No Matter Where You Are: Flexible Graph-guided Multi-task Learning for Multi-view Head Pose Classification under Target Motion

Yan Yan[1], Elisa Ricci[2,4], Ramanathan Subramanian[3], Oswald Lanz[4], Nicu Sebe[1]
[1]*Department of Information Engineering and Computer Science, University of Trento, Italy*
[2]*Department of Electrical and Information Engineering, University of Perugia, Italy*
[3]*Advanced Digital Sciences Center (ADSC), University of Illinois at Urbana-Champaign, Singapore*
[4]*Fondazione Bruno Kessler, Trento, Italy*
{yan,sebe}@disi.unitn.it, {elisa.ricci}@unipg.it, {Subramanian.R}@adsc.com.sg, {lanz}@fbk.eu

Abstract

We propose a novel Multi-Task Learning framework (FEGA-MTL) for classifying the head pose of a person who moves freely in an environment monitored by multiple, large field-of-view surveillance cameras. As the target (person) moves, distortions in facial appearance owing to camera perspective and scale severely impede performance of traditional head pose classification methods. FEGA-MTL operates on a dense uniform spatial grid and learns appearance relationships across partitions as well as partition-specific appearance variations for a given head pose to build region-specific classifiers. Guided by two graphs which a-priori *model appearance similarity among (i) grid partitions based on camera geometry and (ii) head pose classes, the learner efficiently clusters appearance-wise related grid partitions to derive the optimal partitioning. For pose classification, upon determining the target's position using a person tracker, the appropriate region-specific classifier is invoked. Experiments confirm that FEGA-MTL achieves state-of-the-art classification with few training data.*

1. Introduction

Head pose estimation and tracking is critical for surveillance and human-behavior understanding, and has been extensively studied for over a decade [15]. However, most existing approaches compute the head pose from high resolution images, where facial features are clearly visible. Estimating the head pose from large field-of-view surveillance cameras, where faces are typically captured at 50×50 or lower pixel resolution, has received importance only recently [5, 16, 19]. Computing the head pose under these conditions is difficult, as faces appear blurred and models employing detailed facial information are ineffective.

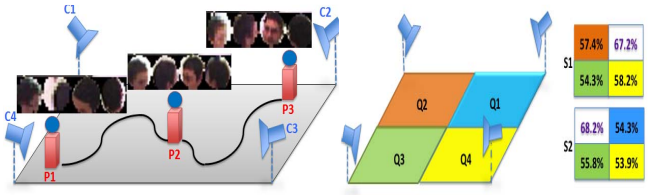

Figure 1. (left) Facial appearance change under target motion: for the same 3D head pose, automatically extracted face crops corresponding to camera C1-C4 are shown for target positions P1-P3. (right) Space division: S1, S2 denote classification accuracies when the training images come from the white quadrant (figure is best viewed in color).

Fewer still are head pose estimation methods that utilize information from multiple surveillance cameras. Employing a single camera view is insufficient for studying people's behavior in large environments and multi-view images have been exploited to achieve robust pose estimation [14, 22, 17, 20, 23]. However, methods such as [14, 20] estimate pose as a person rotates in place, but is not freely moving around in the environment. The broader goal of this work is to analyze behavior [13] from head pose cues in unstructured interactive settings (*e.g.* parties), where targets (persons) can move around freely. Therefore, in this paper we consider the problem of *multi-view head pose classification under target motion*.

Fig.1(left) illustrates the challenges involved in the considered scenario. The facial appearance of a target with identical 3D head pose but at different positions varies considerably due to perspective and scale. As the target moves, the face can appear larger/smaller and face parts can become occluded/visible due to the target's relative position with respect to the camera. We investigated the effect of appearance change on pose classification using the DPOSE dataset [17], which comprises synchronously recorded images of moving persons from four camera views, associated target positions and head pose annotations. Upon dividing the DPOSE space into four quadrants Q1-Q4, we trained an

SVM classifier with HOG [7] features extracted from the 4-view images corresponding to a particular quadrant. The SVM was then tested with images from each of the quadrants and the task was to assign head pose to one of eight classes, each denoting a quantized 45^o ($360^o/8$) head-pan. Fig.1(right) presents the results. Much lower accuracies were obtained when training and test images came from different quadrants, confirming the adverse impact of position-induced appearance changes on head pose classification.

To address this issue, we propose **FEGA-MTL**, a FlExible GrAph-guided Multi-Task Learning framework for multi-view head pose classification under target motion. Given a set of related tasks, MTL attempts to learn relationships among the tasks as well as task-specific differences. Upon dividing a physical space into a discrete number of planar regions (as in Fig.1), we seek to learn the pose-appearance relationship in each region. Analogous with the MTL problem, one can expect some similarity in facial appearance for a given head pose across the regions, and region-specific differences owing to perspective and scale.

FEGA-MTL seeks to simultaneously learn the relationship between facial appearance and head pose across all partitions of a dense uniform 2D spatial grid. Since the facial appearance is likely to be more similar for neighboring regions (as against spatially disjoint partitions), employing a single model to denote the inter-region appearance relationship can lead to *negative transfer*, arising when knowledge sharing has a negative impact on the performance of the learned model. Therefore, we devise a method where appearance-wise related grid clusters (which denote related tasks) are flexibly discovered, and the within-cluster appearance similarity is modeled via the MTL parameters.

Two graphs, which respectively define appearance similarity among (i) grid partitions for a particular head pose given camera geometry, and (ii) head pose classes, guide the learning process to output the optimal spatial partitioning comprising a number of grid clusters and an associated MTL classifier. During the classification stage, upon determining the position corresponding to a test instance using a person tracker, the corresponding region-specific classifier is invoked. Our approach is *flexible* owing to three main reasons: (1) It can work with arbitrary camera setups; (2) The learning algorithm can adaptively deal with multiple feature descriptors having differing discriminative power, and (3) Given the camera geometry and face appearance features, the optimal grid-cluster configuration is automatically discovered using our approach. Experiments confirm that FEGA-MTL outperforms competing head pose classification and MTL approaches.

