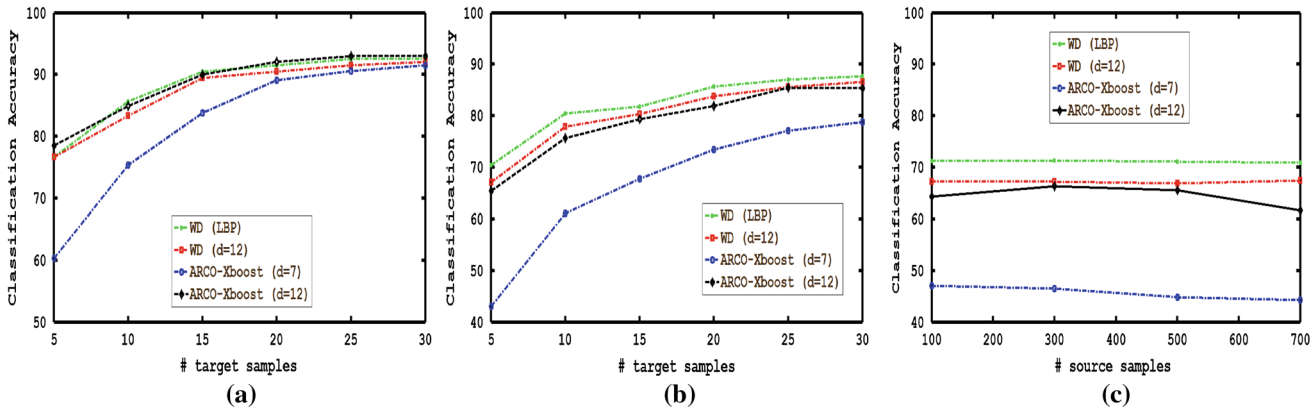

Fig. 8 (**a**) Variation in head pose classification performance with increasing number of *target* examples with stationary target. Variation in classification performance upon increasing number of (**b**) *target* examples and (**c**) *source* examples under target motion

accuracy is obtained with LBP, followed by $Cov(d = 12)$ and $Cov(d = 7)$ features. However, performance gain with the learning of patch weights is greatest for $Cov(d = 7)$ (gain of 55 % with WD and NWD accuracies respectively being 63.4 and 40.9), followed by $Cov(d = 12)$ (34.6 % gain) and LBP (18.6 % gain).

ARCO-Xboost and WD classification accuracies are predictably higher for $Cov(d = 12)$ as compared to $Cov(d = 7)$ features. Adaptive weights learning comfortably outperforms ARCO-Xboost with $Cov(d = 7)$ features, while WD accuracy is slightly higher than ARCO-Xboost with $Cov(d = 12)$ descriptors. Comparing the best WD and ARCO performances achieved with LBP and $Cov(d = 12)$ features respectively, WD outperforms ARCO-XBoost by 9.5 % (70.9 vs 64.8). Also, classification performance achieved using MSVM is only slightly better than ARCO-Xboost with $Cov(d = 7)$ features. This is because MSVM uses only gradient features for learning, and is not designed to handle appearance changes arising from varying target position.

While adaptive weighted distance learning is designed under the assumption that acquiring many head pose training examples under target motion is expensive, we also analyzed how ARCO-Xboost and WD classification accuracies vary when the number of *target* training examples vary from 5 to 30 samples/class—results are presented in Fig. 8. Figure 8a shows ARCO-Xboost and WD accuracies when the target position is fixed at the room-center (as in Sect. 4), while Fig. 8b presents mean accuracy plots for the moving target scenario (we assume 5–30 *target* samples/class/region here). Very similar accuracies are achieved with both distance learning and ARCO-Xboost when large *target* training sets are employed for the stationary case. Nevertheless, WD outperforms ARCO-Xboost even with large *target* training sets asssuming freely moving targets. With 30 DPOSE training samples/class/region, accuracies achieved with WD (LBP), WD ($Cov,\ d = 12$), ARCO-Xboost ($Cov,\ d = 12$) and ARCO-Xboost ($Cov,\ d = 7$) are 87.7, 86.6, 85.3 and 78.7 respectively.

We also examined the impact of varying the number of *source* training examples on WD and ARCO-Xboost classification performance, with 5 *target* examples/class/region in the training set—Fig. 8c presents the results. While the size of the *source* training set has little influence on WD classification accuracy, a small reduction in ARCO-Xboost accuracy is observed for large *source* training set sizes. This is because when the *target* to *source* training data ratio is very low, *source* data dominate the learning process resulting in a *source*-tuned model. In such cases, Pardoe and Stone (2010) note that many iterations are required to obtain misclassified *target* weights comparable to *source* weights employing the re-weighting scheme used in boosting frameworks such as Dai et al. (2007).

