"Here's looking at you, kid"

"Here's looking at you, kid."
Detecting people looking at each other in videos.

Manuel J. Marín-Jiménez
mjmarin@uco.es
Dept. of C.S. and Numerical Analysis
University of Cordoba

Andrew Zisserman
az@robots.ox.ac.uk
Dept. of Engineering Science
University of Oxford

Vittorio Ferrari
ferrari@vision.ee.ethz.ch
Dept. ITET
ETH Zurich

Abstract

The objective of this work is to determine if people are interacting in TV video by detecting whether they are looking at each other or not. We determine both the temporal period of the interaction and also spatially localize the relevant people. We make the following three contributions: *(i)* head pose estimation in unconstrained scenarios (TV video) using Gaussian Process regression; *(ii)* propose and evaluate several methods for assessing whether and when pairs of people are looking at each other in a video shot; and *(iii)* introduce new ground truth annotation for this task, extending the TV Human Interactions Dataset [22]. The peformance of the methods is evaluated on this dataset, which consists of 300 video clips extracted from TV shows. Despite the variety and difficulty of this video material, our best method obtains an average precision of 86.2%.

1 Introduction

If you read any book on film editing or listen to a director's commentary on a DVD, then what emerges again and again is the importance of eyelines. Standard cinematography practice is to first establish which characters are looking at each other using a medium or wide shot, and then edit subsequent close-up shots so that the eyelines match the point of view of the characters. This is the basis of the well known 180^o rule in editing.

The objective of this paper is to determine whether eyelines match between characters within a shot – and hence understand which of the characters are interacting. The importance of the eyeline is illustrated by the three examples of fig. 1 – one giving rise to arguably the most famous quote from *Casablanca*, and another being the essence of the humour at that point in an episode of *Fawlty Towers*. Our target application is this type of edited TV video and films. It is very challenging material as there is a wide range of human actors, camera viewpoints and ever present background clutter. Determining whether characters are interacting using their eyelines is another step towards a fuller video understanding, and complements recent work on automatic character identification [7, 11, 27], human pose estimation [1, 6, 9, 13, 14, 24], human action recognition [16, 17, 18], and specific interaction recognition [22] (e.g. hugging, shaking hands). Putting interactions together with previous

Figure 1: **Are they looking at each other?** Answering this question enables richer video analysis, and retrieval based on where actors interact. From left to right: *Friends, Casablanca, Fawlty Towers*.

character identification work, it now becomes possible to retrieve shots where two particular actors interact, rather than just shots where the actors are present in the same scene.

In order to determine if two people are looking at each other, it is necessary to detect their head and estimate their head pose. There are two main strands in previous work: *2D* approaches, where detectors are built for several aspects of the head (such as frontal and profile [27]) or the pose is classified into discrete viewpoints [9, 28], or regressed [20]. The alternative are *3D* approaches, where a 3D model is fitted to the image and hence the pose determined [5, 10]. A survey of head pose estimation is given in [19].

In this work, we start by detecting human heads in video shots and grouping them over time into tracks, each corresponding to a different person (sec. 2). Next, we estimates the *pitch* and *yaw* angles for each head detection (sec. 3). For this, we propose a 2D approach and train a Gaussian Process regressor [23] to estimate the head pitch and yaw directly from the image patch within a detection window using publicly available datasets. In the third step, we explore three methods to determine if two people (tracks) are Looking At Each Other (*LAEO*, sec. 4). Two people are LAEO if there is eye contact between them. We start with a simple 2D analysis, based on the intersection of gaze areas in 2D defined by the sign of the estimated yaw angles (sec. 4.1). In a more sophisticated alternative, we use both the continuous yaw and pitch angles as well as the relative position of the heads (sec. 4.2). Finally, we propose a '2.5D' analysis, where we use the scale of the detected head to estimate the depth positioning of the actors, and combine it with the full head pose estimate to derive their gaze volumes in 3D (sec. 4.3).

We apply these methods to the TV Human Interactions Dataset (TVHID) [22]. This is very challenging video material with far greater variety in actors, shot editing, viewpoint, locations, lighting and clutter than the typical surveillance videos used previously for classifying interactions [3, 21, 29] where there is a fixed camera and scene. We provide additional ground truth annotation for the dataset, specifying which shots contain people looking at each other. Originally, the dataset only had annotations for four specific interactions (hand-shake, high-five, hugging and kissing) but there are many other shots where people are looking at each other.

