Chapter 10

Recognizing social relationships from an egocentric vision perspective

Stefano Alletto*, Marcella Cornia*, Lorenzo Baraldi*, Giuseppe Serra†, Rita Cucchiara*

*University of Modena and Reggio Emilia, Department of Engineering "Enzo Ferrari", Modena, Italy †University of Udine, Department of Mathematics, Computer Science and Physics, Udine, Italy

CONTENTS

10.1 INTRODUCTION

Humans are inherently good at understanding and categorizing social formations. A clear signal of this cue is that we respond to different social situations with different behaviors: while we accept to stand in close proximity to strangers when we attend some kind of public event, we would feel uncomfortable in having people we do not know close to us when we take a coffee. This processing happens so naturally in our brains that we

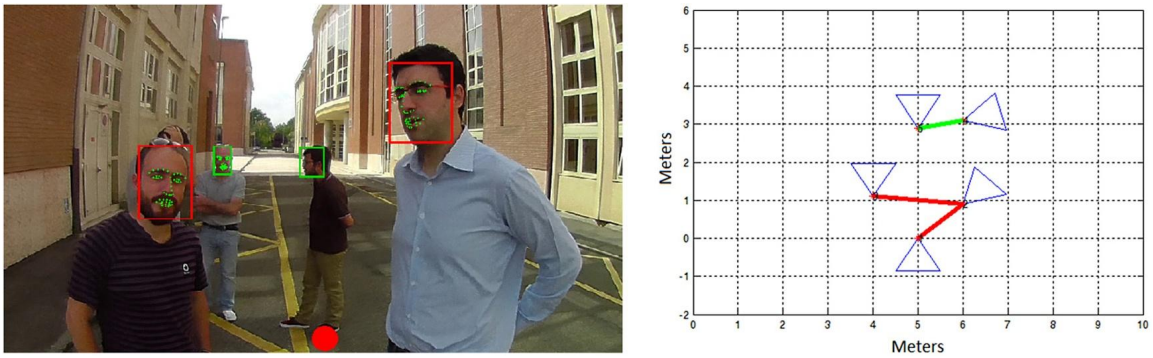
Multimodal Behavior Analysis in the Wild. https://doi.org/10.1016/B978-0-12-814601-9.00015-8

rarely stop wondering who is interacting with whom or which people form a group and which do not. Nevertheless, understanding social relationships and social groups is a complex task which can hardly be transferred to a fully automatic system.

Initial work has started to address the task of social interaction analysis from the videosurveillance perspective [24,33]. Fixed cameras, however, lack the ability to immerse in the social environment, effectively losing an extremely significant portion of the information about what is happening. Wearable cameras, on the contrary, put the research on this matter in a new and unique perspective. A video taken from the egocentric perspective provides a more meaningful insight in the social interaction, given that the recording is performed by a member of the group itself with a clear view of the social formation.

This privileged perspective lets researchers use wearable cameras for acquiring and processing the same visual stimuli that humans acquire and process. In this regard, first-person vision (or egocentric vision) assumes the broader meaning of understanding what a person sees calling for similar learning, perception and reasoning paradigms of humans. While this approach carries exceptional benefits, it also features several problems: the camera continuously follows the wearer's movements, resulting in severe camera motion, steep lighting transitions, background clutter and severe occlusions. These situations are required to be properly tackled in order to process the video automatically and extract higher level information.

There are significant cues which can be captured by an egocentric camera and which can help the automatic understanding of the social formation the user is involved in. First, when we are interacting with each other we naturally tend to place ourselves in determined positions to avoid occlusions in our group, stand close to the ones we interact with and orientate our head so as to place the focus on the subjects of our interest. Moreover, when engaged in a conversation we naturally tend to look at the people we are interacting with, and to ignore others, so eye fixations are an important cue to determine the strength of a social relationship between people. Distances between individuals and mutual orientations also assume clear significance and must be interpreted according to the situation. *F-formation* theory [18] describes patterns that humans naturally tend to create when interacting with each other and can be used to understand whether an ensemble of people forms a group or not, based on the mutual distances and orientations of the subjects in the scene. *F-formations* have recently been successfully applied in videosurveillance, with fixed cameras, in studies aimed at social interaction analysis showing great promise [9,14].



■ **FIGURE 10.1** An example of our method output. In the left image: different colors in bounding box indicate their belonging to different groups. The red dot represents the first person wearing the camera. In the right image: the bird's eye view model where each triangle represents a person and links among them represent the groups.

Following these cues, we adopt distance and orientation information and use them to build a pairwise feature vector capable of describing how two people relate. Since orientations and distances differ as regards importance and meaning in different situations, we use a supervised correlation clustering framework to learn about social groups. Once social groups are detected, we estimate the gaze of the camera wearer by using a saliency prediction approach. This lets us recover important information, which the camera cannot record, and predict the importance of the social relation between the user and the people involved in his social group. While head orientations and distances can be inferred for the people the user can see (e.g. *the others*), saliency estimates a component of the behavior of the wearer himself. An example of the output of our method is shown in Fig. 10.1.

In the rest of this chapter, we will discuss the following issues:

- The definition of a novel head pose estimation approach which can cope with the challenges of the egocentric vision scenario: using a combination of facial landmarks and shape descriptors, our head pose method is robust to steep poses, low resolutions and background clutter.
- The formulation of a 3D ego-vision people localization method capable of estimating the position of a person without relying on calibration. Camera calibration is a process that cannot be automatically performed on different devices and would cause a loss in generality for our method. We use instead random regression forests that employ facial landmarks and the head bounding box as features, resulting in a robust pose-independent distance estimation of the head.
- Modeling of a supervised correlation clustering algorithm using structural SVM to learn how to weight each component of the feature vector depending on the social situation it is applied to. This is due to the fact that humans perform differently in different social situations and the way groups are formed can greatly differ.

- The estimation of the degree of social interaction between the camera wearer and the other people involved in his social group, through the definition of a social strength score derived from a supervised saliency prediction model.

The proposed method is evaluated on publicly available datasets, and by comparing it with several recent algorithms. Each component of the framework is extensively discussed. While experimental results highlight some open problems, they show a new way for computer vision to deal with the complexity of unconstrained scenarios such as egocentric vision and human social interactions.

10.2 RELATED WORK

In this section, we review the literature related to head pose estimation and social interactions with particular attention to egocentric vision approaches.

10.2.1 Head pose estimation

The problem of estimating the head pose has been widely studied in computer vision. Existing methods can be roughly divided into two main categories, regardless of whether their aim is to assess the head pose on still images or video sequences.

Considering the most important solutions for head pose estimation in still images, We et al. [31] proposed a two-level classification framework based on Gabor wavelets in which the first level has the objective of deriving a good estimate of the pose within some uncertainty, while the second level aims at minimizing this uncertainty by analyzing finer structural details captured by bunch graphs. Ma et al. [21] presented a multiview face representation based on local Gabor binary patterns extracted on different sub-regions of the images. Despite these methods performing well on different publicly available datasets, they have significant performance losses when applied to less constrained environments, as egocentric vision contexts.