Finally, it is pertinent to point out two design-related differences between the ARCO-Xboost and weights-based transfer learning approaches. First, adaptive weighted-distance learning explicitly considers reliability of face patches in the learning process unlike ARCO-Xboost. However, the ARCO-Xboost learning framework is inherently robust, where a classifier is trained for *every patch* and the sample class is determined based on the majority vote of all patch classifiers. We noted earlier that peripheral face patches are affected more than central face patches by perspective and scale changes under target motion. But the inferior pose classification performance of ARCO-Xboost suggests that facial appearance variations under motion are not just restricted to a few face patches—this demonstrates that head pose estimation for freely moving targets is a non-trivial and salient research problem. A second difference is that ARCO-Xboost, being an instance-based transfer learning approach, requires retraining each time the *target* data attributes change (e.g., varying scene geometry and illumination conditions) which

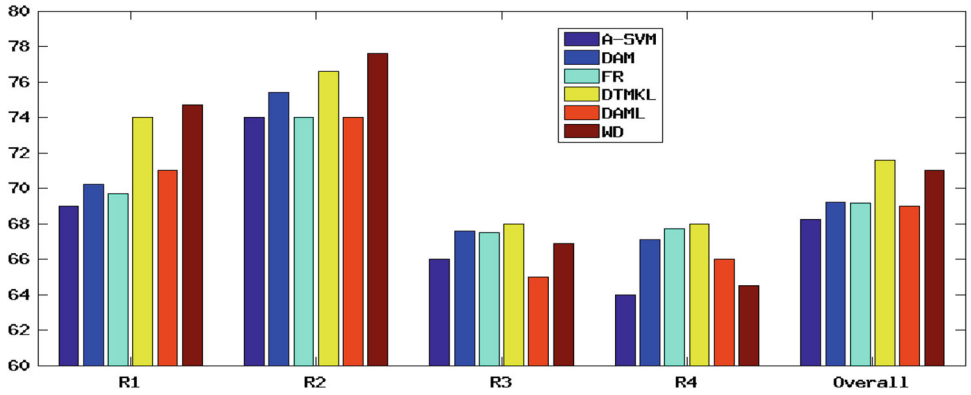

Fig. 9 Comparison with state-of-the-art transfer learning approaches for scenario P2. Experiments are performed using *source*+5 *target* samples/class/qudrant, and with LBP features

is time and computation-intensive. In contrast, the adaptive weighted distance learning approach employs a two-step process: *source* weights are learned in the first step, and this learning is performed exactly once. With varying *target* attributes, *target*-specific adaptation can be achieved almost on-the-fly since this learning process requires only a few training examples.

A further series of experiments were conducted for comparing WD performance with other state-of-the-art transfer learning methods (Fig. 9). The adaptive weights learning approach, which explicitly incorporates camera geometry information in the learning framework, outperforms most other competing approaches. However, DTMKL, which is a powerful framework employing 20 pre-learned kernel classifiers for domain adaptation, produces the highest classification accuracies.

It needs to be noted here that while some competing methods use multiple auxiliary (*source*) classifiers for knowledge transfer (e.g., DAM uses three auxiliary classifiers), our approach employs only a single *source* classifier for transfer learning. Extending our current transfer learning framework to integrate knowledge from multiple sources and incorporate kernel learning will be the focus of future work. Finally, facial feature representation influences the performance of all methods and that in turn, dependent on the quality of facial cropping. In addition to its utility for transfer learning under target motion, camera geometry information is also used by the 3D tracker employed in our framework to enable accurate face localization and cropping of moving targets.

We also show some qualitative results obtained with the adaptive weights learning approach in Fig. 10. Figure 10a, b correspond to a single moving target, while Fig. 10c shows computed pose labels for 2 of 6 freely moving targets having an informal conversation as in a party. Figure 10a corresponds to a correct result, while Fig. 10b shows an incorrect result, because the face localization and ensuing face crops (on the top-right inset) are erroneous. Figure 10c demonstrates that this approach can work well even with multiple targets. While no pose ground-truth was available for this sequence, the computed pose labels can be observed to be correct from visual inspection.

6 Head Pose Classification in Naturalistic Settings

Now, we focus on our original problem P3: learning from many training examples where stationary targets exhibit a frontal head-tilt and adapting this knowledge to determine head pose (both pan and tilt) of freely moving targets showing unrestricted head movements. While P3 essentially represents the combination of afore-discussed problems P1 and P2, we also have an unequal number of *source* and *target* classes here—the range of *source* head poses is discretized into 8 classes, while the *target* head pose range is divided into 24 classes (arising from 8 pan and 3 tilt intervals).

To address the adaptation problem where no *source* training examples are available for a number of classes, we adapt the transferable distance learning approach proposed in Yang et al. (2010, 2009). Inspired by Ferencz et al. (2008), where *hyperfeatures* measuring saliency of patches are used for object identification, a transferable framework employing hyperfeatures for action recognition is described in Yang et al. (2009). Here again, samples are compared using a weighted-distance measure $D = \langle w \cdot d_{ij} \rangle$, where w is a vector of patch weights, d_{ij} is a Q dimensional vector denoting patch-based distance between samples i, j and $\langle \cdot \rangle$ denotes dot product. Patch weights are defined as a linear function of the patch hyperfeature matrix $\mathbf{F}$, i.e., $w = P^T\mathbf{F}$, where P denotes a vector of *transferable* parameters. If the similarity between the *source* and *target* datasets is effectively captured by the hyperfeatures, P learnt on *source* data can be directly applied to compute the saliency of a *target* patch from its hyperfeatures *without any* learning on the *target*. It is therefore possible to classify *target* data even when only a single example per class is available.

