Structured learning of human interactions in TV shows

Alonso Patron-Perez, *Member, IEEE*, Marcin Marszalek, Ian Reid, *Member, IEEE* and Andrew Zisserman

Abstract—The objective of this work is recognition and spatio-temporal localization of two-person interactions in video. Our approach is person-centric. As a first stage we track all upper bodies and heads in a video using a tracking-by-detection approach that combines detections with KLT tracking and clique partitioning, together with occlusion detection to yield robust person tracks. We develop local descriptors of activity based on the head orientation (estimated using a set of pose-specific classifiers) and the local spatio-temporal region around them, together with global descriptors that encode the relative positions of people as a function of interaction type. Learning and inference on the model uses a structured output SVM which combines the local and global descriptors in a principled manner. Inference using the model yields information about which pairs of people are interacting, their interaction class, and their head orientation (which is also treated as a variable, enabling mistakes in the classifier to be corrected using global context). We show that inference can be carried out with polynomial complexity in the number of people, and describe an efficient algorithm for this. The method is evaluated on a new data-set comprising 300 video clips acquired from 23 different TV shows and on the benchmark UT-Interaction Dataset.

Index Terms—Human interaction recognition, video retrieval, structured SVM

1 Introduction

The recognition of human activities is an important step towards the long-term goal of achieving a fully automatic understanding of a scene. When describing a scene, people are often characterized in terms of the actions that they perform (e.g. *playing* football, *talking* on the phone, *hugging* each other, etc.).

As the title of this paper suggests, we are interested in modeling human *interactions*, focusing on the recognition of interactions between two people in complex scenarios and their spatio-temporal localization. In particular we deal with four symmetrical interaction classes: hand shakes, high fives, hugs and kisses. These interaction classes are symmetric since the people involved in it more or less perfom the same body movements. Also, we describe how to model asymmetric interactions such as: pushing or kicking. Our goal is to recognize these interactions in TV video such as sitcoms and dramas. Because of the complexity and variability of (edited) TV video, finding a way to represent interactions that captures relevant and distinctive information of each interaction class is challenging. An interaction descriptor has to be simultaneously (i) relatively coarse in order to avoid representing irrelevant variation, and (ii) to some extent focused to avoid learning background noise when codifying the interaction.

To address these challenges we propose the following solutions. We choose, in the first instance, to describe an interaction from a person-centered perspective, therefore avoiding learning information derived from the background clutter. In order to do this, we first need to locate people in the videos. This allows us to focus on regions close to people (where interactions could be happening). We can also see this first step as reducing the search space of an interaction. This is in contrast to other approaches in single-action recognition (as discussed in Section 2), where features are estimated in the whole frame or video and then clustered to localize where the action occurs. Another advantage of implementing a person-centered descriptor is that, depending on the camera angle, both persons are not always visible in a given frame, and we would like to be able to provide a classification even in these instances. Localization of people is done by using an upper body detector, analogous to [1], [2], [3]. The detections obtained in this way are linked by a combination of KLT tracking and clique partitioning (CP) clustering to form *tracks*. A track is a temporally ordered set of bounding boxes corresponding to the upper body of the same person. Once the location of a person is known, we describe the local neighborhood by coarsely quantifying appearance and motion, which we term the *local* context. Furthermore, we use the person's head orientation to select relevant regions inside his local context. We train a discrete head orientation classifier using a multiclass linear Support Vector Machine (SVM) and

- *Alonso Patron-Perez, Ian Reid and Andrew Zisserman are with the Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK. E-mail: {alonso, ian, az}@robots.ox.ac.uk.*
- *Marcin Marszalek is with Google Research, Zurich, Switzerland. E-mail: marcin@robots.ox.ac.uk.*

This work was supported by the ERC grant VisRec no. 228180 and CONACYT.

Digital Object Indentifier 10.1109/TPAMI.2012.24 0162-8828/12/$31.00 © 2012 IEEE

use it to estimate the head pose in each bounding box of every computed track.