To summarize, the paper's contributions are: (i) It represents one of the first works to explore multi-view head pose classification under target motion; (ii) To our knowledge, an MTL framework for head pose classification has not been proposed before; (iii) A novel graph-guided approach for simultaneously learning a set of classifiers and their relationships is proposed, and an efficient solver is devised; (iv) We seamlessly connect camera geometry (traditional computer vision) with machine learning for head pose classification through a novel graph modeling strategy; (v) FEGA-MTL is a general framework, potentially applicable to many computer vision and pattern recognition problems.

2. Related Work

Head Pose Classification from Low Resolution Faces. Head-pose classification from surveillance images has been investigated in a number of works [3, 5, 16, 19]. In [16], a Kullback-Leibler distance-based facial appearance descriptor is proposed for low resolution images. The array-of-covariances (ARCO) descriptor is introduced in [19], and is found to be effective for representing faces as it is robust to scale and illumination changes. In [3, 5], head pose estimation with weak or no supervision is achieved employing motion-based cues and constraints imposed by joint modeling of head and body pose. However, all these works address single view head pose classification.

Few works estimate head pose fusing information from multiple views [14, 17, 20, 23]. A particle filter is combined with a neural network for pan/tilt classification in [20]. A HOG-based confidence measure is also used to determine the relevant views. In [14], SVMs are employed to calculate a probability distribution for head pose in each view. The results are fused to produce a more precise estimate. Nevertheless, both these works attempt to determine head orientation as a person rotates in place and position-induced appearance variations are not considered.

A weighted distance approach for classifying pose under target motion is proposed in [17]. Upon dividing the space into four quadrants, max-margin distance learning is employed to learn a classifier per region– such a rigid space partitioning scheme will not optimally encode the pose-appearance relationship under motion, with arbitrary camera geometry. In [23], head pose under motion is determined by mapping the target's face texture onto a spherical head model, and subsequently locating the face in the unfolded spherical head image. However, many camera views are required to produce an accurate texture map– 9 cameras are used in [23]. In contrast, our approach is predominantly image-based, applicable even with few camera views.

Multi-task Learning. MTL methods aim to simultaneously learn classification/regression models for a set of related tasks. This typically leads to better models as compared to a learner that does not account for task relationships. Traditional MTL methods consider a single shared model, assuming that all the tasks are related [1, 8, 21]. However, when some of the tasks are unrelated, this may lead to negative transfer. Recently, more sophisticated

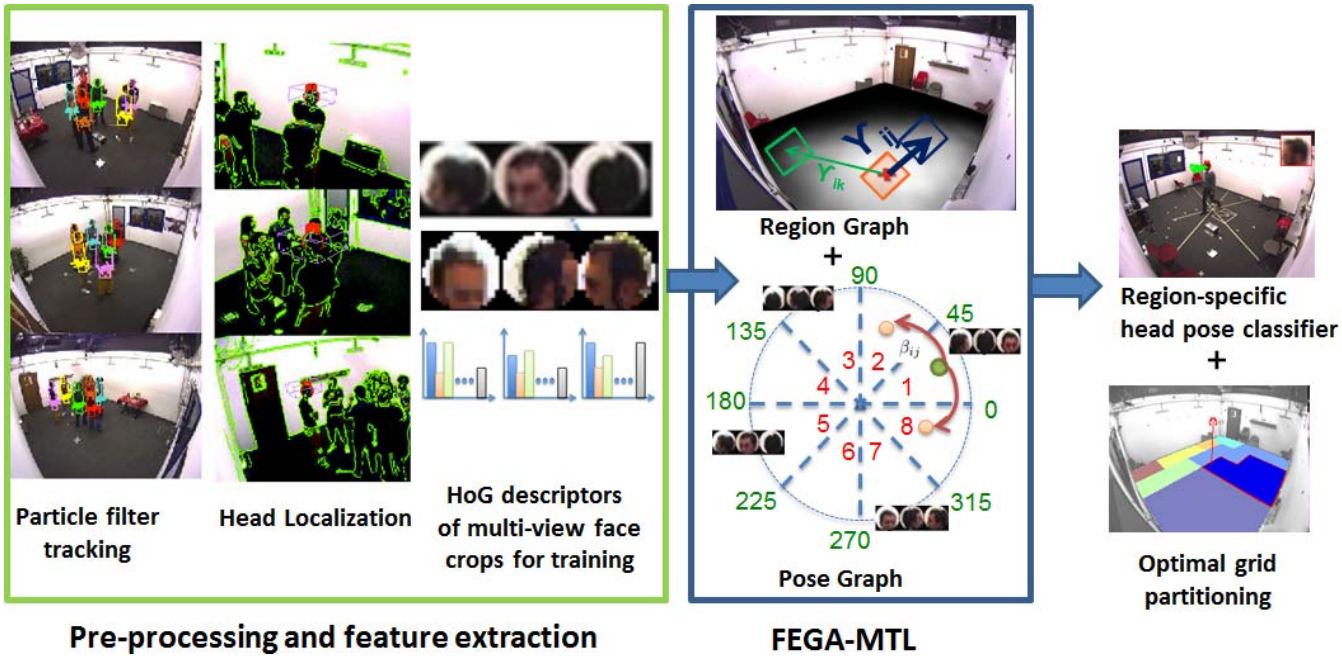

Figure 2. Overview of the proposed head pose classification framework assuming three camera views. The region graph and optimal partitioning are as seen from a fourth (camera-less) view. Figure is best viewed in color and under zoom.

approaches have been proposed to counter this problem. These methods assume some *a-priori* knowledge (*e.g.* in the form of a graph) defining task dependencies [6] or learn the task relationships simultaneously with task-specific parameters [11, 24, 10, 25, 9]. Among these, the work most similar to ours is [6]. Similar to [6], our algorithm adopts a graph to specify *a-priori* task dependencies. We also overcome the limitations of [6] as FEGA-MTL automatically discovers task relationships and refines the initial graph structure. For multi-view head pose estimation under motion, the graph structure is very useful as it reflects inter-region facial appearance similarity as derived from the camera geometry.