The patch hyperfeature, which captures its saliency for classification, is calculated using a codebook approach. A codebook vocabulary of size $|C|$ is obtained by performing

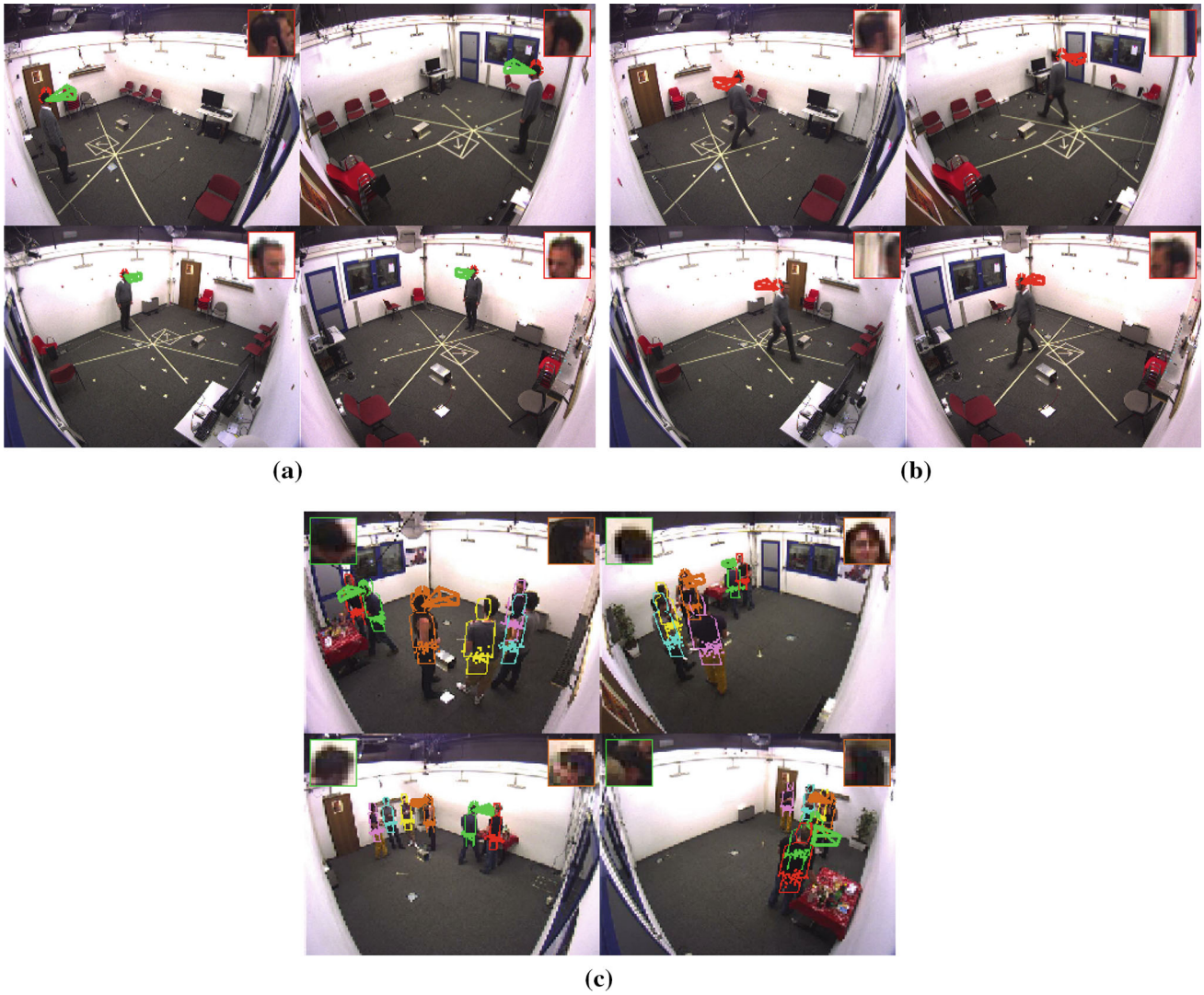

Fig. 10 Head pose estimation results with target moving (**a**, **b**). *Green cone* indicates accurate pan estimation while the *red cone* denotes wrongly predicted pose label. **c** Results with the proposed approach for a *party* scenario involving multiple targets (corresponding video is available is provided as supplementary material)

k-means clusterng on features accumulated over all *source* patches. The *j*th element of the hyperfeature matrix corresponding to the *i*th patch, F_{ji}, is then computed as the normalized distance between the patch feature h_i and the *j*th codebook word, c_j, as given below

$$F_{ji} = \frac{K_\sigma(\mathrm{d}(h_i, c_j))}{\sum_{k=1}^{|C|} K_\sigma(\mathrm{d}(h_i, c_k))} \tag{8}$$

where $K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(\frac{-x^2}{2\sigma^2})$ is the Gaussian kernel with appropriately chosen σ and d(.) denotes Euclidean metric. The transferable parameter P is learnt from *source* data by solving the dual of a max-margin optimization problem in Yang et al. (2009). We formulate the max-margin optimization problem as

$$\min_{\boldsymbol{P}, \xi_i \geq 0} \frac{\lambda}{2} ||\boldsymbol{P}||^2 + \frac{1}{N_s} \sum_{i=1}^{N_s} \xi_i$$
$$\text{s.t.} \quad \min_{l_i \neq l_k} \langle \boldsymbol{P^T F_i}, \mathbf{d}_{ik} \rangle - \max_{l_i = l_j} \langle \boldsymbol{P^T F_i}, \mathbf{d}_{ij} \rangle \quad \geq 1 - \xi_i$$

