This is the peer reviewd version of the followng article:

Understanding social relationships in egocentric vision / Alletto, S., Serra, G., Calderara, S., Cucchiara, R.. - In: PATTERN RECOGNITION. - ISSN 0031-3203. - ELETTRONICO. - 48:12(2015), pp. 4082-4096. [10.1016/j.patcog.2015.06.006]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

07/11/2026 15:29

(Article begins on next page)

Understanding Social Relationships in Egocentric Vision

Stefano Alletto[a], Giuseppe Serra[a,*], Simone Calderara[a], Rita Cucchiara[a]

[a]*DIEF, University of Modena and Reggio Emilia, Via Vignolese, 905 - 41125 Modena - Italy*

Abstract

The understanding of mutual people interaction is a key component for recognizing people social behavior, but it strongly relies on a personal point of view resulting difficult to be a-priori modeled. We propose the adoption of the unique first person perspective of head mounted cameras (ego-vision) to promptly detect people interaction in different social contexts. The proposal relies on a complete system that reliably extract people head pose combining landmarks and shape descriptors in a temporal smoothed HMM framework. Finally, interactions are detected through supervised clustering on mutual head orientation and people distances exploiting a structural learning framework that specifically adjusts the clustering measure according to a peculiar scenario. Our solution provides the flexibility to capture the interactions disregarding the number of individuals involved and their level of acquaintance in context with a variable degree of social involvement. The proposed system exhibits competitive performance over both publicly available ego-vision datasets and ad-hoc benchmarks built with real life situations.

Keywords: Egocentric Vision, Social Interactions, Group Detection, Video Analysis, Head Pose Estimation

1. Introduction

Social interactions are so natural that we rarely stop wondering who is interacting with whom or which people form a group and which do not. Nevertheless,

*Corresponding author.

Email addresses: stefano.alletto@unimore.it (Stefano Alletto), giuseppe.serra@unimore.it (Giuseppe Serra), simone.calderara@unimore.it (Simone Calderara), rita.cucchiara@unimore.it (Rita Cucchiara)

humans naturally do that neglecting that this task complexity increases when only visual cues are available. Different situations call for 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 is the case where we rarely exchange mutual gaze with people we are not interacting with, an important clue when trying to discern different social clusters.

Humans are inherently good at recognizing situations and understanding groups formation, but transferring this task to a fully automated system is still an open and challenging issue.

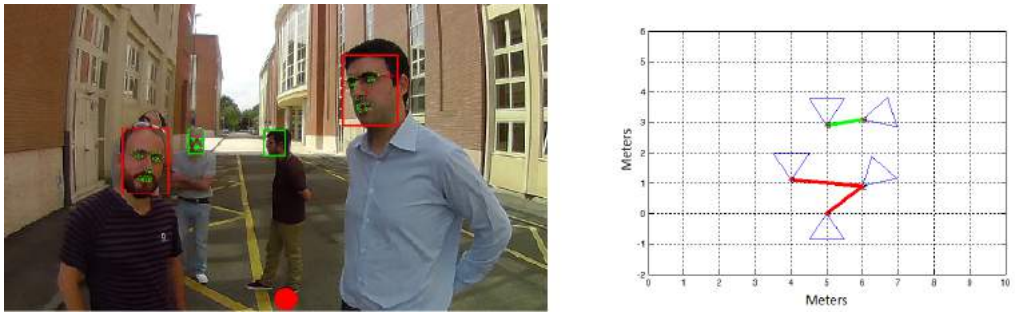
Recently initial works have started to address the task of social interaction analysis from the videosurveillance perspective [1, 2]. Fixed cameras (used in videosurveillance scenarios) lack the ability to immerse in the social environment, effectively losing an extremely significant portion of the information about what is happening. An egocentric video (ego-video) provides an insight in the social interaction, the recording is performed by a member of the group itself resulting in a completely new and inherently social perspective.

The recent spread of wearable cameras puts the research on the matter in a new and unique position. Often called first-person vision, to recall the needs of using wearable cameras for acquiring and processing the same visual stimuli that humans acquire and process. Indeed, ego-vision assumes the broader meaning of understanding what a person sees calling for similar learning, perception and reasoning paradigms of humans. This new approach carries exceptional benefits but it exposes several problems: being the camera tied to its user, it follows his or her movements and severe camera motion, steep lighting transitions, background clutter and severe occlusions occur; these situations often require new solutions in order to automatically process the video stream and extract information. Recently efforts this direction has benn done. Li *et al.* [**?**] proposed an pixel-level hand detection approach based on sparse feature and a set of Random Forests indexed by a global color histogram, which was shown to be robust to illumination changes. Lu and Grauman *et al.* [**?**] presented a method that produces a compact storyboard summary of the camera wearer's day that is driven by detecting the most important objects and people with which the camera wearer interacts.

We believe that egocentric vision (or ego-video) can provide an insight in the social interaction: the recording is performed by a member of the group itself resulting in a completely new and inherently social perspective.

In this paper we address the problem of partitioning people in a video sequence into socially related groups from an egocentric vision (from now on, ego-vision)

Figure 1: An example of our method's output. In the left image: eople 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 link among them represents the groups.



perspective. Human behavior is by no means random: when 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 organize orientations so as to naturally place the focus on the subjects of our interest. Distance between individuals and mutual orientations assume clear significance and must be interpreted according to the situation. *F-formation* theory [3] 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 [4, 5].

Hence, 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 relate. Instead of using the resulting information in a simple classification framework, we follow our idea that different situations call for different social interaction types and consequently orientation and distances assume different importance and meaning, depending on the situation. By this aim we learn social groups in a supervised correlation clustering framework. We present a novel approach for detecting social groups using a correlation clustering algorithm that exploits social features to truly capture the social clues inferred from human behavior.

Here, we provide our main contributions:

- the definition of a novel head pose estimation approach developed to cope with the challenges of the ego-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.

- the 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 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.

- an extensive evaluation of the results of our method on publicly available datasets, comparing it to several state of the art algorithms. We test each component of our framework and extensively discuss the results obtained in our experiments.

With regard to our previous conference work [6], the method we propose here presents several differences on its key aspects: the head pose estimation framework has been extended including landmarks information, improving the accuracy and robustness of the method. We also introduce a 3D distance estimation method based on a combination of landmarks and bounding box of the tracked head that does not rely on camera calibration. An extensive evaluation of several state of the art trackers applied to ego-vision is also presented and we perform larger analytic experiments in order to test our method in real and unconstrained ego-vision scenarios. Finally, we enrich the proposed dataset with new sequences.

Results are very promising and, while they highlight some open problems, they show a new way for computer vision to deal with the complexity of unconstrained scenarios such ego-vision and human social interactions.

2. Related Work

According to the main contributions of this paper, it is useful to describe the related work on its main areas.

Head pose estimation has been widely studied in computer vision. Already existing approaches can be roughly divided in two major categories, whether their aim is to estimate the head pose on still images or video sequences.

Among the most notable solutions for HPE in still images, Ma *et al.* [7] proposed a multi-view face representation based on Local Gabor Binary Patterns (LGBP) extracted on many subregions in order to obtain spatial information. Wu *et al.* [8] presented a two-level classification framework: the first level has the objective of deriving pose estimates with some uncertainty; the second level minimizes this uncertainty by analysing finer structural details captured by bunch graphs. While being very accurate on several publicly available datasets (e.g. [8] achieves a 90% accuracy over the Pointing 04), these works suffer significant performance losses when applied to less constrained environments like the ones typical of ego-vision. A recent successfull approach to head pose estimation on still images in the wild is [9], which models every facial landmark as a part and uses global mixtures to capture topological changes due to viewpoint. However this technique has high computational costs, resulting in up to 40 seconds per image, excessively demanding for the real-time requirements of an ego-vision based framework. A notable approach is proposed by Li *et al.* [10]: using 3D information, they exploit a physiognomical feature of the human head called *central profile*. The central profile is a 3D curve that divides the face and has the characteristic of having its points lying on the symmetry plane. Using Hough voting to identify the symmetry plane, Li *et al.* estimate the head pose using the normal vectors of the central profile which are parallel to the symmetry plane. Recently a comprehensive study that has summarized the head pose estimation methods and systems published over the past 14 years has been presented in [11].