3. Multi-view Head Pose Classification

3.1. System Overview

Fig.2 presents an overview of our multi-view head pose classification system which consists of three phases: (1) preprocessing and extraction of multi-view face appearance descriptors, (2) learning of head pose-appearance relationships under motion with FEGA-MTL and (3) classification. As we deal with freely moving targets, in the preprocessing stage, a color-based particle filter tracker incorporating multi-view geometry information is employed to reliably localize the target's face and extract multi-view face crops. Also, the tracker allows for determining the target position corresponding to a test instance, so that the appropriate region-based pose classifier can be invoked. Features extracted from the multi-view face appearance images are fed to the FEGA-MTL module for learning region-specific classification parameters.

The learning process is guided by two graphs that respectively model appearance-based task dependencies among grid partitions and head pose classes– (a) the *region graph* quantifies the multi-view facial appearance distortion based on camera geometry, as the target moves from one grid partition to another, and (b) the *pose graph* posits that neighboring head pose classes tend to have more similar facial appearance. FEGA-MTL outputs the pose classification parameters for each grid partition, and the configuration of grid clusters so that the facial appearance for a given head pose is very similar in those partitions constituting a cluster– these grid clusters denote the learnt task relationships given the features and task dependencies. We will now describe each of these modules in detail.

3.2. Preprocessing

Tracking and Head Localization. A multi-view, color-based particle filter [12] is used to compute the 3D body centroid of moving targets. A 30×30×20 cm-sized dense 3D grid (with 1cm resolution) of hypothetic head locations is then placed around the estimated 3D head-position provided by the particle filter[1]. Assuming a spherical model of the head, a contour likelihood is computed for each grid point by projecting a 3D sphere onto each view using camera calibration information. The grid point with the highest likelihood sum is determined as the head location. The

[1]The grid size accounts for the tracker's variance and horizontal and vertical offsets of the head from the body centroid due to pan, tilt and roll.

tracking and head localization procedures are illustrated in Fig.2. The head is then cropped and resized to 20×20 pixels in each view.

Feature Extraction. Head crops from the different views are concatenated to generate the multi-view face crops as shown in Fig.2, and similar to previous works [3, 5], we employ HOG descriptors to effectively describe the face appearance for head pose classification. The multi-view face appearance image is divided into non-overlapping 4×4 patches, and a 9-bin histogram is used as the HOG descriptor for each image patch.

3.3. Space Partitioning and Graph Modeling

Region Graph Modeling. To apply FEGA-MTL, we initially divide the 2D ground space into a uniform grid with R partitions, as shown in Fig.3. We want to learn the pose-appearance relationship in each partition. The algorithm learns from a training set $\mathcal{T}_t = \{(\mathbf{x}_i^t, y_i^t) : i = 1, 2, \ldots, N_t\}$ for each region $t = 1, 2, \ldots, R$, where $\mathbf{x}_i^t \in I\!R^D$ denote D-dimensional feature vectors and $y_i^t \in \{1, 2, \ldots, C\}$ are the head pose labels (C = 8 classes in our setting). One of the graphs guiding the learning process specifies the similarity in appearance for a given head pose across regions based on camera geometry. If grid partitions form the graph nodes, we determine the edge set $\mathcal{E}_1$ and the associated edge weights γ_{mn} quantifying the appearance distortion between $\mathcal{T}_m$ and $\mathcal{T}_n$ due to positional change from region m to region n– these edge weights indicate whether knowledge sharing between regions m and n is beneficial or not.

As mentioned earlier, we model the target's head as a sphere. Let Z_k denote the sphere placed at the target's 3D head position p_k, and whose multi-view camera projection yields training image I_k in $\mathcal{T}_m$. Using camera calibration parameters, one can compute the correspondence between surface points in Z_k and pixels in I_k. Then, we move Z_k to position p_l corresponding to image I_l in $\mathcal{T}_n$, and determine how many surface points in Z_k are still visible in I_l. The appearance distortion over U camera views due to displacement v from p_k to p_l is defined as $\delta(Z_k, p_k \rightarrow p_l) = \sum_{u=1}^{U} \|v\| + \xi n_0$, where n_0 is the number of surface points in Z_k that are occluded after translation and ξ is a constant that penalizes such occlusion.

The appearance similarity between regions m and n is then computed based on a Gaussian model by considering distortion between all image-pairs associated to $\mathcal{T}_m, \mathcal{T}_n$ as:

$$\gamma_{mn} = e^{-\frac{\Omega}{N_m N_n \sigma^2}}$$

where $\Omega = \sum_{\forall I_k \in \mathcal{T}_m, I_l \in \mathcal{T}_n} [\delta(Z_k, p_k \rightarrow p_l) + \delta(Z_l, p_l \rightarrow p_k)]$, N_m and N_n are number of images in $\mathcal{T}_m$ and $\mathcal{T}_n$. $\sigma = 1$ and $\mathcal{E}_1$ is the set of edges for which $\gamma_{mn} \geq 0.1$.

Fig.3 depicts the appearance similarity maps for two different camera configurations when the head-sphere at p_k is moved around in space (the projection of p_k on the ground is denoted by the red 'X'). When p_k is close to the cameraless room corner in the 3-camera setup, a number of regions around p_k share a high appearance similarity, implying that pose-appearance relationship can be learnt jointly in these regions. However, the similarity measure decreases sharply as the target moves from p_k towards any of the three cameras, and tends to zero for the upper diagonal half of the room. Also, when a camera is introduced in the fourth room corner, appearance similarity holds only for a smaller portion of space around p_k as compared to the 3-camera case.

Pose Graph Modeling. A second graph guiding the learning process models the fact that facial appearances should be more similar for neighboring pose classes as compared to non-neighboring classes. For example, as shown in Fig.2, the facial appearance of exemplars from class 1 should be most similar to exemplars from class 2 and 8. Exploiting this information, a pose graph $\mathcal{E}_2$ is defined with associated edge weights β_{ij} = 1 if i and j correspond to neighboring pose classes c_i, c_j, and β_{ij} = 0 otherwise.