This primal formulation, as such, is solved using the stochastic gradient descent method outlined in Algorithm 2. Finally, since the *target* dataset involves freely moving persons, we modulate the patch weights by their reliability scores (as given by the B matrices) to account for appearance distortions at positions other than the reference location (room-center). To this end, as previously, we divide the scene into 4 non-overlapping quadrants and compute B for each region from *target* training examples. Based on the target's position as given by the person tracker, the appropriate B is applied to compute NN distance for determining the head pose class

Algorithm 4 Computing Head Pose under Target Motion Using Transferable Distance with Hyperfeatures and Patch reliability Scores

Input: *Source* and *target* training data $\mathcal{T}_s$, $\mathcal{T}_t$. *Target* test data $\mathcal{T}e_t$.

Learning on Source Data
Computing Source Hyperfeatures
Codebook $\{C\}$ are the $|C|$ centers obtained using k-means clustering on patch features extracted from $\mathcal{T}_s$.
Compute sample hyperfeature matrix $\mathbf{F}_i$ $\forall i = 1 \ldots |\mathcal{T}_s|$ using (8) ($\sigma = 1$, $|C| = 150$ in our experiments).

Set λ_1 to a fixed value ($\lambda_1 = 1$ in our experiments).
Substituting $\boldsymbol{w}$ by $P^T\mathbf{F}_i$ in Algorithm 2, output $\boldsymbol{P}$=**ComputeDistance**($\mathcal{T}_s$, λ_1, 0, $\tilde{\mathbf{0}}$, $\mathbf{I}$).

Determining $\mathbf{B}$*'s from target training data*
Compute patch reliability matrix $\mathbf{B}_r$ for each region $r = 1 \ldots R$ from appropriate training examples in $\mathcal{T}_t$.

Testing on target data
for $i = 1, \ldots, |\mathcal{T}e_t|$ **do**
Calculate hyperfeature matrix $\mathbf{F}_i$ for test sample.
for $j = 1, \ldots, |\mathcal{T}_t|$ **do**
if $\{\mathcal{T}_t^j, \mathcal{T}e_t^i\} \in r$
Compute patch-based Euclidian distance between samples d_{ij}
Vector of patch weights $w_i = P^T\mathbf{F}_i$
Reliability-modulated patch weight vector $\hat{w}_i = \mathbf{B}_r w_i$
Weighted distance $\hat{D}_{ij} = \langle \hat{w}_i \cdot d_{ij} \rangle$
endif
endfor
Output: Pose class label of $\mathcal{T}e_t^i = \arg\min_j \hat{D}_{ij}$
endfor

of a test instance. The entire procedure is outlined in Algorithm 4.

6.1 Experimental Results and Discussion

In this section, we evaluate the transferable distance framework for determining head pose on the *target*, comprising 24 pose classes and moving persons, upon learning from the *source* comprising 8 pose classes and stationary persons. To this end, we compare classification accuracies achieved with ARCO-Xboost, multi-view SVM and the transferable distance approaches assuming that (i) the person rotates in-place and (ii) is freely moving in the *target* dataset. Tables 4, 5, 6 and 7 present the respective results. We also examine classification accuracy for source classes (acc_{sc}) where *source* examples are available, non-source classes (acc_{nsc}) for which no *source* training data exist, and overall accuracy (acc_{ov}).

We assume that 5 *target* samples/class are available for nearest neighbor comparison in the fixed target scenario. Assuming freely moving targets, we again consider two conditions: (a) *localized train-test setting* where 5 *target* examples/class are available per quadrant and test samples arise from the same region as the *target* training data and (b) *sparse training data setting*, where test samples span all regions but only 5 *target* examples/class are available (ensuring at least one *target* example/class/region). Condition (a) represents an identical setting as in Sect. 5.2, where region-specific $\mathbf{B}$ matrices are employed for adaptation. While *target* learning is not required for the transferable distance framework, condition (a) denotes the situation where the *target* training and test sets are homogeneous with respect to perspective and scale-related appearance variations, while condition (b) represents a more challenging scenario, involving fewer *target* examples and a larger, heterogenous test set.

For the transferable distance framework, two types of descriptors are needed—one for computing inter-sample distance, and another to compute hyperfeatures characterizing patch saliency. For the distance metric, we used $Cov(d = 12)$ and LBP features following Sect. 5.2. For computing patch hyperfeatures, we used 66-dimensional feature vectors combining patch centroid coordinates with 64-bin HoG (Dalal and Triggs 2005) or LBP descriptors. The presented results correspond to the $Cov(d = 12) + LBP$, $Cov(d = 12) + HoG$ and $LBP + HoG$ distance feature+hyperfeature combinations. As before, we compare classification accuracies achieved using the weighted distance (WD) measure against the Euclidian distance (NWD) measure in the distance feature space. A codebook size of 150 is used to generate hyperfeatures—we observed that the codebook size had little influence on classification accuracy upon varying the codebook size from 50-500.