Literature focusing on video streams for head pose estimation can be further divided in whether it uses any kind of 3D information or not. If such information can be used, a significant accuracy improvement can be achieved as in [12], which uses a stereo camera system to track a 3D face model in real-time, or [13] where the 3D model is recovered from different views of the head and then the pose estimation is done under the assumption that the camera stays still. Wearable devices used for ego-vision video capture, being aimed to more general purpose users and being on a mid-low price tier, usually lack the ability to capture 3D information; furthermore due to the unpredictable motion of both the camera and the object a robust 3D model is often hard to recover from multiple images. Rather then using a 3D model, Huang *et al.* [14] utilized a computational framework for robust detection, tracking, and pose estimation of faces captured by video arrays. To estimate face orientations they presented and compared respectively

two algorithms based on MLKalman filtering and multi-state CDHMM models. Orozco *et al.* [15] proposed a technique for head pose classification in crowded public space under poor lighting condition on low-resolution images using mean appearance templates and multi-class SVM.

In addition, there is a growing interest towards social interactions and human behavior. Social interactions are a key feature in improving tracking results in the work by Yan *et al.* [2]: by mapping pedestrians in the space and linking them, trajectories tracking can be refined by accounting for the repulsions and attractions that occur between people. They model links between pedestrians as *social forces* that can lead one individual towards another or drive him away. Multiple motion prediction models are then created and multiple trackers are instanced following the different models. A different approach to social interactions is proposed by Noceti *et al.* [1], who perform activity classification through recognizing groups of people socially engaged. Pedestrian pose is estimated and a graph of people is built. The groups are then detected using spectral kernel SVM.

All these methods, while providing interesting insights on social interactions, are based on the videosurveillance setting. This scenario presents some significant differences with the first person perspective of ego-vision to the point that completely different approaches may be needed to deal with this change in perspective.

Attempts using this new unique perspective are very few. In particular, the work by Fathi *et al.* [16] 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. However, this approach focuses of recognizing single interaction classes and does not take into account group dynamics and their social relations.

3. Understanding People Interactions

To deal with the complexity of understanding people interaction and detecting groups in real and unconstrained ego-vision scenarios, our method relies on several components (see Fig. 2). We start with an initial face detection and then track

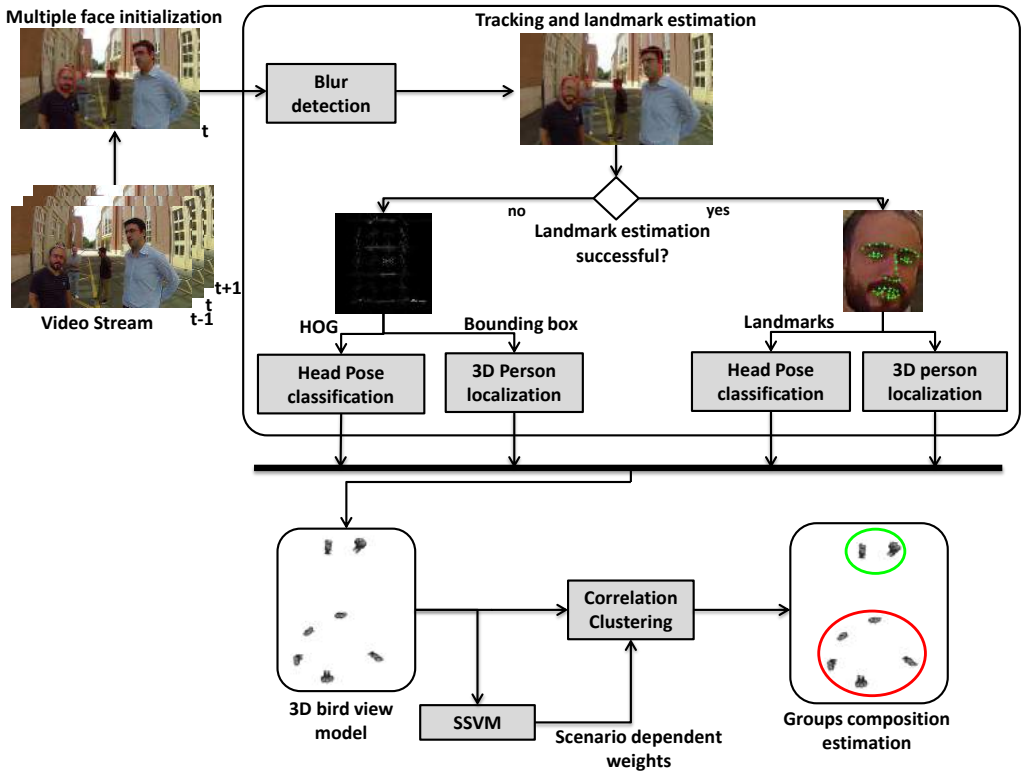


Figure 2: Schematization of the proposed approach.

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 group in different contexts based on the estimation of pairwise relations of their members.

3.1. Detection and Tracking

In order to be capable of working in unconstrained ego-vision scenarios, our method requires a robust tracking algorithm that can deal with steep camera motion leading to poor image quality and to frequent target losses. Furthermore, occlusion between members of different groups can often occur and must be treated accordingly. We evaluate several state of the art trackers [17] on egocentric videos in order to study their behavior w.r.t. the peculiarity of the ego-vision setting. The results of this comparison are presented in Section 5.

A preliminary step to the tracking of a subject is to understand whether tracking should be performed or not. In fact, a typical ego-vision characteristic is that

(a) Fast camera motion (b) Still camera

Figure 3: Two examples of frames from an ego-vision sequence showing the amount of blur in case of fast head motion or still camera.

the camera wearer can have very fast head motion, e.g. when he is looking around for something (see Fig. 3). This situation means, from a semantic point of view, that he hasn't focused his attention on some point of interest and hence that those frames are probably not worth to elaborate. From a more technical point of view, the high head motion can cause a significant blur in the video sequence resulting in an extremely low quality video. If not addressed properly, this situation can degrade the tracking at the point that it may not be possible to resume it when the attention of the subject stabilizes again.

To deal with this challenge typical of the ego-vision scenario, we evaluate at each frame the amount of blurriness and decide whether to proceed with the tracking or to skip it. The idea behind our approach is to compute the amount of gradient in the frame and to learn a threshold that discriminates a fast head movement due to the user looking around 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 I, according to a threshold θ_B:

$$Blur(I, \theta_B) = \sum_I \sqrt{\nabla S_x^2(I) + \nabla S_y^2(I)}, \tag{1}$$

where $\nabla S_x^2(I)$ and $\nabla S_y^2(I)$ are the x and y components of Sobel's gradient in the image 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, that can be done in real-time, effectively allows to remove those 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 employ the state of the art tracker TLD [18]. TLD framework features three main components: a *Tracker* which, under the assumption of a limited motion between consecutive frames, estimates the object's motion. This component of the framework is likely to fail if the object exits the camera field of view and it is not able to resume the tracking by itself. A *Detector* intervenes treating each frame independently and performs the detection localizing the appearances of the object which have been observed and learned in the past, recovering tracking after the *Tracker* fails. The *Learning* component observes the performance of both *Tracker* and *Detector*, estimates their error and generates training samples in order to avoid such errors in the future under the assumption that both *Tracker*, in terms of object loss, and *Detector*, in terms of false positives or false negatives, can fail. The tracking component of TLD is based on a Median-Flow tracker extended with failure detection: it represents the object with a bounding box and estimates displacement vectors of a number of points. The 50% most reliable points are then used to displace the bounding box between the two frames using median flow. If the target gets fully occluded or exits the camera field of view, being d_i the displacement of a single point of the Median-Flow tracker and d_m the median displacement, it will result in having the individual displacements scattered around the image and the residual of a single displacement $|d_i - d_m|$ will rapidly increase. If $median|d_i - d_m|$ is greater than a fixed threshold, the failure of the tracker can be decreed and the framework will rely on the *Detector* to resume it. At each frame, the bounding box resulting from the tracking phase is merged with the one output of the detection process and only if neither the *Tracker* nor the *Detector* return a bounding box the object is declared as non-visible.