When dealing with human interactions, extracting information about relations between different people (or their local context) can induce special constraints for each interaction. These relations are what we call the *global* context of an interaction. The global context cues that will be used here are depicted in Fig. 1. Relative spatial locations of people can indicate the type of interaction being performed (if any). For example, two people are more likely to be labeled as hugging if they are close together than if they are far away from each other. Some interaction classes will impose more constraints on their "favorite" spatial configuration. Though in all cases the relative spatial distances depend not only on the interaction class but also on the camera's point of view and the scale of the bounding boxes. A second cue corresponds to head orientation. Visual attention of people usually corresponds to image regions of high interest, as was explored by Benfold and Reid [4] in the context of automatically guiding a surveillance system. We combine the extracted local and global context information to learn a structured SVM classifier for an interaction set, whose output is the set of head poses, the pairs (if any) of interacting people and their interaction labels.

To quantitatively evaluate the recognition and retrieval of interactions we introduce a new dataset of interactions. This contains 300 video clips compiled from 23 different TV shows. It is ground truth annotated with the spatio-temporal localization and category of the interactions, as well as with localization and head pose of each person in every frame. We also perform tests on a current standard interaction dataset (UT-Interaction).

This paper is an extension of our previous work [5]. These extensions include: *(i)* a more robust method for computing tracks based on a combination of KLT tracking and CP; *(ii)* a new structured learning formulation where head pose is now included as an extra variable to be estimated; *(iii)* a far more efficient (polynomial instead of exponential complexity) inference algorithm; *(iv)* a quantitative comparison of the proposed algorithm with a baseline on our new dataset; *(v)* an extension to handle asymmetrical interactions; and *(vi)* an evaluation on a stardard benchmark dataset (UT-Interaction). Additionally, throughout we provide a more in-depth description and analysis of each stage of the approach.

The remainder of this paper is divided as follows. Section 2 reviews approaches to interaction recognition and structured learning. Section 3 deals with the process of extracting peoples' tracks. We introduce our local context descriptor in Section 4, while the intuition behind using global cues and structured learning is explained in Section 5. We test our method in a video retrieval task using the new TV dataset, and in a classification task using a benchmark dataset in Section 6. Conclusions and directions for future work are given in Section 7.

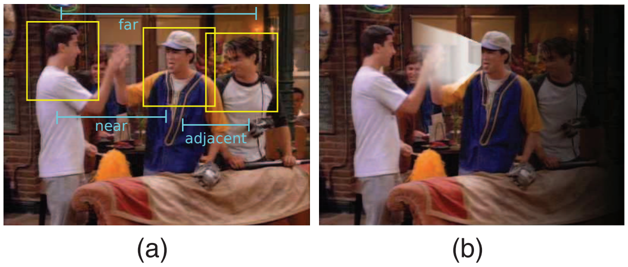

Fig. 1. Global cues for recognizing human interactions. (a) Relative spatial relations between people can be indicative of the type of interaction being performed. (b) Head orientation (here indicated by a conical beam) is a cue to which people, in a given frame, are more likely to be interacting with each other.

2 Related work

Although a considerable amount of work has been done in the past ten years on single-person action recognition, only over the last few have researchers moved from constrained scenarios to realistic ones [6], [7], [8], [9], [10]. The topic of recognizing interactions remains, with few exceptions, rather unexplored. In many of these works, two-person interactions have been included, but they have been treated in the same way as single-person actions without distinction. While this could give reasonable results in some cases, the task of recognizing two-person interactions has its own characteristics and constraints that can be exploited to achieve more accurate and robust results.

Among the first attempts at interaction recognition is the work by Oliver *et al.* [11]. Their method was based on the analysis of motion trajectories obtained from blob-tracking of people. Such a representation would be too coarse to capture the subtleties of the interaction classes of interest in our work. More recent approaches include Park and Aggarwal [12], [13] and Ryoo and Aggarwal [14]. They used a hierarchical method where the lower levels dealt with the tracking of body parts while higher levels introduced a semantical interpretation of the interactions. In the same way as [11], they used a very constrained dataset where people are always viewed from the side and there is no camera motion nor background clutter.