From the classification results presented in Tables 4–7, we make the following remarks. In Table 4, acc_{sc} values are higher for ARCO-Xboost as compared to WD. This can be also observed examining Table 5. This is in contrast to the trends observed in Sect. 5.2. A key difference between adaptive weights learning and transferable distance frameworks is that explicit learning on *target* is performed in the former incorporating patch reliability information to modify patch weights learnt on *source* data. But no learning on the *target* is performed with transferable distance learning. We simply modulate the saliency weight of a face patch by its reliability to account for appearance variations under motion in the *target* dataset in this case.

On the other hand, low accuracies are achieved with both ARCO-Xboost and multi-view SVM for non-source classes in all cases (Tables 4, 6), which adversely impacts overall accuracies as well (Tables 4, 7). With no *source* training examples available for non-source classes, only *target* training examples are utilized for learning. A Euclidian distance-based nearest neighbor classifier consistently performs better than ARCO-Xboost and MSVM in this scenario, as seen from NWD accuracies. These results demonstrate that standard machine learning techniques do not work well with few

Table 4 Comparison of classification accuracies obtained with different approaches when target position is fixed at room-center

	ARCO-Xboost $Cov\ (d = 7)$	ARCO-Xboost $Cov\ (d = 12)$	WD $Cov\ (d = 12) + LBP$	WD $Cov\ (d = 12) + HoG$	WD $LBP + HoG$	Multi-view SVM
acc_{sc}	57 ± 1.1	**73.8** ± 0.9	$56.1 \pm 1\ (49.7 \pm 0.9)$	$56.8 \pm 0.6\ (49.7 \pm 0.8)$	$59 \pm 1\ (51.8 \pm 0.8)$	57.5 ± 0.7
acc_{nsc}	14.6 ± 0.8	31.1 ± 0.7	$60.1 \pm 1\ (58.3 \pm 0.5)$	$59.4 \pm 0.7\ (58.3 \pm 0.5)$	**60.4** $\pm 0.7\ (58.9 \pm 0.8)$	27.6 ± 0.6
acc_{ov}	28.8 ± 0.8	45.4 ± 0.7	$58.3 \pm 0.8\ (54.4 \pm 1)$	$58.1 \pm 0.7\ (54.4 \pm 0.9)$	**59.7** $\pm 0.8\ (55.4 \pm 0.7)$	41 ± 0.6

Source Training set comprises 300 samples/class for 8 frontal tilt classes. *Target* training set comprises 5 examples/class for 24 pose classes. # Test = 25424. NWD accuracies are reported in braces
Best performances are shown in Bold

Table 5 Classification accuracies for source classes assuming freely moving targets

	ARCO-Xboost $Cov\ (d = 12)$	WD $Cov\ (d = 12) + LBP$	WD $Cov(d = 12) + HoG$	WD $LBP + HoG$	Multi-view SVM
R1	**60.7** ± 1.5	$49.2 \pm 1.9\ (46.6 \pm 2.1)$	$50 \pm 2\ (46.6 \pm 1.9)$	$53.3 \pm 1.7\ (51.4 \pm 1.8)$	48.2 ± 1.3
R2	**61.7** ± 1.5	$46.4 \pm 1.6\ (44.3 \pm 1.5)$	$46.1 \pm 1.8\ (44.3 \pm 1.7)$	$47.3 \pm 1.5\ (46 \pm 1.4)$	25.7 ± 1.2
R3	**68.3** ± 1.3	$52.5 \pm 1.8\ (48.9 \pm 1.5)$	$51.3 \pm 1.5\ (48.9 \pm 1.4)$	$54.5 \pm 1.4\ (51.1 \pm 1.4)$	23 ± 1.4
R4	**62.3** ± 1.1	$46 \pm 1.5\ (44.4 \pm 1.6)$	$47.6 \pm 1.5\ (44.4 \pm 1.4)$	$50.8 \pm 1.6\ (47.6 \pm 1.1)$	31.9 ± 1
Region Average	**63.3** ± 1.4	$48.5 \pm 1.7\ (46.1 \pm 1.7)$	$48.8 \pm 1.7\ (46.1 \pm 1.6)$	$52.5 \pm 1.6\ (49 \pm 1.4)$	32.2 ± 1.2
all	**53** ± 0.9	$35.8 \pm 0.9\ (32 \pm 0.8)$	$34.4 \pm 1.1\ (32 \pm 0.9)$	$35.6 \pm 0.9\ (33.4 \pm 1)$	29.8 ± 0.5

Source Training set comprises 300 samples/class for 8 frontal tilt classes. The space is divided into 4 quadrants R1-R4. Results are presented for the *localized train-test setting* and *sparse training data setting* ('all' condition) considered above. # Test = 4664 (R1), 6330 (R2), 6249 (R3), 5536 (R4) and 22779 (all). NWD accuracies are reported in braces
Best performances are shown in Bold

Table 6 Classification accuracies for non-source classes assuming freely moving targets

