From Ego to Nos-vision: Detecting Social Relationships in First-Person Views

Stefano Alletto, Giuseppe Serra, Simone Calderara, Francesco Solera and Rita Cucchiara
Università degli Studi di Modena e Reggio Emilia
Via Vignolese 905, 41125 Modena - Italy
{*name.surname*}*@unimore.it*

Abstract

In this paper we present a novel approach to detect groups in ego-vision scenarios. People in the scene are tracked through the video sequence and their head pose and 3D location are estimated. Based on the concept of f-formation, we define with the orientation and distance an inherently social pairwise feature that describes the affinity of a pair of people in the scene. We apply a correlation clustering algorithm that merges pairs of people into socially related groups. Due to the very shifting nature of social interactions and the different meanings that orientations and distances can assume in different contexts, we learn the weight vector of the correlation clustering using Structural SVMs. We extensively test our approach on two publicly available datasets showing encouraging results when detecting groups from first-person camera views.

Figure 1: An example of our method's output. People different colors in segmentation indicate their belonging to different groups. The red dot represents the first-person wearing the camera.

1. Introduction

Wearable computing devices are becoming more and more common: first-person camera views present some unique advantages if compared to the setting used in the last 20 years of computer vision. The video is recorded by the same perspective humans see, focusing exactly on what we focus, seeing what we see. Ego-vision applications hence have a great potential allowing for a completely new approach to social analysis, object detection and recognition or human actions recognition. Some great challenges come with this new scenario as well: having a camera tied to one's head instead of being placed in a fixed position presents strong ego-motion, background clutter and the ability to move with the camera can also imply strong changes in lighting conditions.

Recently, efforts in the direction of a better understanding of human-objects interactions or egocentric video summarization have been made [11, 13, 12]. Furthermore, social interaction is a very interesting field due to the unique perspective of ego-vision. In particular, the work by Fathi *et al*. [6] aims to the recognition of five different social situations (monologue, dialogue, discussion, walking dialogue, walking discussion). By using day-long videos recorded from an egocentric perspective in an amusement park, they extract features like the 3D position of faces around the recorder and ego-motion. They estimate the head pose of each subject in the scene, calculate their line of sight and estimate the 3D location they are looking at under the assumption that a person in a social scenario is much more likely to look at other people. A multi-label HCRF model is then used to assign a category to each social situation in the video sequence.

Differently, in this paper we address the problem of partitioning people in the scene into socially related groups. Human behavior is by no means random: when interacting with each other we naturally tend to place ourselves in determined positions to avoid occlusions, stand close to the ones we interact with and organize orientations so as to naturally place the focus on the subjects of our interest. In order model this behavior, we follow the formalism of the *f-formation* defined in [10]. A *f-formation* is a pattern that people naturally tend to create when interacting 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 theory has been successfully applied in recent works aimed at social interaction analysis showing great promise [4, 9]. The idea behind our approach is to adopt distance and orientation information and use them to build a pairwise feature vector capable of describing how two people stand in relation to one another. In this paper we present a novel framework for detecting social groups by using a correlation clustering algorithm that exploits social features to truly capture the social clues inferred from human behavior. In order to achieve this result, we present (i) a novel head pose estimation framework developed for ego-vision, (ii) a 3D scene reconstruction method capable of estimating the position of people without relying on calibration, (iii) a Structural SVM based approach to learn how to weight each component of the feature vector depending on the social situation is applied to. Our experimental results (see an example in Figure 1) on two publicly available datasets show that our approach is capable of dealing with the complex challenges of the egocentric point of view. To our knowledge, our work is the first that tackles the group detection task in an ego-centric video scenario.

2. Group detection

To present how our method deals with the group detection problem, we formally introduce the concept of relationship between individuals. Given two people $\mathbf{r}$ and $\mathbf{t}$, we describe their relation ϕ_{rt} in terms of the distance between the two, the rotation needed by the first to look at the second and vice versa $\phi_{rt} = (d, o_{rt}, o_{tr})$. Note that d is symmetric while o_{rt} and o_{tr} are not and thus the need of two orientation features instead of just one. This can be better explained with an example: if two people are facing each other, $o_{rt} = o_{tr} = 0$; on the contrary if they both have the same orientation resulting in $\mathbf{r}$ looking at $\mathbf{t}$'s back, we will have $o_{rt} = 0$ and $o_{tr} = \pi$.