In a thorough experimental evaluation on the TVHID, we show that the full head pose estimate (i.e. yaw and pitch angles) in combination with the relative position of the heads in a 3D scenario are needed for most real situations to clearly define if two people are LAEO.

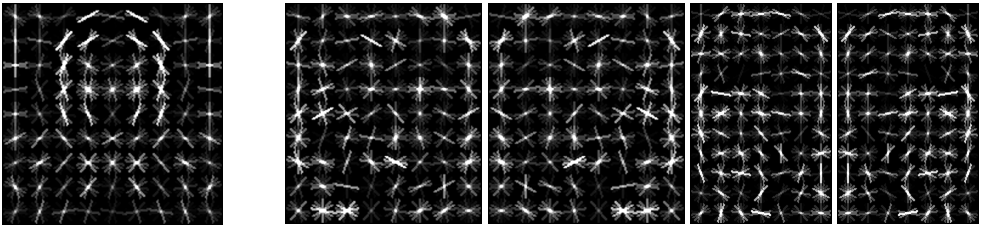


Figure 2: **Models for the multi-view upper-body and head detectors. (Left)** Root filter of the UB detector. This model contains a single component trained from a mixture of all viewpoints. **(Right)** Root filters of the 4 components of the head detector. Each component provides coarse information about the head orientation (i.e. two profile viewpoints and two near frontal viewpoints).

2 Detecting and tracking heads in video shots

The first step in our approach is to detect and to track the heads of the people present in a video shot. We split the task in the following subtasks: *(i)* human upper-body detection in individual frames; *(ii)* grouping upper-body detections over time into tracks; *(iii)* detecting heads within upper-body detections; and, *(iv)* grouping head detections into tracks.

We propose this two-level pipeline because upper-body detection is more robust to clutter than head detection, as it benefits from wider, more distinctive context. The precise localization of the head within the limited area defined by an upper-body detection can then proceed safely. In particular, direct detection of profile heads in uncontrolled scenes would otherwise produce many false positives.

On the other hand, although we already have tracks in step *(ii)*, another tracking stage is performed in step *(iv)* in order to solve the situations where two heads are so spatially close that they fall into the same upper-body bounding box.

2.1 Upper-body detection and tracking

We train a human upper-body (UB) detector using the Felzenswalb *et al.* [12] model. This model usually comprises several components, each specialized for a different viewpoint. In turn, every component is a deformable configuration of parts, each represented as a HOG template [8]. As positive training samples, we used annotated keyframes from the Hollywood movie database [15]. These contain upper-bodies viewed from different angles and at different scales. As negative training samples, we used those images in the INRIA-person dataset [30] which do not contain people. The root filter of the learned UB model is shown in fig. 2 (left). Note that we train a single component using all data at once (note, only a single component is trained because [12] only creates different components if there are different aspect ratios, but all UB annotations have the same aspect ratio).

We process each frame independently with this detector and then group detections over time into UB tracks. For this we use a tracking approach similar to Everingham *et al.* [11]: detection windows in different frames that are connected by many KLT point tracks [25] are grouped in the same track. Next, false-positive tracks are discarded based on track length and the score of the detections they contain. We discard a track if it contains fewer than 20 detections, or if the sum of detection scores over it is below a threshold.

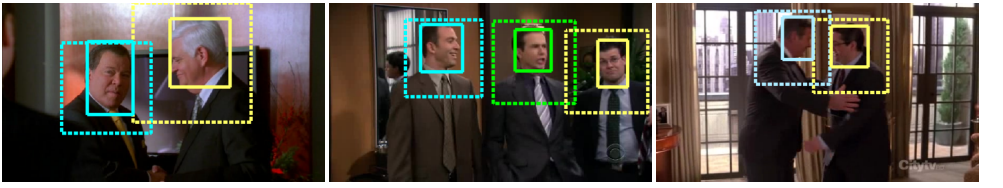


Figure 3: **Examples of UB (dashed) and head (solid) detections.** The head detector is only run inside UB detection windows. Note how heads are localised in various relative positions within the UB windows, adapting to the image content.