3.2. Head Pose Estimation

To obtain a reliable estimation of the subject's head pose, we rely on two different techniques: facial landmarks and shape based head pose estimation.

Using the first approach, head pose can be accurately computed provided that the resolution of the face is high enough and that the yaw, pitch and roll angles of the head are not excessively steep. However when these conditions are not met, the landmark estimation process fails and hence the head pose cannot be computed. To render our method robust against such situations, we integrate the pose estimation based on the landmarks with an appearance based head pose estimation that uses HOG features and a classification framework composed of SVM followed by HMM. Our method effectively integrates these two different components achieving the ability to cope with the complexity of the ego-vision scenario

by applying each one of the two techniques when they can yield the best results.

The first component of our method that is used to estimate the head pose is based on facial landmarks: if these can be computed, head pose can be reliably inferred and no further processing is needed. To estimate facial landmarks, we employ the method by Smith *et al.* [19]. This allows to estimate a set of semantically significant landmarks $L = \{l_i = (x_i, y_i), i = 1, \ldots, N\}$, where (x_i, y_i) are the coordinates of the i-th landmark on the image plane and N is the total number of landmarks. In our experiments we fix $N = 49$, since it is the minimum number of points for a semantic face description [20]. The pose estimation results from the face alignment process done by applying the supervised gradient descent method, which minimizes the following function over $\Delta\mathbf{x}$

$$f(\mathbf{x_0} + \Delta\mathbf{x}) = \|\mathbf{h}(\lambda(\mathbf{x_0} + \Delta\mathbf{x})) - \phi_*\|_2^2, \tag{2}$$

where x_0 is the initial configuration of landmarks, $\lambda(\mathbf{x})$ 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. $\phi_* = \mathbf{h}(\lambda(\mathbf{x}_*))$ represents the SIFT descriptors computed over manually annotated landmarks in the image. Finally, the obtained pose is quantized over 5 classes, representing the intervals $[-90, -60)$, $[-60, -30)$, $[-30, 30]$, $(30, 60]$ and $(60, 90]$.

This approach, while providing extremely reliable head poses if the landmark set can be estimated, fails if applied to steep head poses or to lower resolution images, we combine it with a second approach based on the shape of the subject's head.

To estimate head pose using shape features, a preprocess step is taken before calculating the head descriptor: to effectively remove the large amount of noise caused by the background in the unconstrained scenario of ego-vision, we rely on segmentation. We use an adapted version of the GrabCut [21] algorithm: it aims at minimizing the energy function

$$E(\alpha, \mathbf{k}, \theta, \mathbf{z}) = U(\alpha, \mathbf{k}, \theta, \mathbf{z}) + V(\alpha, \mathbf{z}), \tag{3}$$

where N is the number of pixels, $\mathbf{z}$ is the image vector, α is the segmentation mask with $\alpha_i \in \{\pm 1\}$. $\mathbf{k}$, $k_n \in \{1, \ldots, K\}$ is the vector assigning each pixel to a unique GMM and θ is the set of parameters of the GMMs.

The data term $U(\cdot)$ is defined

$$U(\alpha, \mathbf{k}, \theta, \mathbf{z}) = \sum_n -log(p(z_n|\alpha_n, k_n, \theta)) - log(\pi(\alpha_n, k_n)). \tag{4}$$

where the Gaussian Mixture Models for either the foreground T_F or background T_B are hence defined as $p(z|\alpha_n, k_n, \theta_F)$ and $p(z|\alpha_n, k_n, \theta_B)$.

Our initial experiments showed that a segmentation step based on the bounding box resulting from the tracking phase yields poor results if applied to situations where no assumptions on the background model were possible. This occurs because in the tracked bounding box small portions of background pixels are included. 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 background by the algorithm and produce a noisy segmentation. To address this issue, after an initialization of both foreground and background regions T_F and T_B, we build the respective GMMs: these models represent the initial distribution of color and are used to assign a label α to each pixel. Exploiting the high frame rate of ego videos is possible to assume that only slight changes in the foreground and background mixtures will occur between two subsequent frames. This allows, at time t, to build a GMM_t based on GMM_{t-1} instead of reinitializing the models: pixels are assigned to foreground or background based on GMM_{t-1} and then the GMM for the current frame is computed. This is equivalent to soft assigning pixels that would end up in the T_U region, which is sensitive to noise.

Once that a precise segmentation of the head is obtained, the resulting image is converted to grayscale, resized to a fixed size of 100×100 in order to ensure invariance to scale and, eventually, histogram equalization is applied to it. A dense HOG descriptor is then computed over the resulting image using 64 cells and 16 bins per cell.

Given their potential to increase the overall performance of the classification step, feature normalization techniques have been applied to the resulting HOG descriptor. Using a Linear SVM that relies on dot-product, applying power normalization techniques shows to effectively increase the accuracy of our results. We apply the following function to our feature vectors:

$$f(\mathbf{x}) = sign(\mathbf{x})|\mathbf{x}|^{\alpha} \quad with \quad 0 < \alpha < 1. \tag{5}$$

Based on initial observations we fix $\alpha = 0.5$. By optimizing this value, the performance could slightly improve but it would lead to a data-dependent tuning, a situation in contrast with the highly diversified characteristics of unconstrained ego-vision scenarios. 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. When dealing with high dimensional feature vectors,

the linear SVM has proven competitive w.r.t its kernelized version while requiring less computational resources coping well with low tier ego-vision devices [22].

Typically, 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 chances tend not to occur when one subject is talking.

In order to enforce temporal coherence that derives from a video sequence, a stateful Hidden Markov Model technique is employed. Hidden Markov Models are temporal graphical models in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. 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 variable are the input images.

The probability distribution of $\mathbf{z}_t$ depends on the state of the previous latent variable $\mathbf{z}_{t-1}$ through a conditional distribution $p(\mathbf{z}_t|\mathbf{z}_{t-1})$, namely the transition probability distribution; while the conditional pdf that involves both observed and hidden variable is referred as the *emission function*, $p(\mathbf{o}_t|\mathbf{z}_t)$. In a discrete setting, with an enumerable number of states, the conditional distribution corresponds to a matrix denoted by $\mathbf{A}$, where the elements are transition probabilities among the states themselves. They are given by

$$A = \{a_{jk}, j, k = 1 \dots K\} \equiv p(z_{tk} = 1|z_{t1,j} = 1) \tag{6}$$

so that the matrix $\mathbf{A}$ has $K(K-1)$ independent parameters. During learning, out of the box techniques like the Baum Welch training algorithm can be used to train the Hidden Markov Model. Nevertheless, whenever applicable, transition probabilities among discrete states can be directly set by an expert in order to impose constraint on the possible transitions. In practice, we fixed the $\mathbf{A}$ values in order to encode the context of ego-vision videos. In particular, we set in the state transition matrix 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 not adjacent states[1]. This leads our approach to have continuous transitions

[1]To allow the reimplementation of our method, we report the values employed in the transition matrix: $A = \begin{bmatrix} 0.7 & 0.05 & 0.10 & 0.10 & 0.05 \\ 0.01 & 0.70 & 0.13 & 0.01 & 0.15 \\ 0.19 & 0.19 & 0.6 & 0.01 & 0.01 \\ 0.20 & 0.00 & 0.00 & 0.6 & 0.20 \\ 0.01 & 0.00 & 0.00 & 0.23 & 0.75 \end{bmatrix}$. The i, j indexes in the matrix refer, in order, to

between adjacent poses. Furthermore, this also allows the removal of most of the impulsive errors that are due to wrong segmentation or to the presence of a region of background in the calculation of the descriptor. This translates in a smooth transition among possible poses that is what conventionally happens during social interaction among people in ego-vision settings.