In practice, it can often be hard to fix this definition of relationship and use it independently from the scenario, mainly due to the fact that different situations can form groups in very different manners. Sometimes people are in the same group because of the mutual orientations and distances or sometimes they are all looking at the same object and none of them looks at any other group member. In any case, it clearly emerges the need for an algorithm capable of adapting to different situations *learning* how to treat distance and orientation features depending on the context.

2.1. Correlation Clustering via Structural SVM

In order to categorize groups given the pairwise relations of their members we used the correlation clustering algorithm [1]. In particular given a set of people $\mathbf{x}$ in front of the camera we describe their pairwise relations with an affinity matrix W, where for $W_{\mathbf{rt}} > 0$ two people $\mathbf{r}$ and $\mathbf{t}$ are in the same group with certainty $|W_{\mathbf{rt}}|$ and for $W_{\mathbf{rt}} < 0$ r and t belong to different clusters. The correlation clustering $\mathbf{y}$ of a set of people $\mathbf{x}$ is then the partition that maximize the sum of affinities for item pairs in the same cluster:

$$\arg\max_{\mathbf{y}} \sum_{y \in \mathbf{y}} \sum_{r \neq t \in y} W_{\mathbf{rt}} \quad (1)$$

where the affinity between subjects $\mathbf{t}$ and $\mathbf{r}$, W_{rt}, is modeled as a linear combination of the pairwise features of orientation and distance over a temporal window. The window size determines how many frames are used to calculate the groups, capturing variations among the groups composition and maintaining robustness to noise. In order to obtain the best way to partition people into groups in the current social situation, the weight vector $\mathbf{w}$ should not be fixed but learned directly from the data.

Being the input $\mathbf{x}_i$ a set of distance and orientation features of a set of people and $\mathbf{y}_i$ their clustering solution it is easy to notice that the output cannot be modeled by a single valued function, a graph describing connections between members, which is inherently structured, should instead be employed. Structural SVM [14] offers a generalized framework to learn structured outputs by solving a loss augmented problem. The classifier learns the function mapping the input space $\mathcal{X}$ to the structured output space $\mathcal{Y}$, given a sample of input-output pairs $S = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$. A discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \Re$ is defined over the joint input-output space so that $F(\mathbf{x}, \mathbf{y})$ can be interpreted as measuring the compatibility of $\mathbf{x}$ and $\mathbf{y}$. As a consequence the prediction function f results

$$f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}) \quad (2)$$

where the maximizer over the label space $\mathcal{Y}$ is the predicted label, *i.e.* the solution of the inference problem. Following the parametric definition of correlation clustering in Eq. 1 the compatibility of an input-output pair can be defined as

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T \sum_{y \in \mathbf{y}} \sum_{r \neq t \in y} \phi_{rt} \quad (3)$$

where ϕ_{rt} is the pairwise feature vector of elements r and t. The problem of learning in structured and interdependent output spaces can been formulated as a maximum-margin problem. We adopt the n-slack, margin-rescaling formulation of [14]:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \forall i : \xi_i \geq 0, \\ & \forall i, \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}_i : \mathbf{w}^T \delta\Psi_i(\mathbf{y}) \geq \Delta(\mathbf{y}, \mathbf{y}_i) - \xi_i, \end{aligned} \quad (4)$$

where $\delta\Psi_i(\mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$, ξ_i are the slack variables introduced in order 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.