In [36], an unified model for face detection, pose estimation and landmark localization for "in the wild" images is presented. In particular, the proposed model is based on mixtures of trees with a shared pool of parts in which every facial landmark is modeled as a part and global mixtures are used to capture topological changes due to varying the viewpoint. A different approach is introduced by Li et al. [20] who developed a central profile-based 3D face pose estimation method. The central profile is a 3D curve that divides the face and has the characteristic of having its points lying on the symmetry plane. By relying on the Hough transform to determine the sym-

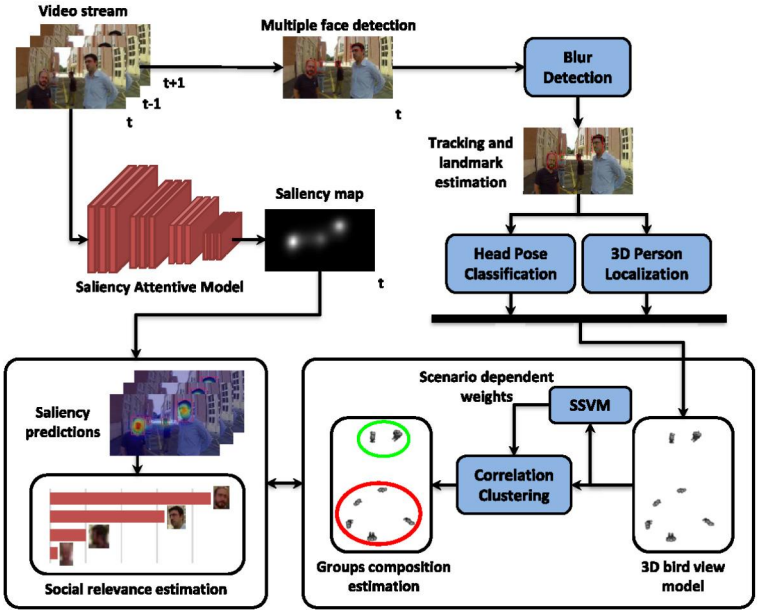
metry plane, Li et al. estimate the head pose using the normal vectors of the central profile which are parallel to the symmetry plane. For a comprehensive study summarizing about 90 of the most innovative head pose estimation methods, we refer to the survey presented in [22].

Moving to the video domain, several existing approaches exploit 3D information to estimate the head pose [23,25]. [23] who However, this kind of information can be hardly employed in an egocentric vision environment in which wearable devices, being aimed at more general purpose users and being on a mid-low price tier, usually lack the ability to capture 3D feature points. Moreover, due to the unpredictable motion of both camera and object, a robust 3D model is often hard to recover from multiple images. Instead of using a 3D model, Huang et al. [12] developed a computational framework capable of performing detection, tracking and pose estimation of faces captured by video arrays. To estimate the face orientation, they compared through extensive experiments a Kalman filtering based tracker and multistate continuous density hidden Markov models. Orozco et al. [26] presented a head pose estimation technique based on mean appearance templates and multiclass SVM, and effectively applied to low-resolution video frames representing crowded public spaces under poor light conditions.

10.2.2 Social interactions

There is a growing interest in understanding social interactions and human behavior of individuals present in video frames using computer vision techniques. However, the majority of these methods are based on the video surveillance setting [24,33,35], which presents significant differences with respect to the first-person perspective. One of the first attempts of studying social interactions in the egocentric vision domain is that presented by Fathi et al. [11] who aim at recognizing five different social situations (monologue, dialog, discussion, walking dialog, walking discussion). By using day-long videos recorded from an egocentric perspective in an amusement park, they extract three categories of features: location of faces around the first person, patterns of attention and roles take by individuals and patterns of first-person head movement. These features are then used in a framework that explores the temporal dependency over time to detect the types of social interactions.

Yonetani et al. [34] presented a method to understand the dynamics of social interactions between two people by recognizing their actions and reactions using a head-mounted camera. In particular, to recognize micro-level actions and reactions, such as slight shifts in attention, subtle nodding, or small hand actions, they proposed to use paired egocentric videos recorded by two interacting people. In [1], instead, a new pipeline for automatic social



■ **FIGURE 10.2** Schematization of the proposed approach.

pattern characterization of a wearable photo-camera user is presented. The proposed pipeline first of all studies a wider set of features for social interaction detection and second categorizes the detected social interactions into two broad categories of meetings (i.e. formal and informal). Even though all these methods provide interesting insights for understanding social interactions in the egocentric vision domain, none of them takes into account the group dynamics and the social relations within the group as presented in this work.

10.3 UNDERSTANDING PEOPLE INTERACTIONS

To deal with the complexity of understanding people interactions and detecting groups in real and unconstrained egocentric vision scenarios, our method relies on several components (see Fig. 10.2). We start with an initial face detection and then track the head to follow the subjects between frames. Head pose and 3D people locations are estimated to build a "bird view" model that is the input of the supervised correlation clustering in order to detect groups in different contexts based on the estimation of pairwise relations of their members. To further analyze the social dynamics, we estimate the social relevance of each subject by means of a saliency prediction model based on deep neural networks.

10.3.1 Face detection and tracking

A typical egocentric vision feature is a steep motion of the camera wearer. This can happen, for example, when a camera wearer is looking around for something. In this case, this person has not focused his attention on some point of interest and hence those frames are likely to be not interesting. Therefore, a first step to the face tracking procedure is to recognize whether tracking itself should be performed or not.

From a technical point of view, the steep motion can significantly increase the blur effect in the video sequence. If not addressed properly, this situation can degrade the tracking to the point that it may not be possible to resume it when the attention of the subject stabilizes again. To deal with this issue, at each frame we compute the amount of blurriness and decide whether to proceed with the tracking or to skip it. The idea of our approach is to evaluate the gradient intensity in the frame and to learn a threshold that can recognize a fast head movement from the normal blur caused by motion of objects, people or background. We define a simple blur function which recognizes the blur degree in a frame F, according to a threshold θ_B:

$$Blur(F, \theta_B) = \sum_F \sqrt{\nabla S_x^2(F) + \nabla S_y^2(F)}, \tag{10.1}$$

where $\nabla S_x^2(F)$ and $\nabla S_y^2(F)$ are the x and y components of Sobel's gradient in the frame and θ_B is the threshold under which the frame is discarded due to excessive motion blurriness, a parameter which can be learned by computing the average amount of gradient in a sequence.

This preprocessing step, which can be performed in real-time, effectively allows us to discard frames that could lead the tracker to adapt its model to a situation where gradient features cannot be reliably computed. To robustly track people in our scenario we adopt the tracker TLD [17] as it is able to deal with fast camera motion and occlusions which often occur between members of different groups.

10.3.2 Head pose estimation

To estimate an accurate head pose our approach is based on two different techniques: facial landmarks and shape-based head pose estimation.

With the facial landmarks approach, head pose can be accurately estimated if the face resolution is high enough and the yaw, pitch and roll angles of the head are not excessively steep. However, when these conditions are not satisfied and the first strategy fails, our method relies on shape-based head pose

estimation and uses HOG features and a classification framework composed of SVM followed by HMM.

The facial landmark estimator is the first component of our solution: if this can be computed, the head pose can be reliably inferred and no further processing is needed.

To estimate facial landmarks, we employ the method proposed by Smith et al. [32]. We fix the number of landmarks at 49 as it is the minimum number of points for a semantic face description [28]. To obtain the head pose we perform a face alignment procedure by applying the supervised gradient descent method, which minimizes the following function over $\Delta\mathbf{x}$:

$$f(\mathbf{x}_0 + \Delta\mathbf{x}) = \|\mathbf{h}(\boldsymbol{\lambda}(\mathbf{x}_0 + \Delta\mathbf{x})) - \phi_*\|_2^2, \tag{10.2}$$

where $\mathbf{x}_0$ is the initial configuration of landmarks, $\boldsymbol{\lambda}(\mathbf{x})$ is the function that indexes the N landmarks in the image and $\mathbf{h}$ is a non-linear feature extraction function; in this case the SIFT operator. $\phi_* = \mathbf{h}(\boldsymbol{\lambda}(\mathbf{x}_*))$ represents the SIFT descriptors computed over manually annotated landmarks in the image. Finally, the obtained pose is quantized over five classes, representing the intervals $[-90, -60)$, $[-60, -30)$, $[-30, 30]$, $(30, 60]$ and $(60, 90]$.