2.2 Head detection and tracking

We train a head detector again using the Felzenswalb *et al*. [12] model, and the same images as used to train the UB detector, but with annotations on heads. As this head detector is intended to be run only inside upper-body windows, we provide negative training samples from the area surrounding the head.

We train four components, which correspond to two profile and two nearly frontal viewpoints. Figure 2 (right) shows the root filter of each component. By having multiple components, the detection of a head in a test image delivers a coarse estimation of its viewpoint, in addition to its $x-y$ position and scale. Moreover, each subgroup of components is specialized to a different aspect ratio of the head.

We detect heads in each frame separately with this detector and then track head detections over time as done with UB detections in sec. 2.1. Figure 3 shows examples of UB and head detections in a variety of situations.

3 Continuous head pose estimation

We describe here our approach to estimate two head pose angles: *yaw* (around the Y axis) and *pitch* (X axis). We do not consider *roll* (Z axis). We use Gaussian Processes to directly regress from the image patch within a head detection window to the two pose angles.

3.1 Training a head pose regressor with Gaussian Processes

For each detected head, we crop an $N \times N$ image window H centred on it, where N is the number of pixels of the largest side of the detection window. Then, H is resized to a predefined common size 48×48. Given an observed head window H, the goal is to predict two angles (θ, α) conveying its pose wrt to the camera viewpoint. We formulate this problem in terms of regression and train two separate regressors, one for yaw (θ) and one for pitch (α). As the method is exactly the same, we restrict the explanation to yaw.

The goal is to find a real-valued regression function $\hat{\theta} = f(g(H))$, so that $\hat{\theta} \approx \theta$, where $g(H)$ is a feature vector of H, and θ and $\hat{\theta}$ are the real and estimated angles respectively. We use a histogram of oriented gradients (HOG) [8] as the head descriptor g.

A Gaussian Processes (GP) [23] regressor is employed using a linear mean function, a squared exponential covariance function with isotropic distance measure, and a Gaussian likelihood. We learn the parameters of the two GP regressors by using the GPML 3.1 library [31]. The set of training data is $\mathscr{D} = \{(g(H_i), \theta_i)\}$, where $g(H_i)$ is the HOG descriptor of the i-th training sample (i.e. head) and θ_i is its ground-truth yaw angle.

Figure 4: **Left: Intersection of gaze areas in 2D.** We show heads as red rectangles and gaze areas as yellow rectangles. This method would incorrectly say that these people are not LAEO, since their 2D gaze areas do not intersect. **Right: Geometric constraints in 2D.** We show the estimated yaw and pitch angles as yellow vectors (yaw determines if left or right facing and length; pitch determines orientation). The angle defined by these vectors for (B,C) would classify such pair as LAEO. (Best viewed in colour).

GPs are attractive because they are non-parametric models, and therefore can flexibly adapt to any distribution of the data. Moreover, at inference time, they return both the mean over the output $\hat{\theta}$ as well as its uncertainty (i.e. variance). This offers the possibility to downweight uncertain pose estimates in later processing stages (e.g. sec. 4.2).

4 Classifying pairs of heads as looking at each other (LAEO)

We present in the following subsections our main contribution: three different methods for classifying a pair of people as LAEO or not.

4.1 Intersection of gaze areas in 2D

The simplest method we propose only considers the head pose as discretized into just two directions, i.e. facing left or right. For this we only use the estimated yaw angle and discard the pitch. In addition, the image position and the height of the head are used.

We define as *gaze area* G_i the image region a person head P_i is looking at: a horizontal rectangle extending from the head towards the gaze direction (fig. 4(left)). The height of G_i is given by the height of P_i, while the width is given by the x position of the farthest other head in the scene. To classify whether two heads P_l, P_r are LAEO, we define the $\text{LAEO}_{GA}(P_l, P_r)$ function. Let (x_l, y_l) and (x_r, y_r) be the centres of P_l, P_r, satisfying the condition $(x_l \leq x_r)$, and O_l, O_r be their orientation (i.e. $+1$ facing left, -1 facing right). With these definitions, LAEO_{GA} is

$$\text{LAEO}_{GA}(P_l, P_r) = \text{IoU}(G_l, G_r) \cdot \delta(O_l \cdot O_r < 0) \quad (1)$$

where $\text{IoU}(G_i, G_j) = \frac{G_i \cap G_j}{G_i \cup G_j}$ is the insersection-over-union of the heads' gaze areas G_i, G_j (fig. 4(left)); $\delta(c)$ is 1 if condition c is true, and 0 otherwise.