The quadratic program in Eq. 4 introduces a constraint for every possible wrong clustering of the set. Unfortunately, the number of wrong clusterings scales more than exponentially with the number of items. As we aim to real-time performances, approximated optimization schemes need to be considered. In particular we adopt the cutting plane algorithm where we start with no constraints, and iteratively find the most violated constraint:

$$\hat{\mathbf{y}}_i = \arg\max_{\mathbf{y}} \Delta(\mathbf{y}_i, \mathbf{y}) - \delta\Psi_i(\mathbf{y}) \tag{5}$$

and re-optimize until convergence. Finding the most violated constraint requires to solve the correlation clustering problem, which we know to be NP-hard [1]. Finley *et al.* [7] propose a greedy approximation algorithm which works by initially considering each person in its own cluster, then iteratively merging the two clusters whose union would produce the worst clustering score.

One remarkable aspect of supervised correlation clustering is that there is no need to know in advance how many groups are present in the scene. Moreover two elements could end up in the same cluster if the net effect of the merging process is positive even if their local affinity measure is negative, implicitly modeling the transitive property of relationships in groups which is known from sociological studies.

2.2. Loss function

The learning ability of the algorithm highly depends on the choice of the loss function since it has the power to force or relax input margins.

The problem of clustering people is in many ways similar to the noun-coreference problem [3] in NLP, where nouns have to be clustered according to who they refer to. Above all, the combinatorial number of potential connections is shared. For this problem, the MITRE score [16] has been identified as a suitable scoring measure. The **MITRE loss**, $\Delta_M(\mathbf{y}, \bar{\mathbf{y}})$, is founded on the understanding that connected components are sufficient to describe dynamic groups and thus spanning trees can be used to represent clusters.

Consider two clustering solutions $\mathbf{y}, \bar{\mathbf{y}}$ and an instance of their respective spanning forests Q and P. The connected components of Q and P are identified respectively by the trees $Q_i, i = 1, \ldots, n$ and $P_i, i = 1, \ldots, m$. Let $|Q_i|$ be the number of people in group Q_i and $p(Q_i)$ the set of subgroups obtained by considering only the relational links in Q_i that are also found in the partition P. A detailed derivation of this measure can be found in [3].

Accounting for all trees Q_i we define the global recall measure of Q as

$$\mathcal{R}_Q = \frac{\sum_{i=1}^{n} |Q_i| - |p(Q_i)|}{\sum_{i=1}^{n} |Q_i| - 1} \tag{6}$$

The precision of Q can be computed by exchanging Q and P, which can be also seen as the recall of P with respect to Q, guaranteeing that the measure is symmetric. Given the recall $\mathcal{R}$ the loss is defined as

$$\Delta_M = 1 - F_1 \tag{7}$$

where F_1 is the standard F-score.

3. Understanding people

A lot of work has been done in detecting, tracking and locating people in 3D environments. This is indeed the first step towards any kind of social interaction study or, in our case, social groups detection.

3.1. Detection and Tracking

In an ego-vision scenario where steep head poses, occlusions or quick changes in lighting conditions can easily occur, even face detection can still be a problem. In order to cope with the complexity of this scenario, our method makes use of the Hough-Based Tracker (HBT) [8]. An extremely useful step of its tracking process is the segmentation of the object it perform as it will be discussed in Section 3.2. We extended and parallelized HBT in order to track simultaneously multiple targets in real-time, we also introduced an automatic initialization step using Viola-Jones face detector. In practice, due to the complexity of many ego-vision scenarios, we often rely on manual initialization on the first frame of the video sequence in order to cope with hard detection situations that could compromise the following steps of our framework.

3.2. Head Pose Estimation

By calculating a rough estimate of someone's head pose is possible to understand with a certain precision where they are looking at. In order to achieve this result in first-person camera views, a two-step approach is used: the first step consists in obtaining a first estimation using spatial features. Given the head bounding box and segmentation provided by our tracking phase (see Section 3.1), a few steps of normalization are applied in order to achieve robustness to various factors such as lighting and scale: contrast normalization, resizing and background subtraction. Eventually, a $8\times$

8×16-dimensional dense HOG descriptor is extracted and a further numeric normalization is applied through *power normalization*: $f(\mathbf{x}) = sign(\mathbf{x})|\mathbf{x}|^{\alpha} \quad with \quad \alpha = 0.5$.