If the landmark estimator fails, we combine it with a second component based on the shape of the subject's head. Before computing the head descriptor, which will be used in the pose classification step, a preprocess step is required in an unconstrained scenario as egocentric vision: background removal inside the bounding boxes of the tracked faces. We use an adaptation of the segmentation algorithm GrabCut [27], which minimizes the following energy function:

$$\hat{a} = \arg\min_{\alpha} U(\alpha, \mathbf{k}, \theta, \mathbf{z}) + V(\alpha, \mathbf{z}), \tag{10.3}$$

where $\mathbf{z}$ is the image; α is the segmentation mask with $\alpha_i \in \{\pm 1\}$. θ is the set of parameters of the K components of a GMM and $\mathbf{k}$, $k_n \in \{1, \ldots, K\}$ is the vector assigning each pixel to a unique GMM. The U term encodes the likelihood of each color that exploits the GMM models, and V is the term describing the coherence between neighborhood pixels (see [28] for more details). Intuitively, the key aspect of the GrabCut algorithm is its usage of GMMs to model pixels belonging to background or foreground in the term U. These models represent the distribution of color and are used to assign a label α to each pixel. Using the standard GrabCut, we manually initialize both foreground and background region T_F and T_B to build the respective GMMs.

Exploiting the high-frame rate of egocentric videos it is possible to assume that only slight changes in the foreground and background mixtures will occur between two subsequent frames. This allows us at time t to build a GMM_t based on GMM_{t-1} instead of reinitializing the models. This is equivalent to soft assigning pixels that would end up in the T_U region, which is sensitive to noise.

In our preliminary experimental evaluation we observe that this initialization is necessary, because a segmentation on a bounding box, resulting from the tracking phase, without any assumptions yields poor results. This is due to the fact that small portions of background pixels are often included in the tracked bounding box. When those elements do not appear outside the target region, $p \in T_U$, $p \notin T_B$ (where T_U is the region of pixels marked as unknown), they cannot be correctly assigned to the background by the algorithm and they produce a noisy segmentation.

Once a precise head segmentation is obtained, the resulting image is resized to a fixed size 100×100 (to ensure invariance to scale), converted to grayscale and equalized. On this image a dense HOG descriptor is extracted using 64 cells and 16 bins per cell. To obtain the final feature vector, a power normalization technique has been applied. Using these features, the head pose is then predicted using a multiclass linear SVM classifier following the same quantization used in the landmark based estimation.

In a social scenario where three or more subjects' activity revolves around a discussion or any kind of similar social interaction, orientation transitions are temporally smooth and abrupt changes are avoided as changes tend not to occur when one subject is talking.

To enforce temporal coherence that derives from a video sequence, a stateful hidden Markov model technique is employed. The HMM is a first order Markov chain built upon a set of time-varying unobserved variables/states $\mathbf{z}_t$ and a set of observations $\mathbf{o}_t$. In our case, we set the latent variables to coincide with the possible head poses, while the observed variables are the input images. In practice, we set in the state transition matrix $\mathbf{A}$ a high probability of remaining in the same state, a lower probability for a transition to adjacent states and a very low probability for a transition to the non-adjacent states. This leads our approach to have continuous transitions between adjacent poses, removing impulsive errors that are due to wrong segmentation. This translates into a smooth transition among possible poses, which is what conventionally happens during social interaction among people in egocentric vision settings.

Our method combines the likelihood $p(\mathbf{z}_t|\mathbf{o}_t)$ of a measure $\mathbf{o}_t$ to belong to a pose $\mathbf{z}_t$ provided by the SVM classifier with the previous state $\mathbf{z}_{t-1}$ and

the transition matrix **A** derived from the HMM, obtaining the predicted pose likelihood, which is the final output.

To calibrate a confidence level to a probability in a SVM classifier, so that it can be used as a observation for our HMM, we trained a set of Venn Predictors (VPs) [19], on the SVM training set. We have the training set in the form $S = \{s_i\}_{i=1\ldots n-1}$ where s_i is the input-class pair $(\mathbf{x}_i, y_i)$. Venn predictors aim to estimate the probability of a new element $\mathbf{x}_n$ belonging to each class $Y_j \in \{Y_1 \ldots Y_c\}$. The prediction is performed by assigning each one of the possible classification Y_j to the element $\mathbf{x}_n$ and dividing all the samples $\{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_n, Y_j)\}$ into a number of categories based on a taxonomy. A taxonomy is a sequence $Q_n, n = 1, \ldots, N$ of finite partitions of the space $S^{(n)} \times S$, where $S^{(n)}$ is the set of multisets of S of length n. In the case of multiclass SVM the taxonomy is based on the largest SVM score; therefore each example is categorized using the SVM classification in one of the c classes.

After partitioning the element using the taxonomy, the empirical probability of each classification Y_k in the category τ_{new} that contains (x_n, Y_j) is

$$p^{Y_j}(Y_k) = \frac{|\{(\mathbf{x}^*, y^*) \in \tau_{\text{new}} : y^* = Y_k\}|}{|\tau_{\text{new}}|}. \tag{10.4}$$

This is the pdf for the class of $\mathbf{x}_n$ but after assigning all possible classifications to it we get

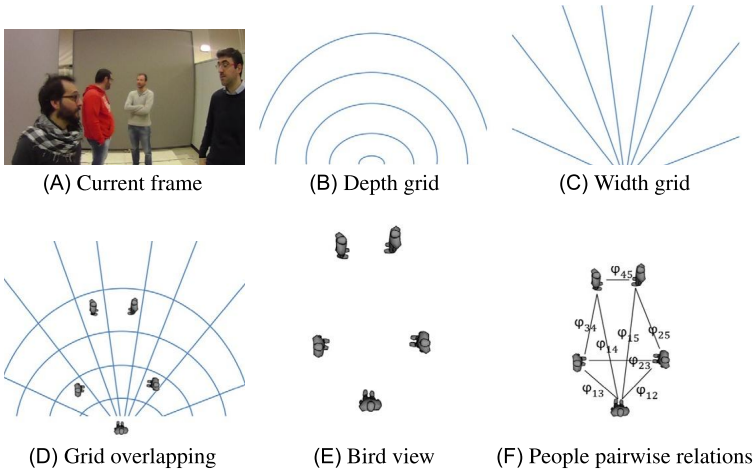
$$P_n = \{p^{Y_j} : Y_j \in \{Y_1, \ldots, Y_c\}\}, \tag{10.5}$$

which is the well-calibrated set of multiprobability predictions of the VP used in the HMM to compute the final output.

10.3.3 3D people localization

To deal with any egocentric camera, we decided not to use any calibration technique in estimating the distance of a subject from the camera wearer. The challenges posed by this decision are somewhat mitigated by the fact that, aiming to detect groups in a scene, the reconstruction of the exact distance is not needed and small errors are lost in the quantization step. We have a depth measure which preserves the positional relations between individual suffices.