The joint probability distribution over both latent and observed variables results:

$$p(\mathbf{z}_t, \mathbf{o}_t) = p(\mathbf{z}_0) \prod_{t=1}^{T} p(\mathbf{o}_t|\mathbf{z}_t) p(\mathbf{z}_t|\mathbf{z}_{t-1}). \tag{7}$$

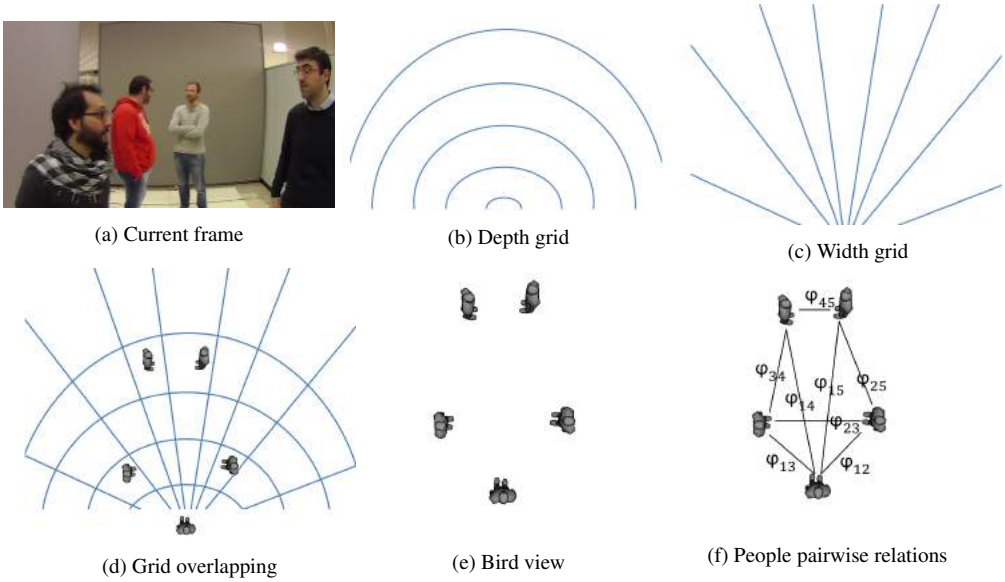
Our method combines the likelihood $p(\mathbf{z}_t|\mathbf{o}_t)$ of a measure $\mathbf{o}_t$ to belong to a pose $\mathbf{z}_t$ provided by the SVM classifier with the previous state $\mathbf{z}_{t-1}$ and the transition matrix $\mathbf{A}$ derived from the HMM, obtaining the predicted pose likelihood which is the final output.

In order to calibrate a confidence level to a probability in a SVM classifier, so it can be used as a observation for our HMM, we trained a set of Venn Predictors (VP) [23], 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 at estimating 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 set of multisets of S of length n. In the case of multi class 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:

$$p^{Y_j}(Y_k) = \frac{|\{(\mathbf{x}^*, y^*) \in \tau_{new} : y^* = Y_k\}|}{|\tau_{new}|} \tag{8}$$

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

the classes 0, 75, 45, -45, -75. Note that the matrix is not symmetric in order to compensate any bias on the dataset used to train the SVM classifier from which the class probability estimation is computed.



(a) Current frame (b) Depth grid (c) Width grid

(d) Grid overlapping (e) Bird view (f) People pairwise relations

Figure 4: Steps used in our distance estimation process.

it we get:

$$P_n = \{p^{Y_j} : Y_j \in \{Y_1, \ldots, Y_c\}\} \tag{9}$$

that is the well-calibrated set of multi probability prediction of the VP used as the emission function of Eq. 7.

3.3. 3D People localization

To provide the most general framework possible, we decide not to use any camera calibration technique in estimating the distance of a subject from the camera. The challenges posed by this decision are somehow 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. A depth measure which preserves the positional relations between individual suffices.

Relying on the assumption that all the heads in the image lay on a plane, 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 employ the facial landmarks computed in the head pose estimation phase. Being $N = |L|$, we build the feature vector

$$\mathbf{d} = \{d_i = \|l_i, l_{i+1}\|, i = 1, \ldots, N-1, l_i \in L\}, \tag{10}$$

where $\|\cdot\|$ is the standard euclidean distance. This feature vector is used in a Random Regression Forest [24] trained using the ground truth depth data obtained from a Kinect sensor. In order to minimize the impact on the distance of a wrong set of landmarks, we apply over a 100 frames window a Robust Local Regression smoothing based on the LOWESS method [25]. The distances vector is smoothed using the robust weights given by

$$w_i = \begin{cases} (1 - (r_i/\alpha MAD)^2)^2 & \text{if } |r_i| < \alpha MAD, \\ 0 & \text{if } r_i \geq \alpha MAD \end{cases}, \tag{11}$$

where r_i is the residual of the *i*-th data point produced by the local regression, MAD is the median absolute deviation of the residuals $MAD = medial(|r|)$ and α is a constant value fixed to 6.

This technique provides a good estimation of the distance of a face from the camera, coping well with the non-linearity of the problem and with the topological deformations that are due to changes in pose. Techniques simpler than using a regression forest, e.g. thresholding the ratios between landmarks, have shown extremely poor performances in our preliminary experiments due to the strong non-linearity of the ratios between landmarks under different poses. In fact, while on frontal faces changes in landmarks ratios directly reflect changes in distance from the camera, if the subject is seen by a different angle the relationship between landmarks and distance is much more complex.

In the case that facial landmarks estimation fails, we compute the distance from the camera by using the tracked bounding box as input for a Random Regression Forest properly trained using this feature. This result in a slightly less accurate estimation but makes our method robust to the failure in the landmark extraction process.

In order to estimate the position of a person accounting for the projective deformation in the image, we build a grid with variable cells sizes. The distance allows to locate the subject with one degree of freedom (x) (Fig. 4b): the semicircle in which the person stands is decided based on the distance computed early on, 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. 4c), it is now possible to place the person with one further degree of freedom z. By overlapping the two grids (Fig. 4d) the cell in which the person stands can be decided and the bird's view model can finally be created (Fig 4e).

Each people is now represented by its position in the 3D space (x, z, o), where

o represents the estimated head orientation and a graph connecting people is created (Fig. 4f). Each edge connecting two people p and q has a weight ϕ_{pq} which is the feature vector that includes mutual distance and orientations.

4. Social Group Detection

Head pose and 3D people information can be used to deal with the group detection problem, introducing the concept of relationship between two individuals. Given two people $\mathbf{p}$ and $\mathbf{q}$, 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 $\phi_{pq} = (d, o_{pq}, o_{qp})$. Note that distance d is by definition symmetric, while rotations o_{pq} and o_{qp} are not, thus the need of two orientation features instead of one. A practical example is given by the situation where two people are facing each other, $o_{pq} = o_{qp} = 0$: in this case both orientations are the same; on the contrary if the two subjects have the same orientation resulting in $\mathbf{p}$ looking at $\mathbf{q}$'s back, they will have $o_{pq} = 0$ and $o_{qp} = \pi$, hence the need of two separate features.

It can often be hard to practically fix this definition of relationship and use it independently from the scenario, due to the human characteristic of forming social groups in very different ways according to the scenario they are in. Sometimes people being in the same group is given away by 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, but still they form a group. In any case, it clearly emerges the need for an algorithm capable of understanding different social situations, effectively learning how to treat distance and orientation features depending on the context.