4.2 Geometric constraints in 2D

The second method we propose takes into account both the yaw and pitch angles defining the full head pose, as well as the image position of the heads. Two people are deemed to be

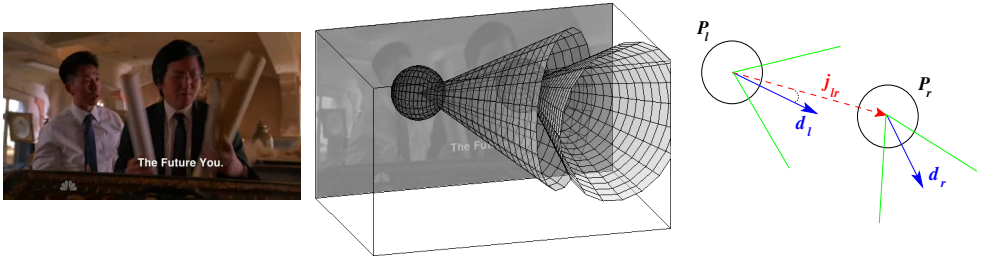


Figure 5: **Geometric constraints in 3D. (left)** Original video frame. **(middle)** 3D representation of a scene with two people. We show heads (spheres) and their gaze volumes (cones). **(right)** View from above, with heads (circles) and gaze direction vectors (blue arrows) $\mathbf{d}_l$ and $\mathbf{d}_r$ defined by the yaw and pitch angles. Green lines are the boundaries of the conic gaze volumes. The red vector is $\mathbf{j}_{lr}$ and goes from P_l to P_r. With this configuration, P_r lays inside P_l gaze area but P_l does not lay inside that of P_r. Therefore, the two people are correctly classified as not LAEO. (Best viewed in colour).

LAEO if *(i)* the person on the left has a positive yaw angle and the person on the right has a negative yaw angle; *(ii)* the cosine of the difference between their yaw angles is close to -1; and, *(iii)* the vectors defined by the pitch angles have to be similar to the vectors that join the heads, in both directions. fig. 4(right) shows an example that should be highly scored as LAEO.

For a head P_i, let (x_i, y_i) be the coordinates of its centre, and θ_i, α_i the estimated yaw and pitch angles. We define the following function $\text{LAEO}_{GC}(P_l, P_r)$ to formalize the above constraints and decide if two heads P_l, P_r are LAEO (with $(x_l \leq x_r)$):

$$\begin{aligned}\text{LAEO}_{GC}(P_l,P_r) = \beta_\theta \cdot [\delta(\theta_l \cdot \theta_r < 0 \wedge \theta_l > \theta_r) \cdot (1-\cos(\theta_l - \theta_r)) \cdot 0.5] + \\ \beta_\alpha \cdot [(1+\cos(\alpha_l - \gamma_{lr})) \cdot 0.25 + (1+\cos(\alpha_r - \gamma_{rl})) \cdot 0.25]\end{aligned} \quad (2)$$

where γ_{ij} is the orientation of the vector going from P_i to P_j in the image plane; the symbol '$-$' between two angles denotes their orientation difference; β_θ and β_α are weights, so that $\beta_\theta + \beta_\alpha = 1$. Note that each row of eq. (2) (omitting their β) ranges in $[0,1]$. Therefore, LAEO_{GC} ranges in $[0,1]$, with 1 the best possible score.

We take advantage of the information about the uncertainty of the estimated angles returned by the GP regressor (i.e. standard deviation σ) by using it to set β_θ and β_α for each test pair of people: $\beta_\theta = (\sigma_{\theta_l}^{-1} + \sigma_{\theta_r}^{-1})/(\sigma_{\theta_l}^{-1} + \sigma_{\theta_r}^{-1} + \sigma_{\alpha_l}^{-1} + \sigma_{\alpha_r}^{-1})$, and, $\beta_\alpha = 1 - \beta_\theta$.