With that in mind, we assume that all the heads in the image lie in a plane, so the only two significant dimensions of our 3D reconstruction are (x, z), resulting in a "bird's eye view" model. To estimate the distance from the person wearing the camera, we first use the facial landmarks computed in



■ **FIGURE 10.3** Steps used in our distance estimation process.

the head pose estimation phase. Letting N be the number of landmarks, we build the feature vector

$$\mathbf{d} = \{d_i = \|l_i, l_{i+1}\|, i = 1, \ldots, N-1, l_i \in L\}, \tag{10.6}$$

where $\|\cdot\|$ is the standard euclidean distance. This feature vector is used in a random regression forest [3] trained using the ground truth depth data obtained from a Kinect sensor. To reduce the impact on the distance of a wrong set of landmarks, we apply over a 100 frame window a robust local regression smoothing (RLOESS) based on the LOWESS method [6].

This solution provides a good estimation of the distance between a subject and the camera wearer dealing with the topological deformations that are due to changes in pose and with the non-linearity of the problem.

In the case where the facial landmarks estimator fails, we compute the distance by using a random regression forest trained on the tracked bounding box used as feature. The estimation accuracy of this approach is less than the landmark solution, but it makes our approach more robust in unconstrained scenarios. To estimate the location of a person accounting for the projective deformation in the image, we build a grid with variable cells sizes. The distance allows us to locate the subject with one degree of freedom (x) (Fig. 10.3B): the semicircle in which the person stands is decided based on the distance computed previously, resulting in a quantization of the distance. Using the x position of the person in the image plane and employing a grid capable of accounting for the projective deformation (Fig. 10.3C), it is now possible to place the person with one further degree of freedom z. By over-

lapping the two grids (Fig. 10.3D) the cell in which the person stands can be decided and the bird's eye view model can finally be created (Fig. 10.3E).

Each person is then represented by its location in the 3D space (x, z, o), where o represents the estimated head orientation, and a graph connecting people is created (Fig. 10.3F). Each edge connecting two subjects p and q has a weight ϕ_{pq} which is the feature vector that includes mutual distances and orientations.

10.4 SOCIAL GROUP DETECTION

To deal with the group detection problem, head pose and 3D people information can be used, introducing the concept of the relationship between two individuals. Given two people **p** and **q**, their relationship ϕ_{pq} can be described in terms of their mutual distance, the rotation needed by the first to look at the second and vice versa $\phi_{pq} = (d, o_{pq}, o_{qp})$.

Notice that the distance d is by definition symmetric, while the orientations o_{pq} and o_{qp} are not. An example is given by the situation where two people have the same orientation resulting in **p** looking at **q**'s back; they will have $o_{pq} = 0$ and $o_{qp} = \pi$, so we need two separate features.

Practically it can be hard to fix this definition of relationship and use it in any scenario. In fact, in some contexts people in the group are looking at the same object/scene and none of them looks at any other. Therefore, the need of an algorithm is obvious that is able to differentiate social contexts and learn how to weight distance and orientation features.

10.4.1 Correlation clustering via structural SVM

To detect social groups based on the pairwise relations of their members we use the correlation clustering algorithm [2]. Let **x** be a set of people in the video sequence, their pairwise relations can be encoded by an affinity matrix W, where for $W_{\mathbf{pq}} > 0$ two people **p** and **q** are in the same group with certainty $|W_{\mathbf{pq}}|$ and for $W_{\mathbf{pq}} < 0$ p and q belong to different clusters. The correlation clustering **y** of a set of people **x** is then the partition that maximizes the sum of affinities for item pairs in the same cluster:

$$\arg\max_{\mathbf{y}} \sum_{y \in \mathbf{y}} \sum_{r \neq t \in y} W_{\mathbf{rt}}, \tag{10.7}$$

where the affinity between two people **p** and **q**, $W_{\mathbf{pq}}$, is represented as a linear combination of the pairwise features of orientation and distance over a temporal window. To obtain the best partition of social groups in different

social contexts, our experiments showed that the weight vector $\mathbf{w}$ should not be fixed but, instead, learned directly from the data.

Given an input $\mathbf{x}_i$, a set of distance and orientation features of a set of people, and $\mathbf{y}_i$, their clustering solution, we can observe that a graph describing connections between members suits better the social dimension of the group interaction. This leads to an inherently structured output that is required to be treated accordingly. Structural SVM [29] offers a framework to learn structured outputs. This classifier, given a sample of input–output pairs $S = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, learns the function mapping an input space $\mathcal{X}$ to the structured output space $\mathcal{Y}$.

A discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \Re$ is defined over the joint input–output space. Hence, $F(\mathbf{x}, \mathbf{y})$ can be interpreted as measuring the compatibility of an input $\mathbf{x}$ and an output $\mathbf{y}$. As a consequence, the prediction function f results:

$$f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}, \mathbf{w}) \tag{10.8}$$

where the solution of the inference problem is the maximizer over the label space $\mathcal{Y}$, which is the predicted label. Given the parametric definition of correlation clustering in Eq. (10.7), the compatibility of an input–output pair can be defined by

$$F(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T \sum_{y \in \mathbf{y}} \sum_{r \neq t \in y} \phi_{pq} \tag{10.9}$$

where ϕ_{pq} is the pairwise feature vector of elements p and q. This problem of learning in structured and interdependent output spaces can be formulated as a maximum-margin problem. We adopt the n-slack, margin-rescaling formulation of [29]:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \forall i : \xi_i \geq 0, \\ & \forall i, \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}_i : \mathbf{w}^T \delta\Psi_i(\mathbf{y}) \geq \Delta(\mathbf{y}, \mathbf{y}_i) - \xi_i . \end{aligned} \tag{10.10}$$

Here, $\delta\Psi_i(\mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$, ξ_i are the slack variables introduced to accommodate for margin violations and $\Delta(\mathbf{y}, \mathbf{y}_i)$ is the loss function. In this case, the margin should be maximized in order to jointly guarantee that, for a given input, every possible output result is considered worst than the correct one by at least a margin of $\Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i$, where $\Delta(\mathbf{y}_i, \mathbf{y})$ is bigger when the two predictions are known to be more different. We rely on the cutting plane

algorithm in which we start with no constraints, and iteratively we find the most violated one and re-optimize it until convergence.

The problem of group detection is similar to the noun-coreference problem [5] in NLP, where nouns have to be clustered according to who they refer to. For this problem, recently a suitable scoring measure has been proposed: the MITRE loss function [30]. It, or formally $\Delta_M(\mathbf{y}, \bar{\mathbf{y}})$, is based on the understanding that, instead of representing each subject's links towards every other person, connected components are sufficient to describe dynamic groups and thus spanning trees can be used to represent clusters.

10.5 SOCIAL RELEVANCE ESTIMATION

Group detection estimates how people interact with each other by analyzing the geometry of their social formations. What cannot be unveiled by detecting social groups are all the properties of the social formation which depend on the particular observer, like the importance attached by an observer to each person in a group. We name this subjective property *social relevance*. By providing complementary information to what is provided by group estimation, we argue that social relevance enables a better understanding of the social dynamics from an egocentric prospective. Clearly, social relevance cannot be fully estimated from features like head pose and distance. To some extent, the camera wearer could give more importance to a distant person than to a closer one, or even to somebody who is turned away.