4.1. Correlation Clustering via Structural SVM

To partition social groups based on the the pairwise relations of their members we apply the correlation clustering algorithm [26]. In particular, given a set of people $\mathbf{x}$ in the video sequence, we model their pairwise relations with an affinity matrix W, where for $W_{\mathbf{pq}} > 0$ two people $\mathbf{p}$ and $\mathbf{q}$ 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 $\mathbf{y}$ of a set of people $\mathbf{x}$ is then the partition that maximize the sum of affinities for item pairs in the same cluster:

$$\arg\max_{\mathbf{y}} \sum_{y \in \mathbf{y}} \sum_{r \neq t \in y} W_{\mathbf{rt}}. \tag{12}$$

Here, the affinity between two people $\mathbf{p}$ and $\mathbf{q}$, W_{pq} is modeled as a linear combination of the pairwise features of orientation and distance over a temporal window. This temporal window effectively determines how many frames are used to compute the current groups, capturing variations among the groups composition and maintaining robustness to noise. In order to obtain the best way to partition people into socially related groups in the given social situation, 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, it can be noticed that the output cannot be modeled by a single valued function, since a graph describing connections between members suits better the social dimension of the group interaction. This leads to an inherently structured output that requires to be treated accordingly. Structural SVM [27] offers a generalized framework to learn structured outputs by solving a loss augmented problem. 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

$$f(\mathbf{x}) \;=\; \arg\max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x}, \mathbf{y}, \mathbf{w}) \tag{13}$$

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. 12, the compatibility of an input-output pair can be defined as

$$F(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T \sum_{y\in\mathbf{y}} \sum_{r\neq t\in y} \phi_{pq} \tag{14}$$

where ϕ_{pq} is the pairwise feature vector of elements p and q. This 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 [27]:

$$\begin{aligned} \min_{\mathbf{w},\xi} \quad & \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \forall i : \xi_i \geq 0, \\ & \forall i, \forall \mathbf{y} \in \mathcal{Y}\backslash\mathbf{y}_i : \mathbf{w}^T \delta\Psi_i(\mathbf{y}) \geq \Delta(\mathbf{y}, \mathbf{y}_i) - \xi_i. \end{aligned} \tag{15}$$

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.

The quadratic program in Eq. 15 introduces a constraint for every possible wrong clustering of the set. Unfortunately, this results in a number of wrong clusterings that scales more than exponentially with the number of items. As performance is a sensitive aspect of each ego-vision application, approximated optimization schemes have to be considered. In particular, we rely on the cutting plane algorithm in which we start with no constraints, and iteratively find the most violated one:

$$\hat{\mathbf{y}}_i = \arg\max_{\mathbf{y}} \Delta(\mathbf{y}_i, \mathbf{y}) - \mathbf{w}^T \, \delta\Psi_i(\mathbf{y}) \tag{16}$$

and re-optimize until convergence. Finding the most violated constraint requires to solve the correlation clustering problem, which we know to be NP-hard [26]. Finley *et al.* [28] propose a greedy approximation algorithm which works by initially considering each person in its own separate cluster, then iteratively merging the two clusters whose members have the highest affinity.

One important aspect of this supervised correlation clustering is that there is no need to know beforehand how many groups are 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.

4.2. Loss function

How such algorithm can be effective its learning phase strictly depends on the choice of the loss function since it has the power to force or relax input margins.

The problem of clustering socially engaged people is in many ways similar to the noun-coreference problem [29] 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 loss function [30] has been identified as a suitable scoring measure. The MITRE loss, 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.

Consider the 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 [29].

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

$$\mathcal{R}_Q = \frac{\sum_{i=1}^{n} |Q_i| - |p(Q_i)|}{\sum_{i=1}^{n} |Q_i| - 1} \tag{17}$$

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

$$\Delta_M = 1 - F_1 \tag{18}$$

where F_1 is the standard F-score.

5. Experimental results

To evaluate our social group detector and each of the main components (head pose estimation algorithm, trackers and 3D people localization approach) we provide two publicly available datasets: EGO-HPE datasets and EGO-GROUP.

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 poses. 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.

EGO-GROUP[3] contains 18 videos, more than 10000 frames annotated with group compositions and 23 different subjects. Furthermore, 5 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 5a), a coffee break scenario with very poor lighting and random backgrounds (Figure 5b), a conference room setting where people movement and positioning is tied to

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

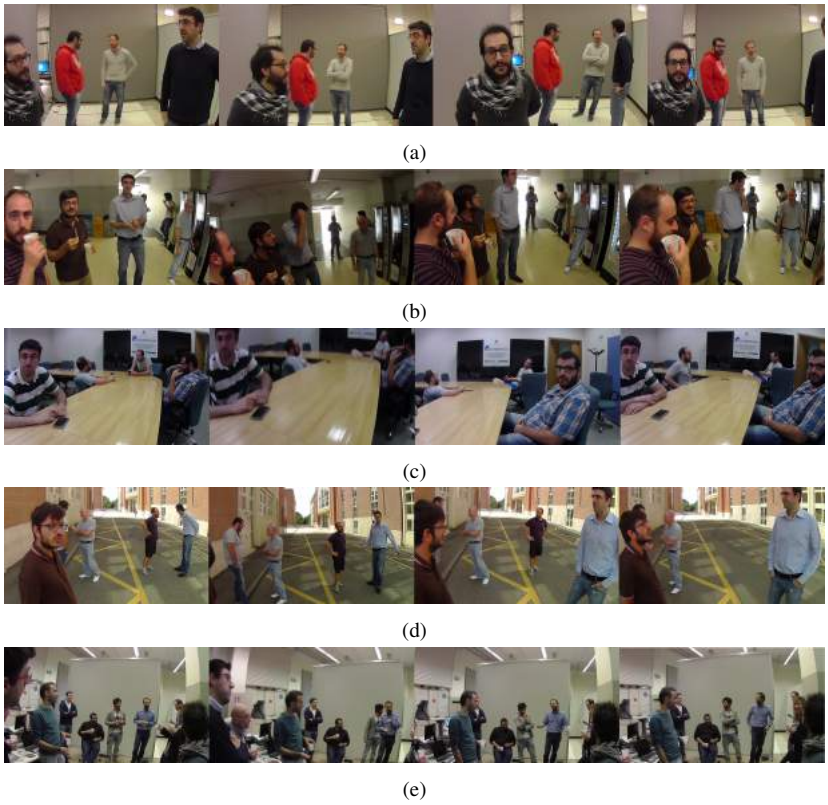


Figure 5: Example sequences from the EGO-GROUP dataset.

seats (Figure 5c), an outdoor scenario (Figure 5d) and a festive moment with a crowded environment (Figure 5e).

In both datasets, the videos are acquired using a Panasonic HX-A100 wearable camera, which features a head mounted camera capable of recording at 30 frames per second with a resolution of 1920×1080. The videos are subsampled to a 960×540 resolution and 15 fps, which reduces processing time while providing the same performance.

5.1. Head Pose Estimation

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.

Our method for estimating the head pose is based on the merging of two separate components: landmarks and HOG-based pose classification. Both approaches

Table 1: Comparison between different approaches evaluated. Results are proposed in terms of accuracy, which in case of Power-Normalized HOG (HOG+PN) and Power-Normalized HOG + HMM (HOG+PN+HMM) is the ratio between correct classifications and total samples. Landmarks results, providing a continue pose value, are quantized to the nearest class and then accuracy is computed with the same metric.

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**

have strong-sides and down-sides: using facial landmarks can be extremely accurate and fast, but it requires them to be successfully computed which can prevent the system to work under steep head poses or low resolutions. While steep profile poses (e.g. ± 90) can be difficult to classify using landmarks, human physiognomy makes it a task that can be performed with more success using shape features like HOG. The HOG descriptor is also much less sensitive to scale, which allows to perform the head pose estimation even of those subjects far from the person wearing the camera. In this scenario, the training of the SVM classifier has been performed using the 80% of the dataset, while the remaining 20% has been used for testing. The results are averaged over five independent runs.

Table 1 provides a comparison between the different approaches we evaluated, showing how the HOG and the landmark based approaches when combined together can achieve performance that none of them could have singularly achieved.

In order to show how ego-vision 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 current state of the art methods over the EGO-HPE dataset. The first method we compared to is proposed by X. Zhu *et al.* [9]: 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 [31]. This method provides real-time head pose estimations by using facial landmark features and a regression forest trained

Table 2: Comparison of our head pose estimation and three state of the art methods on EGO-HPE dataset.

	Our Method	**Zhu *et al.* [9]**	**Dantone *et al.* [31]**	**Xiong *et al.* [19]**
EGO-HPE1	**0.784**	0.685	0.418	0.537
EGO-HPE2	**0.727**	0.585	0.326	0.685
EGO-HPE3	**0.635**	0.315	0.330	0.401
EGO-HPE4	**0.821**	0.771	0.634	0.704

with examples from 5 different head poses. A third comparison is made against the method *Intraface* [19], which performs head pose estimation using landmarks and has been employed in our framework. As discussed above, this method results in a precise estimation on frontal images but can fail under steep orientations. Table 2 shows the results in terms of accuracy of this comparison.