	ARCO-Xboost $Cov\ (d = 12)$	WD $Cov\ (d = 12) + LBP$	WD $Cov\ (d = 12) + HoG$	WD $LBP + HoG$	Multi-view SVM
R1	45.9 ± 1.1	$60 \pm 1.8\ (59.1 \pm 1.7)$	**61.2** $\pm 1.6\ (59.1 \pm 1.7)$	$59 \pm 1.5\ (58.8 \pm 1.5)$	27.4 ± 1.1
R2	41 ± 1.4	**64** $\pm 1.8\ (62.3 \pm 1.6)$	$63.4 \pm 2.1\ (62.3 \pm 1.6)$	$61.1 \pm 1.8\ (60 \pm 1.7)$	11.2 ± 1.1
R3	43.8 ± 1.2	**65.6** $\pm 1.6\ (62 \pm 2)$	$64.6 \pm 1.8\ (62 \pm 2)$	$62.8 \pm 1.9\ (61.8 \pm 2)$	8.4 ± 1.3
R4	40.6 ± 1.1	$60.3 \pm 1.8\ (57.2 \pm 1.5)$	**60.8** $\pm 1.6\ (57.2 \pm 1.5)$	$57.7 \pm 1.5\ (55.9 \pm 1.7)$	14 ± 1.1
Region Average	42.9 ± 1.2	**62.5** $\pm 1.7\ (60.2 \pm 1.7)$	**62.5** $\pm 1.8\ (60.2 \pm 1.7)$	**60.2** $\pm 1.7\ (59.1 \pm 1.7)$	15.3 ± 1.1
all	16 ± 1	$43.4 \pm 0.9\ (38.8 \pm 0.8)$	$40.4 \pm 1\ (38.8 \pm 1)$	**44.1** $\pm 1.3\ (40.9 \pm 1.1)$	10.2 ± 0.9

Results are presented for the *localized train-test setting* and *sparse training data setting* ('all' condition) considered above
Best performances are shown in Bold

training data, which is why transfer learning is adopted to leverage knowledge from related and extensively annotated datasets.

Comparing NWD and WD accuracies for source classes, the largest gain in WD accuracy is observed with stationary targets—a maximum gain of 14.3 % (56.8 vs 49.7) is obtained with the $Cov(d = 12) + HoG$ combination for the stationary target case. With freely moving targets, maximum gain of 5.9 % (48.8 vs 46.1) is obtained with $Cov(d = 12) + HoG$ features for the localized train-test condition (considering mean accuracy over all regions), while highest gain achieved for the sparse training data condition is 11.9 % with $Cov(d = 12) + LBP$ features. For non-source classes, maximum accuracy gains for the stationary target, localized train-test and sparse training data conditions are 3.1 % ($Cov(d = 12)+LBP$), 3.8 % ($Cov(d = 12)+LBP$) and 11.9 % ($Cov(d = 12) + LBP$) respectively. Collectively, these gains suggest that transferable distance learning improves pose prediction performance with respect to a Euclidian distance-based NN classifier for both source and non-source classes.

As with adaptive weights learning, LBP features produce best classification performance with transferable distance learning also. NWD accuracies are consistently higher with

Table 7 Overall classification accuracies with freely moving targets

	ARCO-Xboost $Cov\ (d = 12)$	WD $Cov\ (d = 12) + LBP$	WD $Cov\ (d = 12) + HoG$	WD $LBP + HoG$	Multi-view SVM
R1	50.8 ± 1.6	$54 \pm 2\ (52.7 \pm 1.5)$	$55.5 \pm 1.9\ (52.7 \pm 1.5)$	$\mathbf{56} \pm 1.6\ (55 \pm 1.5)$	38.1 ± 1.3
R2	47.9 ± 1.5	$\mathbf{55.1} \pm 1.4\ (53.2 \pm 1.5)$	$54.7 \pm 1.5\ (53.2 \pm 1.5)$	$54.1 \pm 1.5\ (53 \pm 1.4)$	16.1 ± 1.5
R3	51.9 ± 1.5	$\mathbf{59.2} \pm 1.6\ (55.6 \pm 1.4)$	$58.1 \pm 1.7\ (55.6 \pm 1.4)$	$59.2 \pm 1.3\ (56.8 \pm 1.2)$	14 ± 1.3
R4	47.8 ± 1.4	$52.5 \pm 1.3\ (50.3 \pm 1.2)$	$53.4 \pm 1.4\ (50.3 \pm 1.2)$	$\mathbf{54} \pm 1.2\ (51.4 \pm 1.2)$	17.2 ± 1.1
Region Average	49.6 ± 1.5	$55.2 \pm 1.6\ (53 \pm 1.4)$	$55.4 \pm 1.4\ (53 \pm 1.3)$	$\mathbf{55.8} \pm 1.4\ (54.1 \pm 1.3)$	21.4 ± 1.3
all	28.3 ± 0.9	$39.5 \pm 1\ (35.3 \pm 1)$	$37.3 \pm 0.9\ (35.3 \pm 1)$	$\mathbf{39.8} \pm 0.9\ (37.1 \pm 1)$	18.7 ± 0.9

Results presented for the *localized train-test setting* and *sparse training data setting* ('all' condition) considered above
Best performances are shown in Bold

LBP as compared to covariance features, and the best WD accuracies are observed with the $LBP + HoG$ combination for most cases. Conversely, larger gains with a weighted-distance measure are observed for covariance features. However, it is difficult to judge the better of LBP and HoG for hyperfeature representation from the observed results.