By applying this function over the feature vectors it is possible to improve the classification performances. The resulting feature vector is then classified by a Linear SVM providing a first real-time estimate of the subject orientation.

A second step introduces temporal consistency. In fact, when people talk in a group they usually focus their attention on the one who has the floor, resulting in constant poses for a while and changing when someone different starts talking. A Hidden Markov Model (HMM) is hence introduced, resulting in a set of latent variables z_t coinciding with the head poses, and a set of observations o_t which are the input images of the head. The joint probability of a state z_t and an observation o_t is given by:

$$p(\mathbf{z}_t, \mathbf{o}_t) = p(\mathbf{z}_0) \prod_{t=1}^{T} p(\mathbf{o}_t|\mathbf{z}_t) p(\mathbf{z}_t|\mathbf{z}_{t-1}). \tag{8}$$

The state transition probability $p(z_t|z_{t-1})$, modeled with a transition matrix A, effectively introduces constraints over the set of possible transitions. By controlling this matrix's values one can add temporal consistency to the framework deciding which transitions are possible at a state z_t and which are not. The output of the HMM model is then treated as the final predicted pose of the subject's head.

3.3. 3D People localization

In order to determine the 3D position of each person in the scene, we decided not to use camera calibration due to the loss of generality that this would have resulted in an ego-vision scenario. Aiming to detect the groups in the scene, we do not need the exact position of each person but a location estimation which has to maintain positional relations between individuals. We rely on the assumption that all the heads in the image lay on a plane, thus the only two significant dimensions of our 3D reconstruction are (x, z), resulting in a "bird view" model. In order to estimate the distance from the person wearing the camera, we trained a Random Regression Forest [2] using the area of the head on the image plane, obtained by the segmentation resulted from the HBT tracker.

This provides a good estimation of the distances near the first-person while coping well with the non-linearity of the problem at hand. In order to estimate the x position accounting for the projective deformation in the image, we build a grid with variable cells sizes. Using the inferred distance computed earlier, the z position on the grid is computed (namely, in which row the person stands); then, a cell on that row is decided by using the x position on the image plane of the center of the person's head. The last step in the construction of the 3D model is to add orientation information. The "bird view" model is then complete and features (x, z, o) coordinates, where o represents the estimated head orientation.

4. Experimental results

To evaluate our social group detector and head pose estimation algorithm we provide two publicly available datasets: EGO-GROUP and EGO-HPE datasets. EGO-GROUP[1] contains 10 videos, more than 2900 frames annotated with group compositions and 19 different subjects. Furthermore, 4 different scenarios are proposed in order to challenge our method in different situations: a laboratory setting with limited background clutter and fixed lighting conditions (Figure 2a), a coffee break scenario with very poor lighting and random backgrounds (Figure 2b), a festive moment with a crowded environment (Figure 2c) and an outdoor scenario (Figure 2d).

EGO-HPE dataset[2] is used for testing our head pose estimation method. This dataset presents videos with more than 3400 frames fully annotated with head pose. Being aimed to ego-vision applications, this dataset features significant background clutter, different illumination conditions, occasional poor image quality due to camera motion and both indoor and outdoor scenarios. We also use it to compare our technique against two state of the art approaches.

One of the more crucial and challenging components for our social group detection is the automatic extraction of the head pose of the subjects in the scene. A high error in such data creates a strong noise in the features used to cluster groups. In order to show the impact of the head pose estimation phase in our pipeline, we tested our egocentric head pose estimation method against other current state of the art methods over the EGO-HPE dataset. The first method we compared to is proposed by X. Zhu *et al*. [17]: 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. In order 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 1. This method, while being far from real-time, provides extremely precise head pose estimations even in ego-vision scenarios when it can overcome detection difficulties. The second method used in our comparison is [5]. This method provides real-time head pose estimations by using a regression forest trained with examples from 5 different head poses. The code provided by the authors does not yet perform automatic facial landmark estimation, hence we use the publicly available state of the art

[1] http://imagelab.ing.unimore.it/files/EGO-GROUP.zip
[2] http://imagelab.ing.unimore.it/files/EGO-HPE.zip

(a) Laboratory (b) Coffee (c) Party (d) Outdoor

Figure 2: Examples from the EGO-GROUP dataset. Each picture shows one of the different testing scenarios used by the method.