Sensors that can objectively measure the relevance of a person from the point of view of an observer, like eye-tracking glasses, are expensive and more uncomfortable to wear in public than a tiny camera. Therefore, to estimate social relevance relying only on the frames captured by a wearable camera, we choose to rely on saliency prediction [7,8,13,15]. Saliency prediction architectures predict the distribution of eye fixation points on a given image, and are trained on data captured from eye-tracking devices. By providing a distribution of eye fixations over an image, this lets us estimate the amount of fixations each person on the scene would receive from the wearer, and by extension, the social relevance of each person. More in detail, given a video, we first extract the saliency maps for all video frames. We then define the social relevance of each person as the accumulation of the saliency values inside the person's bounding box summed over time. In this way, for each subject appearing in the video frames, we can obtain a measure of his individual relevance to the social interaction.

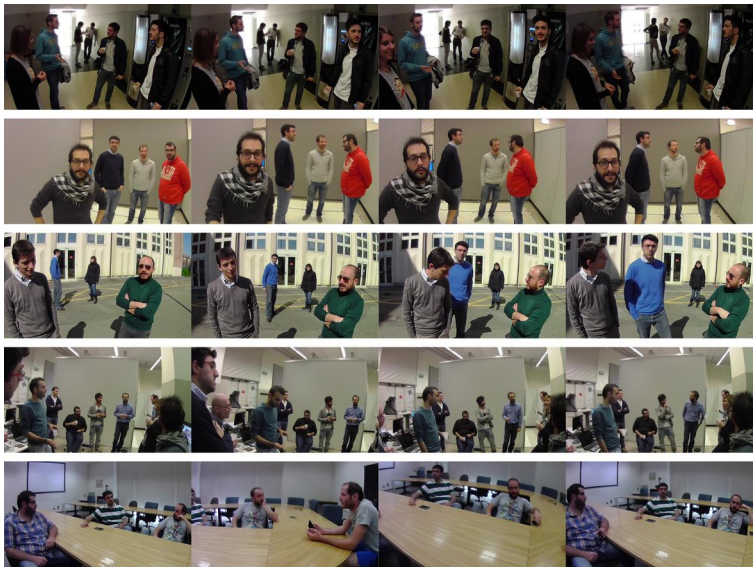
To compute saliency maps, we employ the Saliency Attentive Model (SAM[1]) presented in [8], which has shown state-of-the-art results on popular saliency benchmarks, such as the MIT Saliency Benchmark [4] and the SALICON dataset [16]. This model is composed of three main components. First, a Convolutional Neural Network (CNN) extracts a set of feature maps from the original image. Because of the presence of spatial pooling operations, which compute the maximum activation over a sliding window, this would largely downscale the activations of the last layers with respect to the original image size. As is easy to see, a scaled output reduces the accuracy of predictions and their localization accuracy, which is fundamental in the context of predicting the saliency on faces that might occupy only a small portion of the frame. To control this phenomenon, we employ dilated convolutions. In short, dilation increases the spatial support of convolutions, by enlarging the kernel size, but keeps the number of parameters constant, setting the pixels of the kernel to zero at evenly spaced locations. The feature maps coming from the CNN are then fed through a recurrent layer which, thanks to the incorporation of attentive mechanisms, selectively attends to different regions of the input. In particular, we use a Long Short-Term Memory network (LSTM) with convolutional operations to sequentially refine and enhance the input feature maps. Predictions are finally combined with multiple prior maps directly learned by the network, thus effectively incorporating the center bias present in human eye fixations.

10.6 EXPERIMENTAL RESULTS

To provide an evaluation of the proposed social groups detection method and its main components, we rely on two publicly available datasets, namely EGO-HPE for evaluating the head pose estimation component and EGO-GROUP to assess the performance of group detection, distance estimation and social relevance.

The EGO-HPE dataset[2] is used to test the proposed head pose estimation method. It features more than 3400 frames where bounding boxes and head pose labels are provided for every person in the frame. Aiming at specifically evaluating the head pose estimation in egocentric vision, this dataset features the typical challenges of body-worn cameras, such as background clutter, different lighting conditions and motion blur.

[1] Source code available at http://github.com/marcellacornia/sam. The method has also won the LSUN Saliency Challenge in 2017.
[2] http://imagelab.ing.unimore.it/files/EGO-HPE.zip.



■ **FIGURE 10.4** Example sequences from the EGO-GROUP dataset.

On the other hand, the EGO-GROUP dataset[3] features 18 video sequences in the five different scenarios represented in Fig. 10.4: a coffee break scenario with very poor lighting and random backgrounds (first row), a laboratory setting with limited background clutter and fixed lighting conditions (second row), an outdoor scenario (third row), a festive moment with a crowded environment (fourth row), and a conference room setting where people's movement and positioning is tied to seats (fifth row). A total of 23 different subjects appear in the videos.

10.6.1 Head pose estimation

Among the different features employed in the social group detection, head pose estimation is arguably one of the most important. In fact, errors in the head pose create a strong bias in the features used in the group estimation.

To provide the best results, our head pose estimation relies on two steps: landmark estimation and HOG-based pose classification. Both approaches have different characteristics: facial landmarks are very accurate and fast but their performance drops quickly when facing more extreme head poses, making them suitable for near frontal images but unreliable under steep pose angles. On the other hand, shape features such as histogram of gradients ex-

[3] http://imagelab.ing.unimore.it/files/EGO-GROUP.zip.

Table 10.1 Comparison between different techniques. In the different methods, PN indicates the usage of power normalization, HMM indicates the use of HMM-based temporal smoothing

Method	EGO-HPE1	EGO-HPE2	EGO-HPE3	EGO-HPE4
HOG+PN	0.710	0.645	0.384	0.753
HOG+PN+HMM	0.729	0.649	0.444	0.808
Landmarks	0.537	0.685	0.401	0.704
Landmarks+HOG	0.750	**0.731**	0.601	0.821
Landmarks+HOG+HMM	**0.784**	0.727	**0.635**	**0.821**

cel at discriminating steep head poses and can complete the estimation when landmarks cannot be reliably computed. Furthermore, HOG descriptors are much less sensitive to scale, which can be helpful when dealing with subjects in the background.

Table 10.1 compares the two approaches, clearly showing that the combination of landmarks and HOG features achieves the best results.

To show how egocentric vision's unique perspective can affect the results of an approach if not explicitly taken into account, we tested our egocentric head pose estimation method against other recent methods over the EGO-HPE dataset. The first method we compared to is proposed by Zhu et al. [36]: by building a mixture of trees with a shared pool of parts, where each part represents a facial landmark, they use a global mixture in order to capture topological changes in the face due to the viewpoint, effectively estimating the head pose. To achieve a fair comparison in terms of required time, we used their fastest pretrained model and reduced the number of levels per octave to one. This method, while being far from real-time, provides extremely precise head pose estimations even in egocentric vision scenarios when it can overcome detection difficulties. The second method used in our comparison is [10]. This method provides real-time head pose estimations by using facial landmark features and a regression forest trained with examples from five different head poses. Table 10.2 shows the results in terms of the accuracy of this comparison.