5.2. Tracking Evaluation

In order to employ a current state of the art tracker in our method, we performed an evaluation comparing four state of the art trackers and two baselines. The trackers tested are: STR [32], HBT [33], TLD [18], FRT [34]; for a comprehensive tracking review please refer to [17]. A color histogram based nearest neighbor baseline tracker (NN) and a normalized cross correlation tracker (NCC) have also been included in our experiments. Provided that all four state of the art trackers perform well on normal scenarios, we tested them over 8 videos we manually annotated with target's bounding boxes, extracted from our EGO-GROUP dataset. These videos feature some of the main problems typical of egocentric videos.

The main challenges a tracker has to deal with when applied to first person video sequences are the head motion causing blur and quickly moving the target out of the scene, recurring occlusions, changes in lighting conditions and in scale. One of the more problematic aspects is ego-motion: Fig. 6c shows the performance of the trackers over a video rich of ego-motion. It can be seen how, in this scenario, two main features come in play: loss detection and model adaptability. STR, while having an adaptive model, lacks the ability to detect the loss of target resulting in being unable to cope with the fast movements of the object in and out of the camera field of view. Without loss detection, it results in adapting the model to a portion of background effectively degrading it. HBT cannot recover from a loss resulting in being unable to recover after the initial fast camera motion. It

emerges that, if challenged with fast ego-motion, simpler tracking by detection approaches (NN, NCC) can outperform more complex tracking methods. TLD, performing both tracking and detection, results in being robust to such situations.

A major issue in ego-vision is the severe camera motion, where the camera can abruptly get closer to the subject or change its perspective. Fig. 6d shows an example of such a situation: around frame 50 the target starts moving away from the person wearing the camera and it is where most of the trackers fail. TLD is the only one that can perform scale adaptivity and as the figure shows it is the only one, in the very end of the sequence, that can resume tracking with an high overlap percentage. FRT, HBT and STR can also resume tracking but they have not adapted to the new scale, resulting in a very low overlap due to the low ratio of intersection and union of the bounding boxes.

The results of this analysis, which are summarized in the survival curve plot of Fig. 7, combined with it being the fastest among the considered trackers, led us to employ TLD in our framework. White it can be a useful tool, it clearly emerges that the current tracking state of the art lacks the ability to fully cope with the complexity of egocentric videos. All the analyzed trackers present on some degree a weakness that leads them to failure when facing a particular challenge.

To somehow mitigate the impact of the ego-vision setting on the tracking performances, we introduced a preliminary step of blur detection that effectively removes the frames where the fast head motion causes the failure of the tracking process. Fig. 8 shows a comparison between tracking with TLD with and without the blur detection phase. It can be noticed how the increase in performances strictly depends on the amount of blurriness caused by head motion contained in the video. For example, in the first video a significant amount of blurred frames can be removed effectively preventing the TLD model to degrade, resulting in an increase in performance of 22% going from 0.421 to 0.514. On the other hand, due to them being more still, the blur detector do not remove any frame in videos 4 and 6 resulting in the same performance.

5.3. Distance Estimation

To evaluate our approach in distance estimation, we compare it to two different alternatives. Using the same regression architecture, two commonly employed approaches involve using the bounding box of the head or the segmented area of the face as features (our baseline). In all cases, results are obtained applying a 80-20 ratio between training of the regressor and testing and averaging five different runs. Table 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 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 and backprojection approach similar to the one used in HBT in order to robustly estimate the area of the person's face. The results of this comparison show that relying on biological features like the ratio between facial landmarks can greatly improve results against less complex spacial features.

Aiming to improving our results, we apply to our distances sequence a smoothing filter. As Table 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 LOESS and RLOESS methods the span has been set to 10% of the data.

5.4. Groups estimation

Table 4: 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. All tests have been performed using a window size of 8 frames.

	Training: Laboratory			**Training: Coffee**			**Training: Party**		
Test scenario	**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

	Training: Outdoor			**Training: Conference**			**Training: All**		
Test scenario	**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

Eventually, 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 4 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. In particular, each column provides results of the method obtained training the SSVM classifier using the first video of the setting referred by the column, e.g. ''Training: Laboratory" indicates that the method has been trained using only the first video of the laboratory setting. ''Training: All" uses a subset of windows randomly extracted from the first video of each sequence. Note that while the subset is random, the amount of windows per setting is fixed to 20% of the total in order to avoid overfititng on a particular scenario. To our knowledge, this is the first work that tackles with group partitioning in an ego-centric 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 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 a context dependent training is needed to adapt to how humans change their behavior based on the situation.

To further highlight the need for a training phase capable of learning how to treat each feature, we show the results of the clustering without performing training. This is done by fixing all the feature weights to the same value resulting in the algorithm to equally treat distance and orientations. Table 5 provides the results of this comparison: it can be noticed how without treating each feature

Table 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

with its own significance the algorithm often ends up placing every subject in the same 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 fact, group estimation results have been computed using the precision and recall metrics based on the MITRE loss described in Equation 17. This allows us to compute precision and recall in a way that inherently deals with the problems of a wrong number of clusters or wrong partitions. MITRE loss bases its results on links, which means that splitting in two the same group would lead to an error on 1 link in terms of a missing link (penalizing recall), while merging two separate groups would results an error on 1 link impacting on the precision.

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 to 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 9 reports the results on EGO-GROUP of our method in terms of absolute error, evaluated with the MITRE loss function described in Section 4.2, varying window sizes. As the plot shows, results changing window sizes change depending on the amount of noise in the features used to compute the groups.

In particular, it can be noticed 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 observed at windows size 32 is mainly caused by the loss of information caused by the excessively coarse grain in the group estimation. On the other hand, the *coffee* or *laboratory* settings (blue and green plots) presents some difficulties in the head pose estimation, thus the gain in performances increasing window sizes. However, by increasing it too much 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.

To further evaluate our approach we discuss how the clustering weights vary in different scenarios. Figure 10 shows the comparison between the different components of the weight vectors. As can be noticed, performing the training over different scenarios yield significantly different results. For example, clustering a sequence in the 4th 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.

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 assigns a negative weight to them.

6. Conclusion

In this paper we have presented a novel approach for estimating the group composition in ego-vision settings. We provided a head pose estimation method designed for this scenario that relies on two different features in order to deal with the complexity of the task, resulting robust to steep head poses, low resolutions and background clutter. We provided a 3D people localization method that do not rely in camera calibration, a process that with the widespread diffusion of

wearable devices would have caused a loss in generality. Following our idea that different social situations cause different behaviors in humans, which calls for different weights in the social features that must be used to estimate the group, we used Structural SVM to learn how to treat the distance and pose information. This results in a method capable of adapting to the complexity of human social interactions and allows for the use of a Correlation Clustering algorithm to predict group compositions. Our experiments show promising results in the group estimation task testing the most significant aspects of our algorithm.

We adapted the TLD tracker to use our blur detection algorithm in order to improve its performances by removing the frames where the fast head motion caused the image quality to drop. While this process helps in increasing the robustness of the tracking process, some considerations about tracking in ego-vision could be made. From our experiments, where we compared several state of the art trackers over first-person video sequences, clearly emerged how trackers designed to work with videos recorded from still cameras face some difficulties when applied to the unconstrained scenarios of ego-vision. Target occlusion or its moving out of the camera field of view can often occur and some sort of loss detection is needed to cope with this situation. Many trackers do not provide this feature yet, resulting in having much lower performances than expected. Due to the importance of the tracking step, we feel the need for a tracker solely designed with the ego-vision paradigm in mind. This would greatly help to tackle with further challenges that require a robust tracking process such as action and object recognition or social interactions analysis.