4. Flexible Graph-guided MTL

Given a training set $\mathcal{T}_t$, for each task (region) t, we define the matrix $\mathbf{X}_t = [\mathbf{x}_1^t, ..., \mathbf{x}_{N_t}^t]'$, $\mathbf{X}_t \in I\!R^{N_t \times D}$. If $N = \sum_{t=1}^{R} N_t$ denotes the total number of training samples, we also define $\mathbf{X} = [\mathbf{X}_1', \ldots, \mathbf{X}_R']'$, $\mathbf{X} \in I\!R^{N \times D}$ obtained concatenating the matrices $\mathbf{X}_t$ for all the R tasks. In this paper, the notation $(\cdot)'$ indicates the transpose operator. For each training sample, we construct a binary label indicator vector $\mathbf{y}_i^t \in I\!R^{RC}$ as $\mathbf{y}_i^t = [\underbrace{0, 0, ..., 0}_{Task\ 1}, \underbrace{0, 1, ..., 0}_{Task\ 2}, ..., \underbrace{0, 0, ..., 0}_{Task\ R}]$, *i.e.* the position of the non-zero element indicates the task and class membership of the corresponding training sample. A label matrix $\mathbf{Y} \in I\!R^{N \times RC}$ is then obtained concatenating the $\mathbf{y}_i^t$'s for all training samples.

For each region t and pose class c, we propose to learn the region-specific weight vectors for pose classification $\mathbf{w}_{t,c} = \mathbf{s}_{t,c} + \boldsymbol{\theta}_{t,c}$, $\mathbf{w}_{t,c}, \mathbf{s}_{t,c}, \boldsymbol{\theta}_{t,c} \in I\!R^D$. The $\mathbf{s}_{t,c}$ components model the appearance relationships among regions, while $\boldsymbol{\theta}_{t,c}$'s account for region-specific appearance variations. Defining the matrices $\mathbf{S}, \boldsymbol{\Theta} \in I\!R^{D \times RC}$, $\mathbf{S} = [\underbrace{\mathbf{s}_{1,1}, ..., \mathbf{s}_{1,C}}_{Task\ 1}, ..., \underbrace{\mathbf{s}_{R,1}, ..., \mathbf{s}_{R,C}}_{Task\ R}]$, $\boldsymbol{\Theta} = [\underbrace{\boldsymbol{\theta}_{1,1}, ..., \boldsymbol{\theta}_{1,C}}_{Task\ 1}, ..., \underbrace{\boldsymbol{\theta}_{R,1}, ..., \boldsymbol{\theta}_{R,C}}_{Task\ R}]$, we propose to solve the following optimization problem:

$$\min_{\mathbf{S}, \boldsymbol{\Theta}} \left\| (\mathbf{Y}'\mathbf{Y})^{-\frac{1}{2}} (\mathbf{Y} - \mathbf{X}(\mathbf{S} + \boldsymbol{\Theta})) \right\|_F^2 + \lambda_s \Omega_s(\mathbf{S}) + \lambda_\theta \Omega_\theta(\boldsymbol{\Theta}) \quad (1)$$

where $\| \cdot \|_F$ denotes the matrix Frobenius norm. The normalization factor $(\mathbf{Y}'\mathbf{Y})^{-1/2}$ compensates for different number of samples per task. The regularization term $\Omega_\theta(\boldsymbol{\Theta}) = \|\boldsymbol{\Theta}\|_F^2$ penalizes large deviation of $\mathbf{s}_{t,c}$ from $\mathbf{w}_{t,c}$, while $\Omega_s(\cdot)$ is defined as follows:

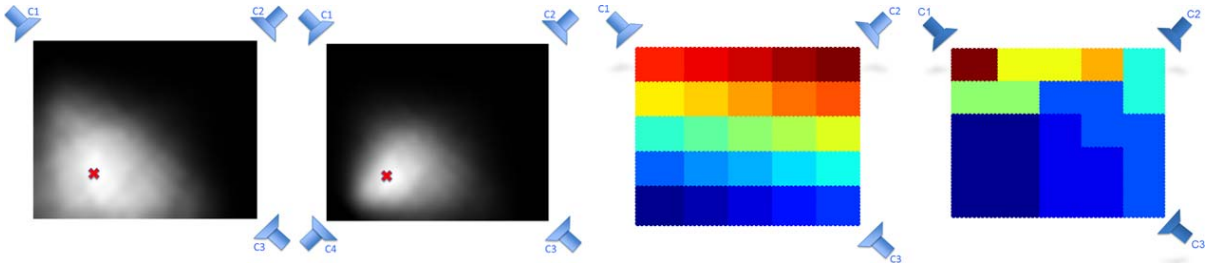

Figure 3. (from left to right) Appearance similarity map computed around a point with 3 camera views and 4 camera views, initial grid partitions and learned grid clusters for the 3-camera setup (figure best viewed in color).

$$\Omega_s(\mathbf{S}) = \|\mathbf{S}\|_F^2 + \lambda_1 \sum_{(i,j)\in\mathcal{E}_1} \gamma_{ij} \|\mathbf{s}_{t_i,c} - \mathbf{s}_{t_j,c}\|_1 + \lambda_2 \sum_{(i,j)\in\mathcal{E}_2} \beta_{ij} \|\mathbf{s}_{t,c_i} - \mathbf{s}_{t,c_j}\|_1$$

where γ_{ij}'s and β_{ij}'s are the appearance similarity-based weights of *region* graph edges $\mathcal{E}_1$ and *pose* graph edges $\mathcal{E}_2$ respectively as described in Sec 3.3. The term $\|\mathbf{S}\|_F^2$ regulates model complexity, while the ℓ_1 norm regularizer imposes the weights $\mathbf{s}_{t,c}$ of appearance-wise related regions and neighboring classes to be close together. In particular, region clusters are formed as $\lambda_1 \to \infty$. Importantly, this effect is *feature-specific*– cluster structure varies from feature to feature, and the clustering obtained for the more and less discriminant features can be very different. This is primarily why our method is *flexible*, and the model as well as the proposed optimization strategy benefit from this important effect.