4.3 Geometric constraints in 3D

The most complex method we propose operates in a simplified 3D space. We place each person's head P_i in a common 3D coordinate system by using the image coordinates of the head centre as (x_i, y_i) and deriving the depth coordinate z_i from the head size in the image. Coordinates z_i are derived as a direct proportion between all the heads present in the scene, by assuming that heads are enclosed in cubes of side length equal to the BB height. Heads are z-ordered so that the largest head in the image is the closest one to the camera.

The gaze volume of a head P_i is represented as a 3D cone C_i with apex at (x_i, y_i, z_i) and axis orientation defined by the estimated yaw and pitch angles (fig. 5). We classify two heads P_l and P_r as LAEO if P_l lays inside C_r, and P_r lays inside C_l. Note how this method uses all the available information.

More formally, we define the LAEO_{3D} score by the following equation:

$$\text{LAEO}_{3D}(P_l, P_r) = \frac{(\varphi - \Delta(\mathbf{j}_{lr}, \mathbf{d}_l) + (\varphi - \Delta(\mathbf{j}_{rl}, \mathbf{d}_r))}{2\varphi} \quad (3)$$

where the angle φ represents the *aperture* of the gaze cone and is a free parameter; $\Delta(\cdot,\cdot)$ is the angle between two vectors; $\mathbf{j}_{ij}$ is the vector from P_l to P_r, i.e. defined as $(x_i, y_i, z_i) \rightarrow (x_j, y_j, z_j)$; and, $\mathbf{d}_i$ is a vector defined by the yaw and pitch angles of P_i (fig. 5). Note that for our experiments, the magnitude of vector $\mathbf{d}_i$ is 1 whereas the direction is given by the estimated yaw and pitch angles.

5 Experiments and results

5.1 Datasets

Head pose. We use two datasets to learn yaw and pitch angles. The first is the *CMU Pose, Illumination and Expression (CMU-PIE)* dataset [26]. It contains images of 68 people from 13 different camera viewpoints, corresponding to 9 discretized yaw angles ($[-90, 90]$ degrees). Images have been captured in two different sessions and in each session there are four subsets, corresponding to different types of variations: *expression*, *illumination*, *lighting* and *talking*. The second dataset is the *IDIAP head pose (IDIAP-HP)* [2]. It contains 8 videos recorded in a meeting room and 15 videos in an office. Yaw, pitch and roll angles ground-truth is provided for each person in every frame.

LAEO. We evaluate our LAEO classifiers on the *TV human interactions dataset* (TVHID) of [22]. It contains a total of 300 video clips grouped in five classes: *hand-shake*, *high-five*, *hug*, *kiss* and *negative*. Each video clip might be composed of several shots, i.e. periods corresponding to continuous camera drives. Therefore, we have computed shot boundaries as maxima in the colour histogram differences between subsequent frames. For our task, we have additionally annotated all the videos by assigning one of the following labels to each shot: *label 0*: no pairs of people are LAEO; *label 1*: one or more pairs of people are LAEO in a clearly visible manner; *label 2*: a pair of people are LAEO, but at least one of them has occluded eyes (e.g. due to viewpoint or hair); and, *label 3*: a pair of people are facing each other, but at least one of them has closed eyes (e.g. during kissing). There are a total of 443 video shots, where 112 have label 0, 197 label 1, 131 label 2 and 3 label 3. Therefore, the dataset contains 112 negative (label 0) and 331 positive samples (labels 1, 2 and 3). Note that we do not distinguish the last three cases (i.e. 1,2,3) in the experiment and, for example, we treat looking at each other but eyes closed as a positive. We release both the LAEO annotations and the shot boundaries at the following URL: `http://www.robots.ox.ac.uk/~vgg/data/`.

5.2 Learning yaw and pitch estimators

In order to train the head pose estimators, the first step is to detect all the heads from the training images by using the detector of sec. 2.2. Next, all detected heads are normalized to a common size of 48×48 pixels and HOG features are extracted[1] . The HOG features are used as input $\mathbf{x}$ to the GP regressor, which outputs the target angle (i.e. θ or α).

[1] Blocks of 8×8 pixels, 9 orientation bins.

We learn the yaw estimator from the subsets *expression* and *illumination* of CMU-PIE dataset, and the pitch estimator from the subset *meeting room* of IDIAP-HP dataset.