Table 1: Comparison of our head pose estimation and two state of the art methods on EGO-HPE dataset.

	Our Method	Zhu *et al.* [17]	Dantone *et al.* [5]
EGO-HPE1	**0.750**	0.685	0.418
EGO-HPE2	**0.670**	0.585	0.326
EGO-HPE3	**0.668**	0.315	0.330
EGO-HPE4	**0.821**	0.771	0.634

landmark estimator [15] and the performance of this method are strictly tied to the output of this step. Table 1 shows the results in terms of accuracy of this comparison: the developed head pose estimation method outperforms its state of the art counterparts if applied to egocentric video sequences while still working in real-time. Nevertheless, error occurs and thus the possibility of misclassified poses must be considered when training and testing the clustering algorithm.

When detecting social groups the choice of which data train onto is extremely crucial: in different social scenarios distances and poses can assume different significances. For this reason, in order to achieve good performances in a real world application training should be *context dependent*. However, the risk of overfitting is considerable: Table 2 shows the performances of our method applied to every scenario of the EGO-GROUP dataset by repeating the training over the first video from each scenario. Results obtained by training the method over the union of the training sets of each scenario are also displayed. To our knowledge, this is the first work that tackles with group partitioning in an egocentric video perspective, hence the lack of further comparisons with other approaches. In particular from this data can be seen how, for example, training the weights over the *outdoor* sequence outperforms training on the *coffee* setting when testing on the *coffee* itself, but performs rather worse on different scenarios. This is due to overfitting on a particular group dynamic present in both the training and the *coffee* videos, but absent from other sequences. In order to have an estimate of how different trainings perform, standard deviation over the absolute error can be computed. It emerges that *laboratory* setting is the more general training solution with an average error of 10.94 and a standard deviation of 1.14, while training over the *party* sequence, although it can achieve impeccable results over its own scenario and an average error of 11.16, presents a much higher deviation (8.66). Training over the set given by the union of each training set from the different scenarios results in a standard deviation of 7.84 over a mean error of 11.35, showing how this solution, while maintaining the overall error rates, does not provide a gain in generality. In the further experiment we will assume that the training has been done over the *laboratory* setting, which as described showed to be the most general and less likely to overfit on one feature rather than on another.

An important parameter of our group detection approach is the dimension of the clustering window: being able to change window size allows to adapt to different situations. The window size effectively regulates over how many frames calculate the groups, resulting in being much less noise-sensitive with bigger windows but less capable of capturing quick variations among the groups composition. On the other hand, a small window size allows to model even very small changes in groups but its performances are strictly tied to the amount of noise in the features, e.g. wrong pose estimations or an imprecise 3D reconstruction. In our experiments we show that a window size of 8 frames provides a good compromise between robustness to noise in the descriptor and fine grained response of our system. Figure 3 reports the results on EGO-GROUP of our method in terms of absolute error, evaluated with the MITRE loss function described in Section 2.2, varying window sizes. As the chart shows, results under different window sizes are tied to the amount of noise in the feature vectors. In particular, one can notice how the *party* sequence (red plot) does not benefit from increasing the window size: this is due to the good performance in head pose and distance estimations. Since there is very little noise to remove, the decay in accuracy is mainly caused by the loss of information caused by the excessively coarse grain in the group estimation. On the other hand, the *coffee* setting (blue plot) presents one of the most challenging scenarios for our head pose estimation method, thus the gain in performances increasing window sizes. However, by increasing it too much

Table 2: Comparison between training variations on our method. The table shows how different training choices can deeply impact on the performances: while the *laboratory* scenario presents a rather balanced training environment, a training set extracted from the *party* or the *coffee* scenarios can overfit on some features leading to very high performances when applied to videos with the very same situation and worse results if used on other data.