To solve (1), we adopt the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [2]. FISTA solves optimization problems of the form $\min_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) + r(\boldsymbol{\mu})$, where $f(\boldsymbol{\mu})$ is convex and smooth, $r(\boldsymbol{\mu})$ is convex but non-smooth. Due to its simplicity and scalability, FISTA is a popular tool for solving many convex smooth/non-smooth problems. In each FISTA iteration, a proximal step is computed [2]:

$$\min_{\boldsymbol{\mu}} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_F^2 + \frac{2}{L_k} r(\boldsymbol{\mu}) \tag{2}$$

where $\hat{\boldsymbol{\mu}} = \tilde{\boldsymbol{\mu}}_k - \frac{1}{L_k}\nabla f(\tilde{\boldsymbol{\mu}}_k)$, $\tilde{\boldsymbol{\mu}}_k$ is the current estimate and L_k is a step-size determined by line search. To apply FISTA to our optimization problem, we define:

$$f(\mathbf{S},\boldsymbol{\Theta}) = \left\|(\mathbf{Y}'\mathbf{Y})^{-1/2}(\mathbf{Y} - \mathbf{X}(\mathbf{S}+\boldsymbol{\Theta}))\right\|_F^2$$

$$r(\mathbf{S},\boldsymbol{\Theta}) = \lambda_\theta\|\boldsymbol{\Theta}\|_F^2 + \lambda_s\|\mathbf{S}\|_F^2 + \lambda_s\lambda_1 \sum_{(i,j)\in\mathcal{E}_1} \gamma_{ij}\|\mathbf{s}_{t_i,c} - \mathbf{s}_{t_j,c}\|_1 + \lambda_s\lambda_2 \sum_{(i,j)\in\mathcal{E}_2} \beta_{ij}\|\mathbf{s}_{t,c_i} - \mathbf{s}_{t,c_j}\|_1$$

Incorporating the above definition in (2) followed by algebraic manipulation, the proximal step amounts to solving the following:

$$\min_{\mathbf{S},\boldsymbol{\Theta}} \left\|\mathbf{S} - \hat{\mathbf{S}}\right\|_F^2 + \left\|\boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}\right\|_F^2 + \hat{\lambda}_1 \sum_{(i,j)\in\mathcal{E}_1} \gamma_{ij}\|\mathbf{s}_{t_i,c} - \mathbf{s}_{t_j,c}\|_1 + \hat{\lambda}_2 \sum_{(i,j)\in\mathcal{E}_2} \beta_{ij}\|\mathbf{s}_{t,c_i} - \mathbf{s}_{t,c_j}\|_1 + \hat{\lambda}_s\|\mathbf{S}\|_F^2 + \hat{\lambda}_\theta\|\boldsymbol{\Theta}\|_F^2 \tag{3}$$

Algorithm 1 FEGA-MTL

INPUT: $\mathcal{T}_t, \forall t = 1,\ldots,R$, λ_s, λ_θ, λ_1, λ_2, $\mathbf{E}$
Initialize $\mathbf{S}_0$, $\boldsymbol{\Theta}_0$, $\alpha_0 = 1$.
OUTER LOOP:
  $\alpha_n = \frac{1}{2}(1 + \sqrt{1 + 4\alpha_{n-1}^2})$
  {*Update* $\boldsymbol{\Theta}$}
  $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}_n - 2\mathbf{X}'(\mathbf{X}\boldsymbol{\Theta}_n - \mathbf{Y})$
  $\boldsymbol{\Theta}_{n+\frac{1}{2}} = \frac{1}{1+\hat{\lambda}_\theta}\hat{\boldsymbol{\Theta}}$
  $\boldsymbol{\Theta}_{n+1} = (1 + \frac{\alpha_{n-1}-1}{\alpha_n})\boldsymbol{\Theta}_{n+\frac{1}{2}} - \frac{\alpha_{n-1}-1}{\alpha_n}\boldsymbol{\Theta}_n$
  {*Update* $\mathbf{S}$}
  $\hat{\mathbf{S}} = \mathbf{S}_n - 2\mathbf{X}'(\mathbf{X}\mathbf{S}_n - \mathbf{Y})$
  Update $\mathbf{S}_{n+\frac{1}{2}}$ with ADMM as follows:
  For each $d = 1 : D$
    Initialize $\mathbf{q}^{d,0}$, $\mathbf{a}^{d,0}$, $\mathbf{s}^{d,0}$
    Set $\mathbf{M} = \rho\mathbf{E}'\mathbf{E} + (2 + 2\hat{\lambda}_s)\mathbf{I}$
    Compute Cholesky factorization of matrix $\mathbf{M}$.
    INNER LOOP:
      {*Update* $\mathbf{s}$} Solve $\mathbf{M}\mathbf{s}^{d,k+1} = \mathbf{b}^k$
      {*Update* $\mathbf{q}$} $\mathbf{q}^{d,k+1} = ST_{\hat{\lambda}_1/\rho}(\mathbf{E}\mathbf{s}^{d,k+1} + \frac{1}{\rho}\mathbf{a}^{d,k})$
      {*Update* $\mathbf{a}$} $\mathbf{a}^{d,k+1} = \mathbf{a}^{d,k} + \rho(\mathbf{E}\mathbf{s}^{d,k+1} - \mathbf{q}^{d,k+1})$
    Until Convergence
  $\mathbf{S}_{n+1} = (1 + \frac{\alpha_{n-1}-1}{\alpha_n})\mathbf{S}_{n+\frac{1}{2}} - \frac{\alpha_{n-1}-1}{\alpha_n}\mathbf{S}_n$
Until Convergence
Output: $\mathbf{W} = \mathbf{S} + \boldsymbol{\Theta}$