We used the GPML 3.1 library [31] for GP regression. In order to evaluate the yaw GP regressor, we split the dataset in two parts: six random people are used for validation and the remaining ones for training. We have repeated this procedure for five trials. We compute the root mean squared error (RMSE) for each validation set. The average RMSE over the five validation sets is 17.4 degrees. We repeat the same procedure for training the pitch GP regressor but, in this case, only one person is used for validation in each trial, and all others for training. The average RMSE for pitch is 6.9 degrees.

For comparison purposes, we trained and validated a linear regressor (i.e. using Matlab's *robustfit* function) on the same data. This linear regressor delivers about twice average RMSE than the GP regressor, which justifies our choice of GPs.

After the above evaluations, we train a final GP regressor from all the available samples and use it in the LAEO experiments below.

5.3 LAEO evaluation

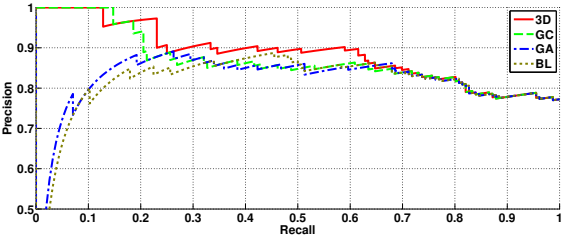
We evaluate here the perfomance of the proposed LAEO classifiers on the following task: *is there any pair of people LAEO at any time in this video shot?* To assign a LAEO score to a shot we: *(i)* assign a LAEO score to each pair of people in every frame using one of the methods in sec. 4; *(ii)* assign a LAEO score to each frame, as the maximum over all pairs of people it contains; *(iii)* slide a window along the temporal axis and average the scores of all frames in the window that are greater than a threshold T; *(iv)* assign a LAEO score to the shot, as the maximum over all temporal window scores. Intuitively, these steps will lead to higher scores for pairs of heads that are LAEO over a sustained period of time. This avoids producing false positives for accidental geometric alignments over a few frames (as opposed to simply averaging the thresholded scores over frames).

We evaluate performance on the TVHID dataset, using our new annotations. Each method is used to score every shot, and then the average precision (AP) is used to compare the performance of the methods.

Baseline method. As a baseline we use the coarse directional information provided by our head detector (i.e. which model component triggered the detection) to define gaze areas as it in sec. 4.1. Eq. (1) is used to score person pairs. Note that this baseline computes neither yaw nor pitch angles.

Experimental results. We evaluate the perfomance of the methods proposed in sec. 4. The TVHID release of [22] defines two disjoint partitions. We run experiments on two trials, where one partition is used for training and the other for testing, and then report mean AP (mAP) over the two trials.

We set the free parameters of the proposed LAEO scoring methods on the training set so as to maximize AP, i.e. (i) the aperture ϕ of the cone for the method of sec. 4.3; (ii) the threshold T on the LAEO scores used by all methods during the temporal window averaging. (iii) the length of the temporal window. The optimal φ is found to be in the range $[30,45]$, depending on the training set. The optimal T was in the range $[0.3,0.5]$ and the optimal length of the temporal window was between 5 and 9 frames, depending on the LAEO scoring method and training set.



	mAP
chance	0.75
BL	0.816
GA	0.822
GC	0.846
3D	**0.862**

Table 1: **Comparision of LAEO methods. (left) Precision-recall curves for partition 2. (right) Mean average precision (mAP) over the two partitions for each method.** BL = baseline method; GA = intersection of gaze areas in 2D (sec. 4.1); GC = geometric constraints in 2D (sec. 4.2); 3D = geometric constraints in 3D (sec. 4.3).

Figure 6: **Test shots according to geometric constraints in 3D. (Top two rows)** Top 10 shots from partition 2 of TVHID, training on partition 1. The frame with red border is a false positive. **(Bottom two rows)** Top 10 shots from partition 1 of TVHID, training on partition 2.