Considering head pose classification for moving targets, a mean overall accuracy of 55.8 % with $LBP + HoG$ represents the best 24 class prediction performance achieved for the localized train-test setting. For the more challenging sparse training data setting, a highest accuracy of 39.8 % is obtained for the same feature combination. Nevertheless, these accuracies are still higher than those achieved with ARCO-Xboost and multi-view SVM as elaborated above. MSVM in particular, performs very poorly with freely moving targets.

7 Summary and Conclusion

This paper represents the first work to explore transfer learning approaches for multi-view head pose classification and in particular, pose classification under target motion, for which very few solutions have been proposed in literature. Since direct learning of pose-related appearance variations under motion would require expensive labeling of a large number of examples, adapting knowledge from annotated datasets with stationary targets is a viable alternative. We propose and evaluate transfer learning solutions for three situations where the *source* and *target* datasets differ with respect to (i) the range of head poses exhibited by targets, denoted as P1 (ii) the nature of targets involved (stationary vs mobile targets), denoted as P2 and (iii) the combination of (i) and (ii) denoted by P3.

ARCO-Xboost, a transfer learning-based adaptive version of the ARCO head pose classifier (Tosato et al. 2010) is first proposed, and is shown to outperform ARCO when very few *target* examples are added to the training set or weaker features are employed for learning. ARCO-Xboost is then used as a benchmark for evaluating other adaptation methods. We observe that the ARCO-Xboost pose classification approach does not work well for freely moving targets, or when only few training examples are available for learning.

To determine head pose of freely moving persons in the *target* dataset, two parameter transfer learning approaches are considered. First, an adaptive weights learning approach is proposed where a set of face patch weights, representative of their saliency for pose classification, are learnt on the *source* dataset. These weights are then adapted to the *target* dataset upon learning from a few *target* examples, incorporating information concerning patch reliability for pose classification under target motion.

A second transferable distance learning method adapted from Yang et al. (2010) assumes that saliency of both *source* and *target* face patches can be learnt through characteristic *hyperfeatures*. Therefore, upon learning the mapping between hyperfeatures and patch saliency on the *source*, the same mapping is directly applied on the *target* without any further learning. To account for appearance distortions under motion, *target* patch saliency weights are modulated by their reliability scores. While transferable distance learning improves pose prediction on *target* data with respect to a Euclidian distance-based nearest neighbor classifier, even for pose classes unseen in the *source*, the improvements are not as high as those achieved with adaptive weights learning.

We also compared the ARCO-Xboost and adaptive weights learning methods with other state-of-the-art transfer learning approaches. For P1, all considered approaches produced similar classification performance, with DTMKL achieving slightly higher accuracies. DTMKL again produced the highest classification accuracies for P2, but the proposed WD classifier outperformed all other competing approaches due to the explicit incorporation of camera geometry information in the transfer learning framework.

Overall, the proposed transfer learning solutions are novel in the context of multi-view head pose classification under

target motion, which is a relevant and important research problem in applications such as surveillance and human behavior understanding. To aid further research in this domain, the extensive dynamic headpose (DPOSE) dataset is presented in this paper. Future research involves (i) integration of knowledge from multiple sources and incorporation of kernel learning in our framework, and (ii) use of multi-task learning and weakly supervised domain adaptation approaches integrating information from multiple sources (such as body pose, walking direction, *etc*.) for estimating head pose of freely moving persons.

Acknowledgments The authors gratefully acknowledge partial support from Singapore's Agency for Science, Technology and Research (A*STAR) under the Human Sixth Sense Programme (HSSP) grant, EIT ICT Labs SSP 12205 Activity TIK—The Interaction Toolkit, tasks T1320A-T1321A and the FP7 EU project DALI.

References

Benfold, B., & Reid, I. (2011). Unsupervised learning of a scene-specific coarse gaze estimator. In *International Conference on Computer Vision* (pp. 2344–2351).

Chen, C., & Odobez, J.-M. (2012). We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video. In *Computer Vision and Pattern Recognition* (pp. 1544–1551).

Dai, W., Yang, Q., Xue, G. R., & Yu, Y. (2007). Boosting for transfer learning. In *International Conference on Machine Learning* (pp. 193–200).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition* (pp. 886–893).

Daume, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of Association for Computational Linguistics* (pp. 256–263).

Doshi, A., & Trivedi, M. M. (2012). Head and eye gaze dynamics during visual attention shifts in complex environments. *Journal of Vision*, *12*(2), 1–16.

Duan, L., Tsang, I. W., & Xu, D. (2012). Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(3), 465–479.

Duan, L., Tsang, I. W., Xu, D., & Chua, T.-S. (2009). Domain adaptation from multiple sources via auxiliary classifiers. In *International Conference on Machine Learning* (pp. 289–296).

Farhadi, A., & Tabrizi, M. K. (2008). Learning to recognize activities from the wrong view point. In *European Conference on Computer Vision* (pp. 154–166).

Ferencz, A., Learned-Miller, E. G., & Malik, J. (2008). Learning to locate informative features for visual identification. *International Journal of Computer Vision*, *77*(1–3), 3–24.

HOSDB. (2006). Imagery library for intelligent detection systems (i-lids). In *IEEE Crime and Security*.

Jiang, J., & Zhai, C. (2007). Instance weighting for domain adaptation in nlp. In *Association of Computational Linguistics* (pp. 264–271).