where $\hat{\lambda}_s = 2\lambda_s/L_k$, $\hat{\lambda}_\theta = 2\lambda_\theta/L_k$ $\hat{\lambda}_1 = 2\lambda_1\lambda_s/L_k$ and $\hat{\lambda}_2 = 2\lambda_2\lambda_s/L_k$, $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta} - 2\mathbf{X}'(\mathbf{X}\boldsymbol{\Theta} - \mathbf{Y})$ and $\hat{\mathbf{S}} = \mathbf{S} - 2\mathbf{X}'(\mathbf{X}\mathbf{S} - \mathbf{Y})$. We solve (3) by considering $\mathbf{S}$, $\boldsymbol{\Theta}$ separately, using the procedure described in Algorithm 1. To optimize with respect to $\mathbf{S}$, we devise a novel approach using alternating direction method of multipliers (ADMM) [4]. In Algorithm 1, $ST_\lambda(x) = \text{sign}(x)\max(|x| - \lambda, 0)$ is a *soft-thresholding* operator and the matrix $\mathbf{E} = \begin{bmatrix} \hat{\lambda}_1\mathbf{E_1} \\ \frac{\hat{\lambda}_2}{\hat{\lambda}_1}\mathbf{E_2} \end{bmatrix}$ is defined considering the edge-vertex incident matrices $\mathbf{E}_1$ $_{e=(i,j),h} = \begin{cases} \gamma_{ij}, & i = h \\ -\gamma_{ij}, & j = h \\ 0, & otherwise \end{cases}$, $\mathbf{E}_1 \in I\!R^{|\mathcal{E}_1|\times RC}$, and $\mathbf{E}_2$ $_{e=(i,j),h} = \begin{cases} \beta_{ij}, & i = h \\ -\beta_{ij}, & j = h \\ 0, & otherwise \end{cases}$, $\mathbf{E}_2 \in I\!R^{|\mathcal{E}_2|\times RC}$.

Regarding the computational complexity of Algorithm

1, the main steps in the outer loop are: the update of $\mathbf{\Theta}$ which takes $\mathcal{O}(DR)$ time, the gradient computation taking $\mathcal{O}(N_t DR)$ time and the update of $\mathbf{S}$. The last step is the most computationally expensive as it requires a Cholesky matrix factorization ($\mathcal{O}(R^3)$) for each dimension $d = 1, \ldots, D$. However, the Cholesky factorization is performed only in the outer loop. In the inner loop, each iteration involves solving one linear system ($\mathcal{O}(R^2)$) and a soft-thresholding operation ($\mathcal{O}(|\mathcal{E}_1| + |\mathcal{E}_2|)$).

After the learning phase, the computed weights matrix $\mathbf{W} = \mathbf{S} + \mathbf{\Theta}$ is used for classification. While testing, upon determining the region t associated to a test sample $\mathbf{x}_{test}$ using the person tracker, the corresponding $\mathbf{w}_{t,c}$'s are used to compute the head pose label as $\arg\max_{c=1,\ldots,C} \mathbf{w}'_{t,c}\mathbf{x}_{test}$.

5. Experimental Results

In this section, we compare head pose classification results achieved with FEGA-MTL against (i) state-of-the-art head pose estimation methods and (ii) other MTL approaches. We perform our experiments on the DPOSE dataset [17]. To our knowledge, there are no other databases for benchmarking multi-view head pose classification performance under target motion. The CLEAR [18] and Uco-Head [14] databases are recorded with targets rotating in-place, while the dataset proposed in [23] does not include ground-truth head pose measurements for moving targets. DPOSE comprises over 50000 4-view synchronized images recorded for 16 moving targets, with associated positional and head pose measurements (target positions are computed using the person tracker [12]).

As mentioned earlier, the larger goal of this work is to detect interactions in informal gatherings such as *parties*, where we mainly focus on classifying the head-pan into one of 8 classes (each denoting a 45° pan range). Since faces are captured at low-resolution by distant, large field-of-view cameras, this task is quite challenging and the state-of-the-art can achieve only about 79% accuracy on the 4-view face images (Table 1). We divide DPOSE into mutually exclusive training/validation/test sets. For all methods, regularization parameters are tuned using the validation set, considering values in the interval $[2^{-3}, 2^{-2}, \ldots, 2^3]$. We consider an initial, uniformly spaced grid with $R = 25$ regions as shown in Fig.3. Our results denote mean classification accuracies obtained from five independent trials, where a randomly chosen training set is employed in each trial.

Table 1 presents results comparing FEGA-MTL with competing head pose classification methods. We gradually increase the training set size from 5 to 30 samples/class/region, while the test set comprises images from all regions. As baselines, we consider the recent multi-view approach which probabilistically fuses the output of multiple SVMs [14] and the state-of-the-art ARCO classifier [19] which is shown to be powerful at low resolution (we feed in the 4-view image features to ARCO in order to extend it to the multi-view setup). As shown in the table, both these methods perform poorly with respect to the proposed approach, as they are not designed to account for facial distortions due to scale/perspective changes.

A better strategy in such cases is to compensate for position-induced appearance distortions in some way [17, 23]. The texture-mapping approach presented in [23] is shown to be accurate, but many cameras are required for effective texture mapping. Instead, we attempted the warping method proposed in [17], which despite its simplicity is shown to effectively work with few low-resolution views. We implemented a radial basis SVM to determine head pose from the warped 4-view images. Warping is greatly beneficial in the considered scenario as the Single SVM+Warping method significantly outperforms Single SVM.

It is pertinent to point out two differences between our approach and [17]– [17] proposes a pre-defined division of space (the room is divided into 4 quadrants) which is not necessarily optimal for describing the pose-appearance relationship under arbitrary camera geometry. Secondly, task relationships are not considered in [17], and an independent classifier is used for each quadrant. In contrast, FEGA-MTL discovers the optimal configuration of grid clusters that best describes the pose-appearance relationship given camera geometry. Considering task relationships enables FEGA-MTL to achieve higher classification accuracy than a single global classifier (Single SVM), Single SVM+Warping and separate region-specific classifiers that do not consider inter-region appearance relationships (Multiple Region-specific SVMs).

Table 1 also presents accuracies obtained with ℓ_{21} MTL [1], which assumes all tasks share a common component. As discussed before, negative transfer adversely affects performance of ℓ_{21} MTL, while FEGA-MTL achieves higher accuracy upon flexibly discovering related tasks. We also repeated the experiments employing only two of the four camera views for head pose classification, and while obtained accuracies are expectedly lower in this case, the accuracy trends are still consistent with the 4-view scenario.