10.6.2 Distance estimation

To assess the quality of our distance estimation method, by keeping the regression architecture unvaried, we test two commonly employed techniques. The first relies on using the dimensions of the head bounding box as features, while the second one uses the area of the segmented face. Table 10.3 shows this comparison in terms of absolute error. The *Bounding Box* method employs the TLD tracker in order to estimate the subject's bounding box,

Table 10.2 Comparison of our head pose estimation and two recent methods on EGO-HPE dataset

	Our method	Zhu et al. [36]	Dantone et al. [10]
EGO-HPE1	**0.784**	0.685	0.418
EGO-HPE2	**0.727**	0.585	0.326
EGO-HPE3	**0.635**	0.315	0.330
EGO-HPE4	**0.821**	0.771	0.634

Table 10.3 Comparison between different distance estimation approaches

Method	Abs. error
Area (baseline)	12.67
Bounding Box	5.59
Landmarks	1.91
Landmarks + Moving Average	1.72
Landmarks + LOESS	1.68
Landmarks + RLOESS	**1.60**

while the *Area* method relies on the segmentation of the face surface. The results show that using biologically-inspired features such as the ratio between different facial landmarks can greatly improve the results when compared to other methods.

Aiming to improve our results, we apply to our distance sequence a smoothing filter. As Table 10.3 shows, using a moving average filter can improve the results by 9,95%, while LOESS and RLOESS smoothing methods yield, respectively, an error reduction of 12.04% and 16.23%. In both the LOESS and the RLOESS methods the span has been set to 10% of the data.

10.6.3 Groups estimation

A critical issue when training the social group detection method is how to deal with data from different scenarios. In fact, depending on the social situation, distance and pose features can have different significance, e.g. when there is not much space available, people cluster together regardless of their will to form a coherent group. For this reason, training should be *context dependent*. Table 10.4 reports the performance of our method on the EGO-GROUP dataset, where the training is repeated on the first video of each scenario. Furthermore, results obtained by training over the union of the individual training sets are also reported. To the best of our knowledge, no other methods deal with the estimation of social groups in egocentric scenarios.

Table 10.4 Comparison between training variations on our method. The table shows how different training choices can deeply impact the performances. All tests have been performed using a window size of eight frames

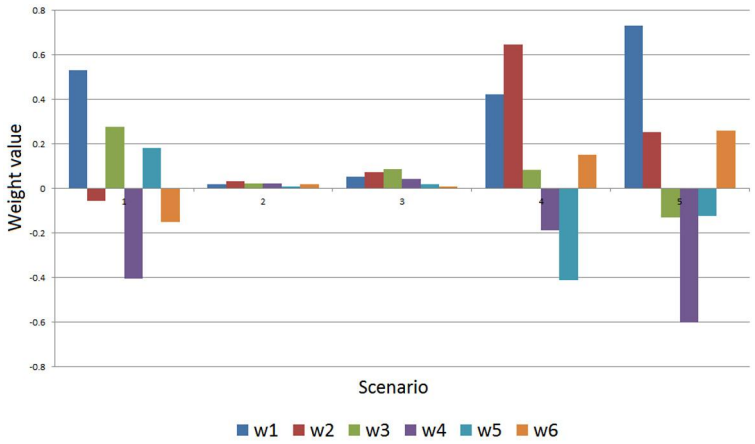
Test scenario	**Training: Laboratory**			**Training: Coffee**			**Training: Party**		
	Error	***Precision***	***Recall***	***Error***	***Precision***	***Recall***	***Error***	***Precision***	***Recall***
Coffee	10.74	83.04	97.29	9.23	82.67	100.00	18.04	68.76	100.00
Party	9.33	100.00	83.63	0.00	100.00	100.00	0.00	100.00	100.00
Laboratory	11.91	91.68	85.79	14.75	74.67	99.43	14.43	74.81	100.00
Outdoor	11.47	87.88	95.11	10.22	82.09	98.27	11.30	81.17	100.00
Conference	16.27	75.24	93.32	14.56	73.94	95.15	18.97	75.58	95.28
Test scenario	**Training: Outdoor**			**Training: Conference**			**Training: All**		
	Error	***Precision***	***Recall***	***Error***	***Precision***	***Recall***	***Error***	***Precision***	***Recall***
Coffee	6.80	92.54	94.92	13.88	79.99	88.41	8.11	85.50	99.60
Party	10.92	100.00	80.34	7.11	90.12	95.42	3.15	96.27	98.05
Laboratory	27.75	72.60	72.81	12.02	90.75	87.22	19.97	74.32	88.05
Outdoor	16.22	81.11	90.24	16.71	74.92	94.81	16.24	84.33	88.67
Conference	14.46	74.09	95.20	13.95	74.67	95.10	17.07	74.04	93.73

Table 10.5 Comparison between training the correlation clustering weights using SSVM and performing clustering without training (fixed weights). The window size used in the experiment is 8

Method		Coffee	Party	Laboratory	Outdoor	Conference
CC	Error	12.75	0.00	14.28	17.13	15.54
	Precision	74.86	100.00	73.12	71.81	74.43
	Recall	96.29	100.00	97.55	97.98	91.39
CC+SSVM	Error	9.23	0.00	11.91	16.22	13.95
	Precision	82.67	100.00	91.68	81.11	74.67
	Recall	100.00	100.00	85.79	90.24	95.10

From the data reported in Table 10.4, it can be noticed how training in specific scenarios can result in overfitting. For example, the weights learned by training on the *outdoor* sequences provide better results when testing on *coffee* than the ones trained on *coffee* itself. This is due to overfitting on a particular dynamic present in both scenarios but, unsurprisingly, it provides poor performances when testing on videos from other scenarios. In order to have an estimate of how different training methods perform, the standard deviation over the absolute error can be computed. It emerges that the *laboratory* setting is the more general training solution with an average error of 11.94 and a standard deviation of 2.61, while training over the *party* sequence, although it can achieve impeccable results over its own scenario and an average error of 12.55, presents a much higher deviation (7.65). Training over the set given by the union of each training set from the different scenarios results in a standard deviation of 7.01 over a mean error of 12.91, showing how this solution, while maintaining the overall error rates, does not provide a gain in generality. This confirms that different social situations call for different feature weights and that context-dependent training is needed to adapt to how humans change their behavior based on the situation.

To stress the importance of domain specific training of the feature weights, we report results of trainingless clustering. That is, all the feature weights are fixed at the same value, effectively assigning the same importance to distance and orientation features. Table 10.5 reports these results: it can be noticed how the algorithm is often biased towards placing all the subjects into one single group. This is showed by the high recall and the lower precision: the MITRE loss function penalizes precision for each person put in the wrong group, while the recall stays high. Placing every person in the same group hence results in an average error due to the fact that, not leaving any subject out of a group provides a high recall. In our experiments, we