Table 1(left) shows the precision-recall curves of the proposed methods for partition 2 and table 1(right) shows the mAP over the two partitions. The baseline method delivers a mAP of 0.816. Moreover, if all test shots are scored with uniformly distributed random values, the mAP over 10 trials is 0.75, which shows the baseline already works better than chance. The LAEO$_{GA}$ method yields a similar mAP of 0.822, suggesting that using the sign of the estimated yaw angles is equivalent to using the head direction directly output by our head detector. The higher performance of the LAEO$_{GC}$ method (0.846) demonstrates the importance of the information provided by both continuously estimated angles and of the 2D geometric relations between the two heads. Finally, the LAEO$_{3D}$ method achieves the highest mAP (0.862). This supports our intuition about the importance of a full 3D reasoning, including the 3D head pose vectors and also the relative position of the people in a 3D coordinate system.

Our method is able to localise the LAEO pair both spatially and temporally. Figure 6 shows the middle frame of the highest scored temporal window for each of the top ten ranked shots, according to LAEO$_{3D}$. Note the variety of scenarios where the method sucessfully works.

6 Conclusions

We presented a technique for automatically determining whether people are looking at each other in TV video, including three methods to classify pairs of tracked people. Our best method uses the scale of the detected heads to estimate the depth positioning of the actors, and combines it with the full head pose estimate to derive their gaze volumes in 3D. While we report quantitative performance at shot level, our method allows the interacting people to be localised both spatially (i.e. the pair of heads with the highest LAEO score) and temporally (i.e. temporal sliding window). In conclusion, the recognition of LAEO pairs introduces a new form of high-level reasoning to the broader area of video understanding.

Acknowledgements. V. Ferrari was supported by a SNSF Professorship. M.J. Marín-Jiménez is grateful to Junta de Andalucía. Financial suport was also provided by ERC grant VisRec no. 228180.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, 2009.

[2] S.O. Ba and J.-M. Odobez. Evaluation of multiple cue head pose estimation algorithms in natural environements. In *ICME*, 2005.

[3] S.O. Ba and J.-M. Odobez. Recognizing visual focus of attention from head pose in natural meetings. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans. on*, 39(1):16 –33, feb. 2009.

[4] B. Benfold and I. Reid. Colour invariant head pose classification in low resolution video. In *Proc. BMVC.*, September 2008.

[5] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE PAMI*, 25:1063–1074, 2003.

[6] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Proc. ECCV*, 2010.

[7] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *Proc. CVPR*, 2009.

[8] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005.

[9] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proc. BMVC.*, 2009.

[10] M. Everingham and A. Zisserman. Identifying individuals in video by combining generative and discriminative head models. In *Proc. ICCV*, 2005.

[11] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proc. BMVC.*, 2006.

[12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 32(9):1627–1645, 2010.

[13] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, Jun 2008.

[14] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. In *Proc. CVPR*, 2009.

[15] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *International Workshop on Sign, Gesture, Activity*, 2010.

[16] I. Laptev and P. Perez. Retrieving actions in movies. In *Proc. ICCV*, 2007.

[17] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

[18] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos. In *Proc. CVPR*, 2009.

[19] E. Murphy-Chutorian and M. Manubhai Trivedi. Head pose estimation in Computer Vision: A survey. *IEEE PAMI*, 31:607–626, 2009.

[20] M. Osadchy, Y.L. Cun, and M.L. Miller. Synergistic face detection and pose estimation with energy-based models. *J. Mach. Learn. Res.*, 8:1197–1215, May 2007.

[21] S. Park and J.K. Aggarwal. A hierarchical bayesian network for event recognition of human actions and interactions. *Association For Computing Machinery Multimedia Systems Journal*, 2004.

[22] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. High Five: Recognising human interactions in TV shows. In *Proc. BMVC.*, 2010.

[23] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[24] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proc. ECCV*, 2010.

[25] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, pages 593–600, 1994.

[26] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE PAMI*, 25(1):1615 – 1618, December 2003.

[27] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" – learning person specific classifiers from video. In *Proc. CVPR*, 2009.

[28] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proc. ICCV*, 2005.

[29] W. Waltisberg, A. Yao, J. Gall, and L. Van Gool. Variations of a Hough-voting action recognition system. In *Proc. of ICPR 2010 Contests*, 2010.

[30] website. INRIA person dataset. http://pascal.inrialpes.fr/data/human/, 2005.

[31] website. GPML Matlab code. http://www.gaussianprocess.org/gpml/code/matlab/doc/, 2011.