Moving towards more realistic datasets, Ryoo and Aggarwal [15] represented an interaction by modeling relations between pairs of spatio-temporal features using spatial and temporal predicates, such as "before", "after", "near" or "far". Spatio-temporal features are grouped into two levels, the first corresponding to what they call atomic actions (e.g. stretch arm, lower leg, etc.), which are in turn used to codify interactions in a second level. To provide localization, spatio-temporal feature pairs vote for a beginning and an end of an interaction. A similar approach was presented by Yuan *et al.* [16], where point trajectories were used instead of spatio-temporal features.

Both methods were tested with the UT-Interaction dataset [15], which, although it presents more realistic scenarios compared to previous human interaction datasets, still contains no background clutter or severe camera motion. For a more complete overview of human interaction recognition methods see [17].

Structured learning has been used for several applications in Computer Vision. In particular, developments in structural Support Vector Machines [18], [19] as well as general off-the-shelf software implementations have made this approach very popular. Blaschko and Lampert [20] used it to learn a mapping between images and object bounding boxes, extending their approach in [21] to model local and global context information. Desai *et al.* [22] used structured SVM to learn spatial relations between object categories aiming to obtain a simultaneous classification of all bounding boxes in an image, while Wang *et al.* [23] use it to learn both dependencies between objects and object attributes, and between the attributes themselves. Similar to the previous methods, we want to obtain a joint classification of the people detected in a frame taking into account different sources of information. In contrast to [22] we learn spatial relations of *people* given their head pose and interaction class label (instead of learning spatial relations between co-occurring interaction classes). Head pose is an important piece of information about a person but is not considered as an attribute of an interaction class (i.e, a profile head orientation by itself doesn't tell you much about a hand shake or a high five class). Our model also obtains a configuration label that indicates which pair(s) of people, if any, are more likely to be interacting. Structured learning has also started to be used for human activity recognition as shown in recent work by Lan *et al.* [24] and Niebles *et al.* [25]. In the former, a latent structured model is used to represent group activities, including two-person interactions like talking, while the latter focused on modeling the temporal structure of human activities.

3 Search space reduction

In order to direct processing power and avoid many false positives due to background clutter we focus only on people in the scene. This approach was taken by Laptev and Perez [6], for priming key frames, and Kläser *et al.* [3] for tracking in videos, for the case of single-person action recognition. For our purposes, four stages are required: first, people are detected by their upper bodies in a similar manner to Ferrari *et al.* [2], [26]; second, the upper body detections are linked together into tracks of the same person. Here, we use the KLT [27], [28] method of Everingham *et al.* [29] as well as the clique partitioning of [2] to join broken tracks; third, tracks are filtered to remove false positives using a linear classifier similar to that of [3]; and finally, head pose is computed in each frame so that both the upper body and direction of the head are available for the descriptor. For interactions a number of modifications are required over previous methods, and we detail the entire process next.

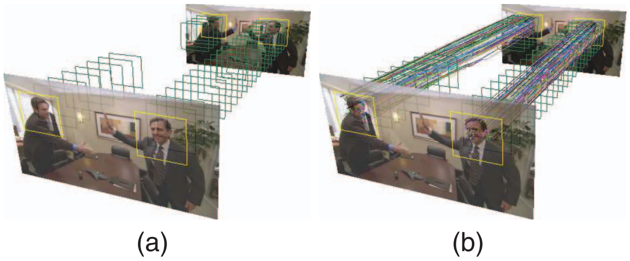

Fig. 2. Computing tracks. (a) Raw upper body detections. (b) Linking detections by KLT tracking of points inside the head region. The detections of a person can be linked even if the person is not detected for several frames.

Detecting people. The first step consists of detecting people in each frame of every video. For this, we train two upper body detectors using a standard Histogram of Oriented Gradients (HOG) [1] descriptor and a linear SVM classifier. The detectors were trained using the Hollywood-2 dataset [7] to look for upper bodies in 128×128 and 64×64 windows. In practice we found a single detector was insufficient to cover the wide range of scales present in the dataset.