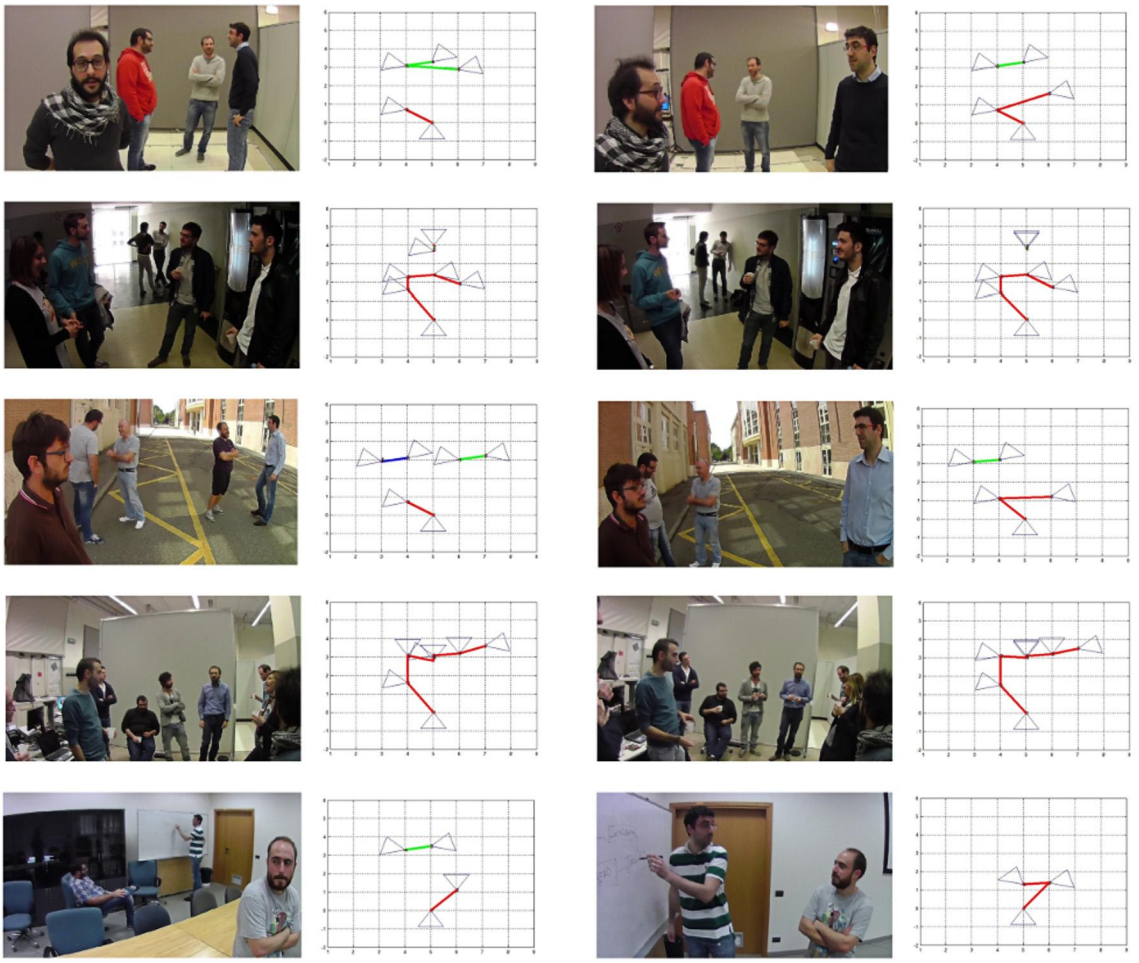


■ **FIGURE 10.5** Weight values in the five different training scenarios. The scenarios are: *1) laboratory, 2) party, 3) conference, 4) coffee, 5) outdoor.*

set the clustering window size to eight frames, a value that our preliminary experiments showed to achieve a good compromise between robustness to noise and fine grained responsiveness to group changes.

To further evaluate our approach, we discuss how the clustering weights vary in different scenarios. Fig. 10.5 shows the comparison between the different components of the weight vectors. As can be noticed, performing the training over different scenarios yields significantly different results. For example, clustering a sequence in the fourth scenario gives more importance to the second feature (the orientation of subject 1 towards subject 2), slightly less importance to the spatial distance between the two and very little importance to the orientation of 2 towards 1. In scenario 5, the outdoor sequence, the most important feature is recognized to be the distance, correctly reflecting the human behavior where, being outdoor, different groups tend to increase the distance between each other thanks to the high availability of space (Fig. 10.6).

A negative weight models the fact that, during the training, our approach has learned that the feature that weight relates to can decrease the affinity of a pair. A typical example of such situation is when a person is giving us the back: while our orientation can have a high similarity value towards that person, that feature will probably lead the system to wrongly put us in the same group. Our approach learns that there are situations where some features can produce wrong clustering results and assign a negative weight to them.

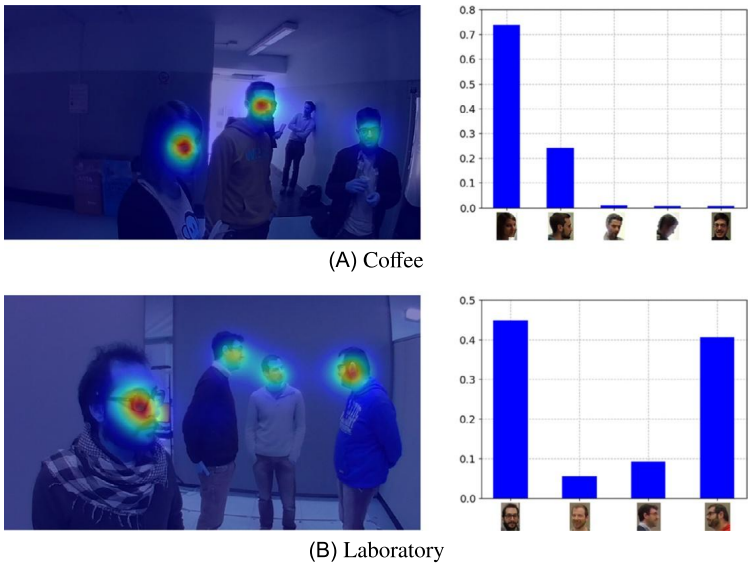


■ **FIGURE 10.6** Examples of the results obtained by our method. Different groups are shown by different link colors.

10.6.4 Social relevance

While the group estimation evaluated in the previous section describes how people interact with each other, we also evaluate the individual relevance of the subjects that partake to the social interaction. In particular, we argue that while head pose and distance information are instrumental in estimating social groups, visual saliency can be used to understand who are the most relevant subjects according to what is recorded by the egocentric camera.

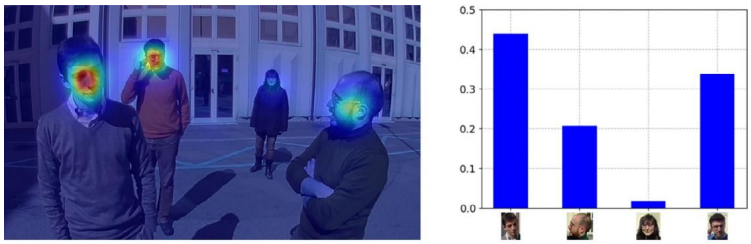
Here, we rely on the method described in Section 10.5 to compute saliency maps of the EGO-GROUP dataset sequences. The overall saliency value of



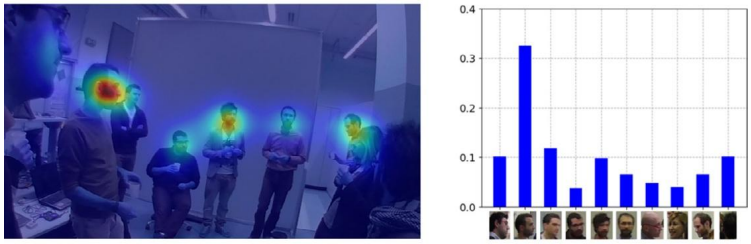
■ **FIGURE 10.7** Sample results on the social relevance of each subject. Left: saliency map overlay. Right: social relevance of the participants to each considered scenario.

a person is then obtained by accumulating the raw saliency contained in the person's bounding box and summing it over the temporal dimension. The insight behind this is that, while group estimation captures the overall group dynamics, not all the members of a group may have the same importance for the person wearing the camera and analyzing the social relevance through the saliency estimation can provide information complementary to what is provided by the group estimation.