References

[1] N. Noceti, F. Odone, Humans in groups: The importance of contextual information for understanding collective activities, Pattern Recognition. 2, 6

[2] X. Yan, I. Kakadiaris, S. Shah, Modeling local behavior for predicting social interactions towards human tracking, Pattern Recognition 47 (4) (2014) 1626 – 1641. 2, 6

[3] A. Kendon, Studies in the behavior of social interaction, Vol. 6, Humanities Press Intl, 1977. 3

[4] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegaz, V. Murino, Social interaction discovery by statistical analysis of f-formations., in: BMVC, 2011, pp. 1–12. 3

[5] H. Hung, B. Kröse, Detecting f-formations as dominant sets, in: Proc. of ICMI, 2011. 3

[6] S. Alletto, G. Serra, S. Calderara, F. Solera, R. Cucchiara, From ego to nos-vision: Detecting social relationships in first-person views, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 580–585. 4

[7] B. Ma, W. Zhang, X. C. S. Shan, W. Gao, Robust head pose estimation using lgbp, Proc. of ICPR. 5

[8] J. P. J. Wu, D. Putthividhya, D. Norgaard, M. Trivedi, A two-level pose estimation framework using majority voting of gabor wavelets and bunch graph analysis, Proc. ICPR Workshop Visual Observation of Deictic Gestures. 5

[9] X. Zhu, D. Ramanan, Face detection, pose estimation and landmark localization in the wild, Proc. of CVPR. 5, 21, 22

[10] D. Li, W. Pedrycz, A central profile-based 3d face pose estimation, Pattern Recognition 47 (2) (2014) 525–534. 5

[11] E. Murphy-Chutorian, M. M. Trivedi, Head pose estimation in computer vision: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (4) (2008) 607–626. 5

[12] R. Newman, Y. Matsumoto, S. Rougeaux, A. Zelinsky, Real-time stereo tracking for head pose and gaze estimation, Proc. of Automatic Face and Gesture Recognition. 5

[13] S. Ohayon, E. Rivlin, Robust 3d head tracking using camera pose estimation, Proc. of ICPR. 5

[14] K. Huang, M. Trivedi, Robust real-time detection, tracking and pose estimation of faces in video streams, Proc. of ICPR. 5

[15] J. Orozco, S. Gong, T. Xiang, Head pose classification in crowded scenes, in: Proc. of BMVC, 2009. 6

[16] A. Fathi, J. Hodgins, J. Rehg, Social interactions: A first-person perspective, in: Proc. of CVPR, 2012. 6

[17] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: An experimental survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 99 (PrePrints) (2013) 1. 7, 22

[18] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (7) (2012) 1409–1422. 9, 22

[19] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, in: Proc. of CVPR, 2013. 10, 22

[20] B. M. Smith, J. Brandt, Z. Lin, L. Zhang, Nonparametric context modeling of local appearance for pose-and expression-robust facial landmark localization, in: Proc. of CVPR, 2014. 10

[21] C. Rother, V. Kolmogorov, A. Blake, "grabcut": interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics 23 (3) (2004) 309–314. 10

[22] Y. Singer, N. Srebro, Pegasos: Primal estimated sub-gradient solver for svm, in: Proc. of ICML, 2007. 12

[23] A. Lambrou, H. Papadopoulos, I. Nouretdinov, A. 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. 13

[24] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32. 15

[25] W. S. Cleveland, Robust locally weighted regression and smoothing scatterplots, Journal of the American Statistical Association 74 (368) (1979) 829–836. 15

[26] N. Bansal, A. Blum, S. Chawla, Correlation clustering, Machine Learning 56 (2004) 89–113. 16, 18

[27] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, in: Proc. of ICML, 2004. 17

[28] T. Finley, T. Joachims, Supervised clustering with support vector machines, in: Proc. of ICML, 2005. 18

[29] C. Cardie, K. Wagstaff, Noun Phrase Coreference as Clustering. 18, 19

[30] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, L. Hirschman, A model-theoretic coreference scoring scheme, in: Proc. of Conf. on Message understanding, 1995. 18

[31] M. Dantone, J. Gall, G. Fanelli, L. Van Gool, Real-time facial feature detection using conditional regression forests, in: Proc. of CVPR, 2012. 21, 22

[32] S. Hare, A. Saffari, P. H. Torr, Struck: Structured output tracking with kernels, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 263–270. 22

[33] M. Godec, P. M. Roth, H. Bischof, Hough-based tracking of non-rigid objects, Computer Vision and Image Understanding 117 (10) (2012) 1245–1256. 22

[34] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 1, IEEE, 2006, pp. 798–805. 22

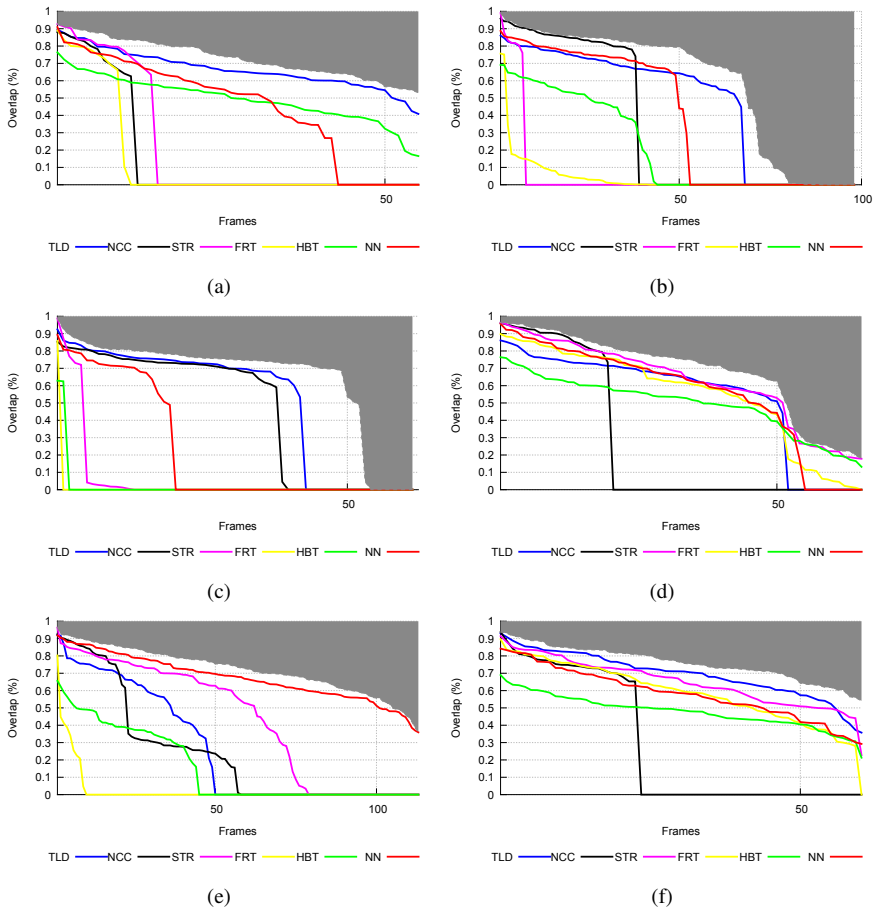


Figure 6: Overlap percentage between predicted bounding box and ground truth in our tracking evaluation. The plots represent data in the form of a survival curve, meaning that if a tracker scores a 0.5 overlap at frame 50, its overlap is ≥ 0.5 in at least 50 frames. The videos features the following settings and main challenges:(a) indoor, crowded sequence; (b) outdoor, low camera motion; (c) indoor, high camera motion; (d) indoor, controlled environment; (e) indoor, first person motion; (f) indoor, poor lighting. The frame number refers to the frame which had its ground truth for the subject's bounding box annotated, which is a frame out of 5. Best viewed in colors.