Linking detections. Next the upper body detections are linked in order to form tracks of people. This involves two steps. First, we use a method similar to the one employed in [3], where feature points initialized inside each upper body detection are tracked forwards and backwards in time. This tracking is done using a KLT tracker; and two detections are linked depending on the number of tracked points that pass through both of them using an overlap ratio. However, unlike [3], we don't track the whole upper body bounding box but instead just the head region, similar to Everingham *et al.* [29] and shown in Fig. 2. There are two main reasons for this choice. First, is that the bounding box of an upper body contains a significant proportion of background pixels (unlike a near frontal face). Second, the section of the bounding box corresponding to the torso usually contains motion (e.g. from the hands) or is occluded in some of the interactions (e.g. hugging), which makes the linking unstable.

After this first step, a set of initial tracks is created. Some of the tracks belonging to the same person are still broken due to brief occlusions (or fast motions) of the head region. To connect these tracks we employ clique partitioning (CP). We use an overlap measure between the last and the first bounding boxes of two tracks. This is only done for tracks that don't overlap in time. The bounding boxes that are interpolated to connect these tracks can be marked as occluded. We avoid computing an interaction label for these tracks due to the lack of reliable information. Temporal gaps in the computed tracks are filled by interpolation while the position and size of the bounding boxes belonging to the same track are temporally smoothed using quadratic smoothing as in [3].

Many of the videos in our dataset include different *shots* (camera view angles). We need to estimate shot boundaries to prevent linking bounding boxes that belong to different shots. In general these boundaries are sharp cuts (rather than fades, or cross-fades), so are easily detectable. Two heuristic measures are used to decide if consecutive frames in a video belong to different shots: a pixel-by-pixel frame subtraction and difference between the frames' color distributions. If the value of each of these measures is significantly above the average of the clip, we consider a shot boundary to have been detected.

Track pruning. Depending on the accuracy of the upper body detector, some of the resulting tracks will inevitably be computed using false positive detections. Because relations between people in the scene play an important role in our approach, tracks that don't represent real people will introduce corrupted information into the interaction model. Therefore, we follow [3] by learning a track classifier in order to discriminate between true and false tracks. From each track a descriptor is computed that consists of 20 different statistical measures: relative length (with respect to the shot length); average and maximum SVM scores of the bounding boxes (obtained from the upper body detector); average, maximum and minimum relative sizes of the bounding boxes (with respect to the frame size); change in scale; completeness; maximum and average overlap with other tracks; relative horizontal and vertical translations; relative leftmost and rightmost positions; percentage of the bounding boxes where the head is occluded; scale; and position variance. A linear SVM classifier is trained from these descriptors and is evaluated in Section 6.1.1.

Head pose estimation. Head orientation is used by the local and global descriptors of an interaction, and we estimate it along the computed tracks. As with [4], [30], we train a discrete head pose classifier. We learn a simple set of one-vs-all linear SVM classifiers using HOG descriptors corresponding to five discrete orientations: profile-left, frontal-left, frontal-right, profile-right and backwards (Fig. 3). Perfect frontal views are very rare, and they are included in either of the two frontal categories. Using these classifiers a score for each head pose in every bounding box of a track is obtained. Additionally, the scores obtained in each track are temporally smoothed by applying a quadratic smoothing in the same manner as when smoothing the tracks' position and scale. An evaluation of these classifiers is given in Section 6.1.2.

4 Modeling the Local Context of Human Interactions

This section presents a person-centered descriptor that uses attention (head orientation) and the local spatial and temporal context in a neighborhood of each detected person. The local context, comprised of histograms of gradients and motion, aims to capture cues such as hand and arm movement.