Katzenmaier, M., Stiefelhagen, R., & Schultz, T. (2004). Identifying the addressee in human-human-robot interactions based on head pose and speech. In *International Conference on Multimodal Interfaces* (pp. 144–151).

Kulis, B., Saenko, K., & Darrell, T. (2011). What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition* (pp. 1785–1792).

Lanz, O. (2006). Approximate bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(9), 1436–1449.

Lanz, O., & Brunelli, R. (2008). Joint bayesian tracking of head location and pose from low-resolution video. In R. Stiefelhagen, R. Bowers, & J. G. Fiscus (Eds.), *Multimodal technologies for perception of humans*, Lecture Notes in Computer Science (Vol. 4625, pp. 287–296). Heidelberg: Springer.

Lepri, B., Subramanian, R., Kalimeri, K., Staiano, J., Pianesi, F., & Sebe, N. (2012). Connecting meeting behavior with extraversion–A systematic study. *IEEE Transactions on Affective Computing*, *3*(4), 443–455.

Lim, J. J., Salakhutdinov, R., & Torralba, A. (2011). Transfer learning by borrowing examples for multiclass object detection. In *Advances in Neural Information Processing Systems* (pp. 118–126).

Muñoz-Salinas, R., Yeguas-Bolivar, E., Saffiotti, A., & Carnicer, R. M. (2012). Multi-camera head pose estimation. *Machine Vision and Applications*, *23*(3), 479–490.

Murphy-Chutorian, E., & Trivedi, M. M. (2009). Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(4), 607–626.

Orozco, J., Gong, S., & Xiang, T. (2009). Head pose classification in crowded scenes. In *British Machine Vision Conference* (pp. 1–11).

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Pardoe, D., & Stone, P. (2010). Boosting for regression transfer. In *International Conference on Machine Learning* (pp. 863–870).

Rajagopal, A., Subramanian, R., Vieriu, R. L., Ricci, E., Lanz, O., Sebe, N., & Ramakrishnan, K. (2012). An adaptation framework for head pose estimation in dynamic multi-view scenarios. In *Asian Conference on Computer Vision* (pp. 652–666).

Ricci, E., & Odobez, J.-M. (2009). Learning large margin likelihoods for realtime head pose tracking. In *International Conference on Image Processing* (pp. 2593–2596).

Smith, K., Ba, S. O., Odobez, J.-M., & Gatica-Perez, D. (2008). Tracking the visual focus of attention for a varying number of wandering people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(7), 1212–1229.

Stiefelhagen, R., Bowers, R., & Fiscus, J. G. (2007). Multimodal Technologies for Perception of Humans. In *International evaluation workshops CLEAR 2007 and RT 2007*, Baltimore, MD, May 8–11, 2007, Revised Selected Papers (Vol. 4625). Heidelberg: Springer.

Subramanian, R., Staiano, J., Kalimeri, K., Sebe, N., & Pianesi, F. (2010). Putting the pieces together: Multimodal analysis of social attention in meetings. In *Acm Int'l Conference on Multimedia* (pp. 659–662).

Subramanian, R., Yan, Y., Staiano, J., Lanz, O., & Sebe, N. (2013). On the relationship between head pose, social attention and personality prediction for unstructured and dynamic group interactions. In *Acm Int'l Conference on Multimodal Interfaces*.

Tosato, D., Farenzena, M., Spera, M., Murino, V., & Cristani, M. (2010). Multi-class classification on riemannian manifolds for video surveillance. In *European Conference on Computer Vision* (pp. 378–391).

Voit, M., & Stiefelhagen, R. (2009). A system for probabilistic joint 3d head tracking and pose estimation in low-resolution, multi-view environments. In *Computer Vision Systems* (pp. 415–424).

Wang, X., Han, T. X., & Yan, S. (2009). An hog-lbp human detector with partial occlusion handling. In *International Conference on Computer Vision* (pp. 32–39).

Williams, C., Bonilla, E. V., & Chai, K. M. (2007). Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems* (pp. 153–160).

Yan, Y., Subramanian, R., Lanz, O., & Sebe, N. (2012). Active transfer learning for multi-view head-pose classification. In *Int'l Conference on Pattern Recognition* (pp. 1168–1171).

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 *Int'l Conference on Computer Vision*.

Yang, J., Yan, R., & Hauptmann, A. G. (2007). Cross-domain video concept detection using adaptive svms. In *Acm Int'l Conference on Multimedia* (pp. 188–197).

Yang, W., Wang, Y., & Mori, G. (2009). Human action recognition from a single clip per action. In *Int'l Workshop on Machine learning for Vision-Based Motion Analysis*.

Yang, W., Wang, Y., & Mori. G. (2010). Efficient human action detection using a transferable distance function. In *Asian Conference on Computer Vision* (pp. 417–426).

Zabulis, X., Sarmis, T., & Argyros, A. A. (2009). 3d head pose estimation from multiple distant views. In *British Machine Vision Conference* (pp. 1–12).

Zhang, Y., & Yeung, D.-Y. (2010). A convex formulation for learning task relationships in multi-task learning. In *Uncertainity in Artificial Intelligence* (pp. 733–742).

Zheng, J., Jiang, Z., Phillips, J., & Chellappa, R. (2012). Cross-view action recognition via a transferable dictionary pair. In *British Machine Vision Conference* (pp. 1–11).