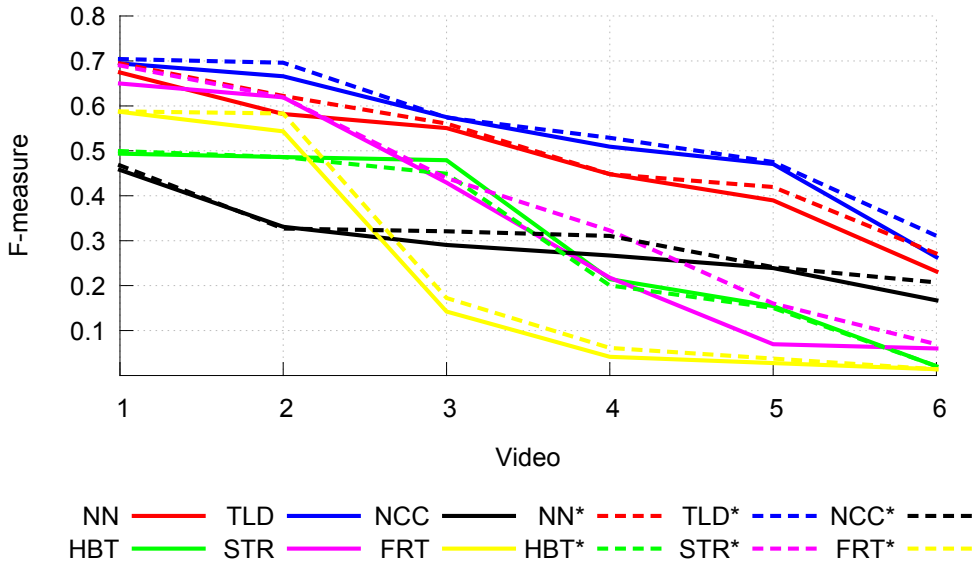


Figure 7: Survival curve of our tracking evaluation. The plot is obtained by sorting the F-measure results for each tracker: it shows that, for example, taken $x = 3$, TLD performs with a f-measure greater then 0.57 in at least 3 videos. The dashed plots refer to trackers employing the blur removal technique described in our method, also indicated by the * symbol in the legend. The solid line shows tracking results without any tampering, obtained using the code provided by the authors.

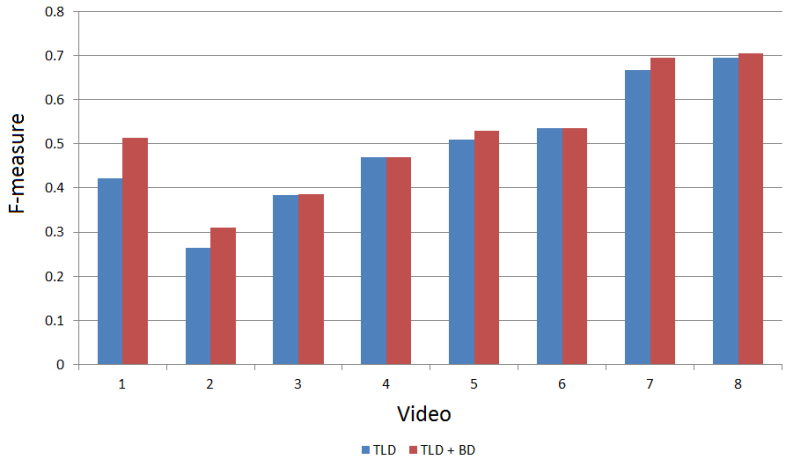


Figure 8: Plot showing the F-measure results of TLD and TLD with blur detection and removal (TLD + BD)

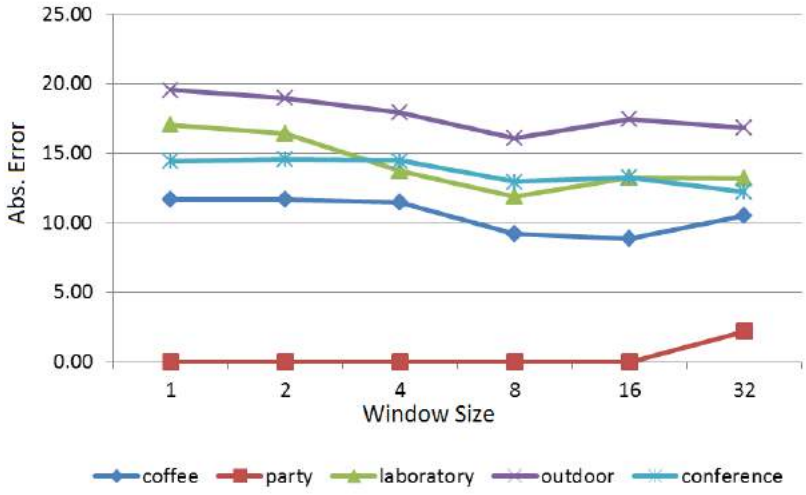


Figure 9: Comparison between absolute error results under various window sizes in our method.

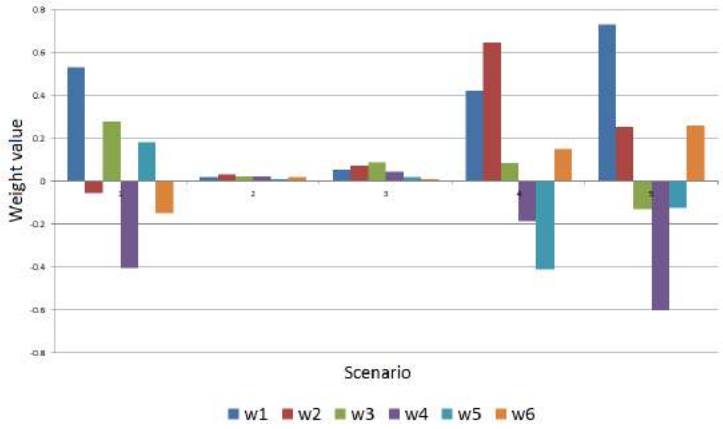


Figure 10: Weights values in the 5 different training scenarios. Scenarios are *1) laboratory, 2) party, 3) conference, 4) coffee, 5) outdoor.*

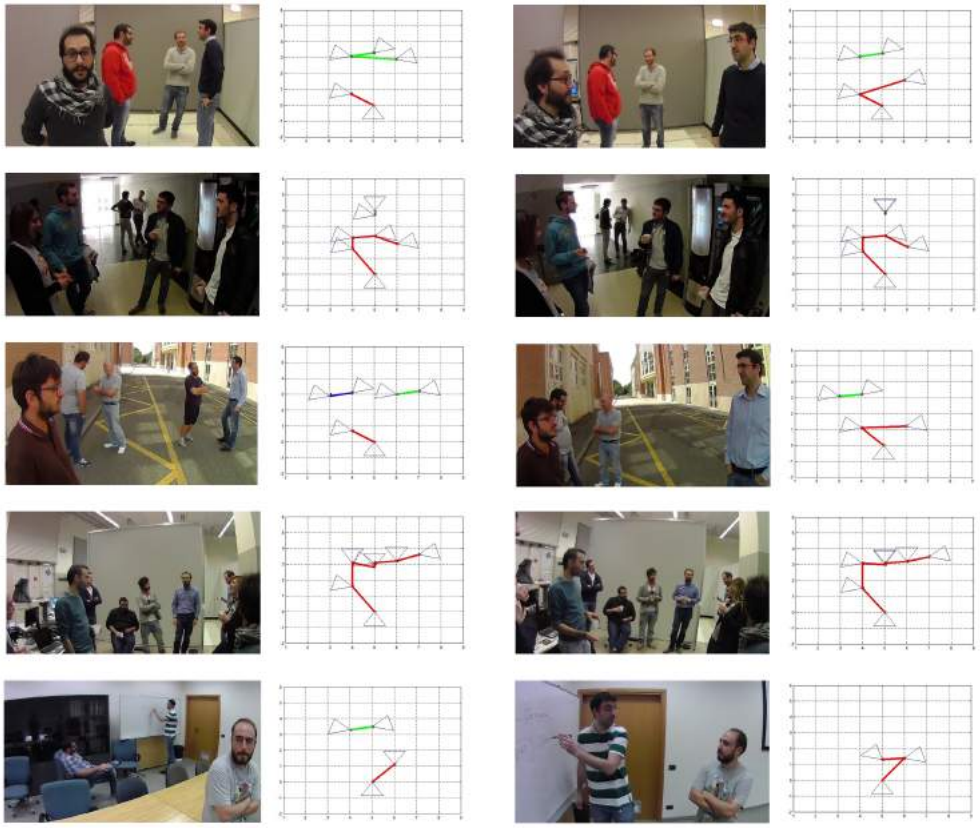


Figure 11: Examples of the results of our method. Different groups are shown by different link colors.