	Training: Laboratory			**Training: Coffee**			**Training: Party**			**Training: Outdoor**			**Training: All**		
Test scenario	**Error**	**Precision**	**Recall**	**Error**	**Precision**	**Recall**	**Error**	**Precision**	**Recall**	**Error**	**Precision**	**Recall**	**Error**	**Precision**	**Recall**
Coffee	11.55	82.99	97.17	11.69	79.17	100.00	18.90	69.44	100.00	6.75	92.62	94.06	6.50	88.80	99.46
Party	9.33	100.00	83.63	0.00	100.00	100.00	0.00	100.00	100.00	10.92	100.00	80.34	3.15	96.27	98.05
Laboratory	11.91	91.68	85.79	14.75	74.67	99.43	14.43	74.81	100.00	27.75	72.60	72.81	19.97	74.32	88.05
Outdoor	10.97	87.39	95.09	11.31	81.25	100.00	11.31	81.25	100.00	29.76	100.00	58.93	15.83	83.93	89.17

the loss of information overcomes the gain from the noise suppression and worsens the performances. In general, it can be noted how increasing the window size past 8 - 16 usually worsens the overall performances of the proposed method.

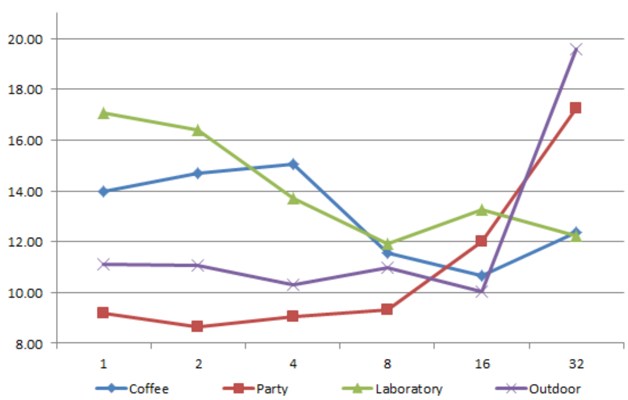

Figure 3: Comparison between absolute error results under various window sizes in our method.

5. Conclusion

In this paper we presented a novel method for estimating group compositions in ego-vision scenarios. We developed a head pose estimation technique designed for first person camera views and used it to effectively compute head pose of the subjects in the scene. Furthermore we estimated the 3D location of the people without the need of camera calibration. Using these information, we employ socially inspired features and the correlation clustering algorithm to partition the people in the scene into related groups. We tested our approach on two publicly available datasets we provide and show its validity in the challenging setting of egocentric camera views.

References

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004. 2, 3

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 4

[3] C. Cardie and K. Wagstaff. Noun Phrase Coreference as Clustering. 1999. 3

[4] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegaz, and V. Murino. Social interaction discovery by statistical analysis of f-formations. In *Proc. of BMVC*, 2011. 2

[5] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proc. of CVPR*, 2012. 4, 5

[6] A. Fathi, J. Hodgins, and J. Rehg. Social interactions: A first-person perspective. In *Proc. of CVPR*, 2012. 1

[7] T. Finley and T. Joachims. Supervised clustering with support vector machines. In *Proc. of ICML*, 2005. 3

[8] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117(10):1245–1256, 2012. 3

[9] H. Hung and B. Kröse. Detecting f-formations as dominant sets. In *Proc. of ICMI*, 2011. 2

[10] A. Kendon. *Studies in the behavior of social interaction*, volume 6. Humanities Press Intl, 1977. 1

[11] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *Proc. of CVPR*, 2013. 1

[12] H. S. Park, E. Jain, and Y. Sheikh. Predicting primary gaze behavior using social saliency fields. In *Proc. of ICCV*, 2013. 1

[13] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Proc. of CVPR*, 2012. 1

[14] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proc. of ICML*, 2004. 2

[15] M. Ui, V. Franc, and V. Hlav. Facial landmarks detector learned by the structured output svm. In *Proc. of VISAPP*. 2013. 5

[16] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *Proc. of Conf. on Message understanding*, 1995. 3

[17] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. *Proc. of CVPR*, 2012. 4, 5