Table 2 compares classification performance of various MTL methods. The advantage of employing MTL for head pose classification under target motion is obvious since all MTL approaches greatly outperform single SVM. Moreover, having a flexible learning algorithm which is able to infer appearance relationships among regions provides some advantages in terms of classification accuracy. This is confirmed by the fact that in all situations (varying training set sizes and number of camera views) FTC MTL [24], Clustered MTL [25] and FEGA-MTL achieve superior performance. FEGA-MTL, which independently considers features and employs graphs to explicitly model region and head pose-based appearance relationships, achieves the best

Table 1. DPOSE dataset: Head pose classification accuracy. Comparison with state-of-the-art head pose estimation methods.

	4-view				**2-view**			
	Training Set Size/Class/Region				Training Set Size/Class/Region			
	5	10	20	30	5	10	20	30
Single SVM	0.495	0.564	0.65	0.70	0.441	0.486	0.559	0.602
Multiple Region-specific SVMs	0.523	0.571	0.664	0.699	0.446	0.51	0.58	0.618
ℓ_{21} MTL[1]	0.589	0.696	0.779	0.795	0.525	0.642	0.724	0.758
Multi-view SVM [14]	0.544	0.573	0.682	0.713	0.447	0.486	0.565	0.672
ARCO [19]	0.603	0.70	0.761	0.784	0.529	0.64	0.695	0.739
Single SVM+Warping [17]	0.563	0.644	0.725	0.752	0.466	0.575	0.653	0.687
FEGA-MTL	**0.660**	**0.759**	**0.822**	**0.861**	**0.602**	**0.711**	**0.759**	**0.799**

Table 2. DPOSE dataset: Head pose classification accuracy. Comparison with MTL approaches.

	5 training samples/class/region			**10** training samples/class/region		
	2-view	3-view	4-view	2-view	3-view	4-view
Single SVM	0.441	0.494	0.523	0.486	0.549	0.564
ℓ_{21} MTL [1]	0.525	0.567	0.589	0.642	0.675	0.696
Flexible Task Clusters MTL [24]	0.555	0.598	0.621	0.65	0.681	0.715
Dirty model MTL [10]	0.546	0.585	0.603	0.655	0.686	0.696
Clustered MTL [25]	0.540	0.590	0.619	0.639	0.682	0.711
Robust MTL [9]	0.550	0.580	0.581	0.655	0.689	0.705
FEGA-MTL (region graph only, $\lambda_2 = 0$)	0.581	0.623	0.643	0.677	0.718	0.733
FEGA-MTL (region graph + pose graph)	**0.602**	**0.643**	**0.660**	**0.711**	**0.748**	**0.759**

performance. The usefulness of modeling both region and pose-based task dependencies through FEGA-MTL is evident on observing the results in Table 2. Using the region graph alone is beneficial as such, while employing the region and pose graphs in conjunction produces the best classification performance.

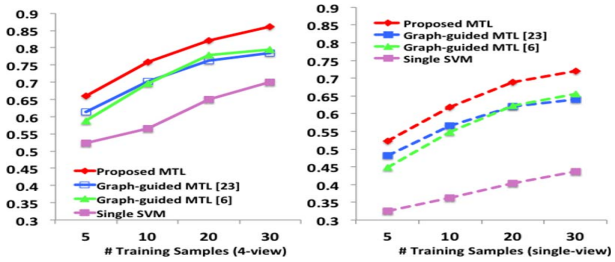

Figure 4. Comparison of graph-guided MTL methods: classification accuracies for (left) 4-views and (right) a single view.

Fig.3 shows the initial spatial grid and the optimal spatial partitioning learned for a three-camera system with 5 training images/class/region. Clustered regions correspond to identical columns of the task similarity matrix $\mathbf{S}$, *i.e.* two regions t_i and t_j merge if $\mathbf{s}_{t_i,c} = \mathbf{s}_{t_j,c}\ \forall c$. Constrained by the appearance similarity graph weights, spatially adjacent regions tend to cluster together. While regions closer to the camera-less room corner tend to form large clusters, smaller clusters are observed as one moves closer to the cameras owing to larger facial appearance distortions caused by perspective and scale changes. Apart from the region and pose-based appearance similarity graph weights, facial appearance features also influence the clustering of related regions, and therefore, the computed optimal partitioning.

To further demonstrate the advantages of FEGA-MTL, we compare it with the other graph-guided MTL methods [6, 26]. Fig.4 shows that higher accuracy is obtained with our approach for different training set sizes. A main difference between FEGA-MTL and these methods [6, 26] is that they do not decompose $\mathbf{w}_{t,c}$ as $\mathbf{s}_{t,c} + \boldsymbol{\theta}_{t,c}$, and due to the non-consideration of task-specific components $\boldsymbol{\theta}_{t,c}$, they have less flexibility. Moreover, in [26] (due to the use of ℓ_2 norm) and [6] (due to smoothing) task-clustering is encouraged but not *enforced*, *i.e.* the $\mathbf{w}_{t,c}$'s corresponding to a cluster are similar but not identical.

Also, it is worth noting that FEGA-MTL can also be used in a single-view setting. However, the use of multiple views is greatly advantageous. Fig.4 presents the accuracies obtained with 4-view features against single-view features (mean of the accuracies obtained with each of the four views is considered here). Expectedly, higher classification accuracy is obtained with the four-view features. The performance gain achieved using FEGA-MTL over an SVM modeling pose-appearance relationship over the entire space is evident, for both single and four-view cases.

Finally, Fig.5 shows some qualitative results obtained with FEGA-MTL for single and multiple targets tracked real-time using [12]. With multiple targets, identical colors are used to denote the pose direction frustum and face crop rectangle for each target. This scenario is quite challenging, as six targets are interacting naturally and freely moving around in the room.

6. Conclusions

We propose a novel graph-guided FEGA-MTL framework for classifying head pose of moving targets from multiple camera views. Starting from a dense 2D spatial grid, two graphs which respectively model appearance similarity among grid partitions and head pose classes guide the learner to output region-specific pose classifiers and the optimal space partitioning. Experiments demonstrate the superiority of FEGA-MTL over competing methods.