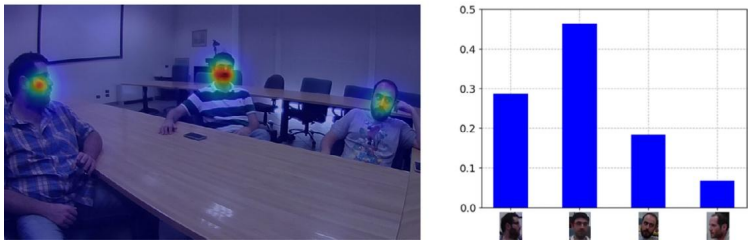
Fig. 10.7 provides qualitative results of this analysis on five different scenarios. On the left, a sample frame with the saliency map overlay is provided, while on the right side a plot of the individual saliency scores of the participants is provided. It may be noticed how the information provided by evaluating the saliency of individual participants is complementary to the group data. In fact, taking Fig. 10.7A as an example, the saliency estimation provides remarkable cues on who are the most interesting members of the foreground group, while it agrees with the group estimation in assigning low relevance to the people forming the background group. Similarly, in the *party* scenario (Fig. 10.7D) there is only one big group, and evaluating the individual saliency values can provide a better insight on the intra-group dynamics.



(C) Outdoor



(D) Party



(E) Conference

■ **FIGURE 10.7** (*continued*)

10.7 **CONCLUSIONS**

In this chapter we presented a novel approach to detecting social groups using a head-mounted camera. The proposal relies on a head pose classification technique combining landmarks and shape descriptors in a temporally smoothed HMM framework. Furthermore, 3D location estimation of the people without the need of camera calibration is also presented. Using this information, the approach is able to build a "bird's eye view" model that is the input of the supervised correlation clustering in order to detect the group in different contexts. An extensive experimental evaluation shows competitive performance on two publicly available egocentric vision datasets, recorded in real and challenging scenarios.

REFERENCES

[1] Maedeh Aghaei, Mariella Dimiccoli, Cristian Canton Ferrer, Petia Radeva, Towards social pattern characterization in egocentric photo-streams, arXiv preprint, arXiv: 1709.01424, 2017.

[2] Nikhil Bansal, Avrim Blum, Shuchi Chawla, Correlation clustering, Mach. Learn. 56 (2004) 89–113.

[3] Leo Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[4] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, Antonio Torralba, MIT saliency benchmark, http://saliency.mit.edu/, 2017.

[5] Claire Cardie, Kiri Wagstaff, Noun phrase coreference as clustering, in: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.

[6] William S. Cleveland, Robust locally weighted regression and smoothing scatterplots, J. Am. Stat. Assoc. 74 (368) (1979) 829–836.

[7] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, Rita Cucchiara, A deep multi-level network for saliency prediction, in: International Conference on Pattern Recognition, 2016.

[8] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, Rita Cucchiara, Predicting human eye fixations via an LSTM-based saliency attentive model, IEEE Trans. Image Process. 27 (10) (2018) 5142–5154.

[9] Marco Cristani, Loris Bazzani, Giulia Paggetti, Andrea Fossati, Diego Tosato, Alessio Del Bue, Gloria Menegaz, Vittorio Murino, Social interaction discovery by statistical analysis of F-formations, in: British Machine Vision Conference, 2011.

[10] Matthias Dantone, Juergen Gall, Gabriele Fanelli, Luc Van Gool, Real-time facial feature detection using conditional regression forests, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2012.

[11] Alircza Fathi, Jessica K. Hodgins, James M. Rehg, Social interactions: a first-person perspective, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2012.

[12] Kohsia S. Huang, Mohan M. Trivedi, Robust real-time detection, tracking and pose estimation of faces in video streams, in: International Conference on Pattern Recognition, 2004.

[13] Xun Huang, Chengyao Shen, Xavier Boix, Qi Zhao, SALICON: reducing the semantic gap in saliency prediction by adapting deep neural networks, in: IEEE International Conference on Computer Vision, 2015.

[14] Hayley Hung, Ben Kröse, Detecting F-formations as dominant sets, in: ACM International Conference on Multimodal Interaction, 2011.

[15] Laurent Itti, Christof Koch, Ernst Niebur, et al., A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.

[16] Ming Jiang, Shengsheng Huang, Juanyong Duan, Qi Zhao, SALICON: saliency in context, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015.

[17] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas, Tracking-learning-detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1409–1422.

[18] Adam Kendon, Studies in the Behavior of Social Interaction, vol. 6, Humanities Press Intl, 1977.

[19] Antonis Lambrou, Harris Papadopoulos, Ilia Nouretdinov, Alexander Gammerman, Reliable probability estimates based on support vector machines for large multiclass datasets, in: Artificial Intelligence Applications and Innovations, vol. 382, 2012, pp. 182–191.

[20] Deqiang Li, Witold Pedrycz, A central profile-based 3d face pose estimation, Pattern Recognit. 47 (2) (2014) 525–534.
[21] Bingpeng Ma, Wenchao Zhang, Shiguang Shan, Xilin Chen, Wen Gao, Robust head pose estimation using LGBP, in: International Conference on Pattern Recognition, 2006.
[22] Erik Murphy-Chutorian, Mohan Manubhai Trivedi, Head pose estimation in computer vision: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 31 (4) (2008) 607–626.
[23] Rhys Newman, Yoshio Matsumoto, Sebastien Rougeaux, Alexander Zelinsky, Real-time stereo tracking for head pose and gaze estimation, in: International Conference on Automatic Face and Gesture Recognition, 2000.
[24] Nicoletta Noceti, Francesca Odone, Humans in groups: the importance of contextual information for understanding collective activities, Pattern Recognit. 47 (11) (2014) 3535–3551.
[25] Shay Ohayon, Ehud Rivlin, Robust 3D head tracking using camera pose estimation, in: International Conference on Pattern Recognition, 2006.
[26] Javier Orozco, Shaogang Gong, Tao Xiang, Head pose classification in crowded scenes, in: British Machine Vision Conference, 2009.
[27] Carsten Rother, Vladimir Kolmogorov, Andrew Blake, "GrabCut": interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (3) (2004) 309–314.
[28] Brandon M. Smith, Jonathan Brandt, Zhe Lin, Li Zhang, Nonparametric context modeling of local appearance for pose- and expression-robust facial landmark localization, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2014.
[29] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, Yasemin Altun, Support vector machine learning for interdependent and structured output spaces, in: International Conference on Machine Learning, 2004.
[30] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, Lynette Hirschman, A model-theoretic coreference scoring scheme, in: Message Understanding Conference, 1995.
[31] Junwen Wu, Jens M. Pedersen, D. Putthividhya, Daniel Norgaard, Mohan M. Trivedi, A two-level pose estimation framework using majority voting of gabor wavelets and bunch graph analysis, in: International Conference on Pattern Recognition Workshops, 2004.
[32] Xuehan Xiong, Fernando De la Torre, Supervised descent method and its applications to face alignment, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2013.
[33] Xu Yan, Ioannis A. Kakadiaris, Shishir K. Shah, Modeling local behavior for predicting social interactions towards human tracking, Pattern Recognit. 47 (4) (2014) 1626–1641.
[34] Ryo Yonetani, Kris M. Kitani, Yoichi Sato, Recognizing micro-actions and reactions from paired egocentric videos, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2016.
[35] Ting Yu, Ser-Nam Lim, Kedar Patwardhan, Nils Krahnstoever, Monitoring, recognizing and discovering social networks, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009.
[36] Xiangxin Zhu, Deva Ramanan, Face detection, pose estimation and landmark localization in the wild, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2012.