Acknowledgements: This work was partially supported by EIT ICT Labs SSP 12205 Activity TIK - The Interaction Toolkit, tasks T1320A-T1321A and A*STAR Singapore under the Human Sixth Sense Program (HSSP) grant.

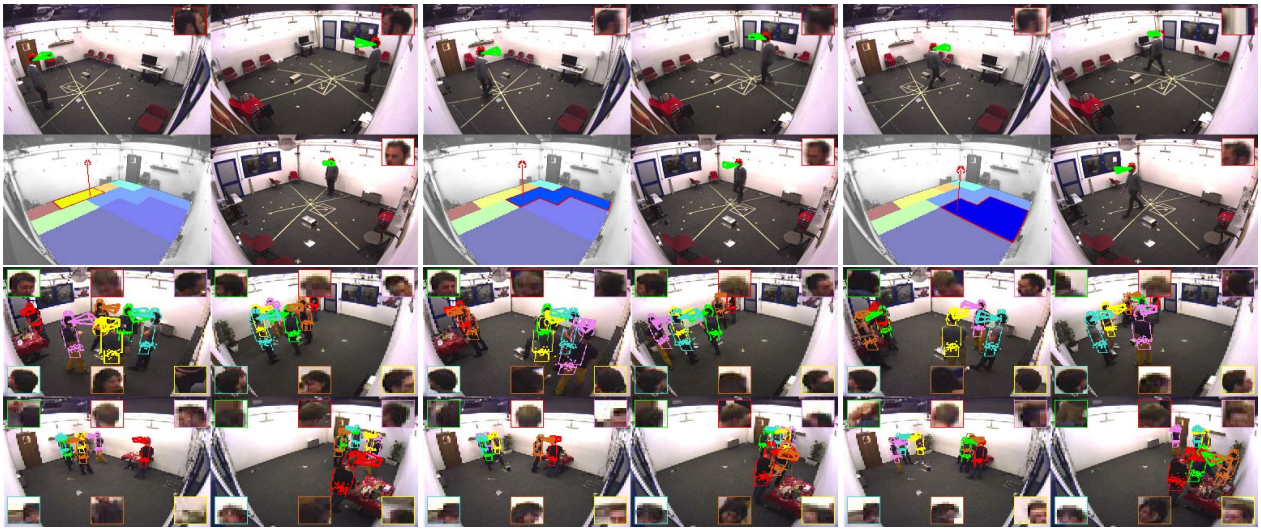

Figure 5. (Top) Head pose classification results for a target moving freely within a 3-camera setup are shown two-by-two. The learned clusters, as seen from a fourth view, are shown on the bottom-left inset. Cluster corresponding to the target position (denoted using a stick model) is highlighted. (Bottom) Pose classification results for a party video involving multiple mobile targets (best viewed under zoom.)

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, 2007. 2, 6, 7

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 5

[3] B. Benfold and I. Reid. Unsupervised learning of a scene-specific coarse gaze estimator. In *ICCV*, 2011. 2, 4

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011. 5

[5] C. Chen and J.-M. Odobez. We are not contortionists: coupled adaptive learning for head and body orientation estimation in surveillance video. In *CVPR*, 2012. 1, 2, 4

[6] X. Chen, Q. Lin, S. Kim, J. Carbonell, and E. Xing. Smoothing proximal gradient method for general structured sparse learning. In *UAI*, 2011. 3, 7

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[8] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *SIGKDD*, 2004. 2

[9] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. In *SIGKDD*, 2012. 3, 7

[10] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. In *NIPS*, 2010. 3, 7

[11] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011. 3

[12] O. Lanz. Approximate bayesian multibody tracking. *IEEE Trans. on PAMI*, 28:1436–1449, 2006. 3, 6, 7

[13] B. Lepri, R. Subramanian, K. Kalimeri, J. Staiano, F. Pianesi, and N. Sebe. Connecting meeting behavior with extraversion: A systematic study. *IEEE Trans. on Affective Computing*, 3:443–455, 2012. 1

[14] R. Muñoz-Salinas, E. Yeguas-Bolivar, A. Saffiotti, and R. M. Carnicer. Multi-camera head pose estimation. *Mach. Vis. Appl.*, 23(3):479–490, 2012. 1, 2, 6, 7

[15] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. on PAMI*, 31:607–626, 2009. 1

[16] J. Orozco, S. Gong, and T. Xiang. Head pose classification in crowded scenes. In *BMVC*, 2009. 1, 2

[17] A. Rajagopal, R. Subramanian, R. Vieriu, E. Ricci, O. Lanz, N. Sebe, and K. Ramakrishnan. An adaptation framework for head pose estimation in dynamic multi-view scenarios. In *ACCV*, 2012. 1, 2, 6, 7

[18] R. Stiefelhagen, R. Bowers, and J. G. Fiscus. Multimodal technologies for perception of humans, CLEAR. 2007. 6

[19] D. Tosato, M. Farenzena, M. Cristani, M. Spera, and V. Murino. Multi-class classification on riemannian manifolds for video surveillance. In *ECCV*, 2010. 1, 2, 6, 7

[20] M. Voit and R. Stiefelhagen. A system for probabilistic joint 3d head tracking and pose estimation in low-resolution, multi-view environments. In *Computer Vision Systems*, pages 415–424, 2009. 1, 2

[21] Y. Yan, G. Liu, E. Ricci, and N. Sebe. Multi-task linear discriminant analysis for multi-view action recognition. In *ICIP*, 2013. 2

[22] Y. Yan, R. Subramanian, O. Lanz, and N. Sebe. Active transfer learning for multi-view head-pose classification. In *ICPR*, 2012. 1

[23] X. Zabulis, T. Sarmis, and A. Argyros. 3d headpose estimation from multiple distant views. In *BMVC*, 2009. 1, 2, 6

[24] L. W. Zhong and J. T. Kwok. Convex multitask learning with flexible task clusters. In *ICML*, 2012. 3, 6, 7

[25] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS*, 2011. 3, 6, 7

[26] J. Zhou, J. Chen, and J. Ye. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University, 2011. 7