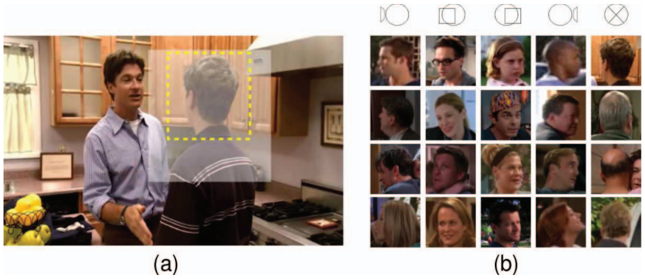

Fig. 3. Head pose estimation. (a) Area inside an upper body detection used to estimate the head pose. (b) Examples of the discrete head orientations used: profile-left, frontal-left, frontal-right, profile-right and backwards.

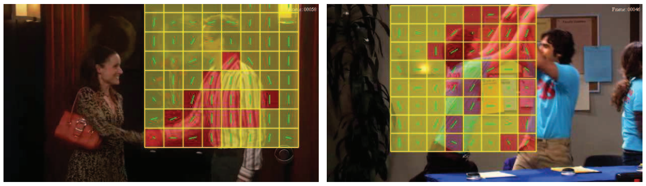

Fig. 4. Local context descriptor. Grid used to describe the person's local context. Highlighted in the grid are the dominant gradient orientation of each cell (displayed with green lines) and cells that contain significant motion (in red).

We start by superimposing an 8×8 grid around an upper body bounding box. The size of each cells is dependent on the bounding box size and thus correctly adapts its scale. Then, histograms of oriented gradients and optical flow are computed in each of its cells. Examples of the area covered by the local context can be seen in Fig. 4. This technique of using histograms of gradients and flow is a coarse analog of the descriptors used by Dalal *et al.* [1]. Gradients are discretized into five bins: horizontal, vertical, two diagonal orientations and a no-gradient bin. Optical flow is also discretized into five bins: no-motion, left, right, up and down. The histograms are independently normalized and concatenated to create an initial grid descriptor $\mathbf{g}$ of size $P = 640$.

To obtain the final descriptor $\mathbf{d}$, the discrete head orientation is taken into account. The inclusion of head orientation is aimed at capturing information correlated with it. Assuming that an interaction occurs in the direction a person is facing, this can provide a weak kind of view invariance. The goal is to create a compact and automatic representation from which a different classifier for each discrete head orientation can be learned. To do this, the discrete head orientation, θ, is used to perform the following operation: $\mathbf{g}^{+} = \mathbf{g} \otimes \boldsymbol{\delta}_{\theta}$, where $\otimes$ is the Kronecker product, $\boldsymbol{\delta}_{\theta}$ is an indicator vector of size $D = 5$ (corresponding to the discrete head orientations) having a one at position θ and zero everywhere else. Effectively the Kronecker product concatenates the vectors obtained

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

by multiplying $\mathbf{g}$ with each element of $\boldsymbol{\delta}_\theta$, resulting in a vector $\mathbf{g}^+$ of size PD. An extra copy of $\mathbf{g}$ is concatenated at the end of the descriptor $\mathbf{g}^+$ to form the final descriptor $\mathbf{d} = [\mathbf{g}^+; \mathbf{g}]$. This is to account for any information that is independent of the head orientation and to help in cases where the automatic estimation of the head orientation is wrong. Because of the symmetry of the discrete head orientations, in practice only three are used (profile-left, frontal-left and backwards) when learning the local context descriptor classifiers. This is achieved by simply flipping the grid descriptor for the profile-right and frontal-right cases. The descriptor $\mathbf{d}$ is used as a data vector for training a linear one-vs-all SVM classifier for each interaction. Details of the training procedure are given in Section 6.

5 Introducing global context cues

In the previous section, head pose was used to capture important correlations located inside the local context of a person. Here we employ it at a more global scale by using a person's head orientation to infer who, of the other people in the frame, is more likely to be interacting with him (we assume that people tend to look at each other while interacting). By combining head orientation with the relative location of people in a frame, we expect to model global characteristics of interactions. These global cues will be used in conjunction with the local context information previously described to obtain joint classification of all the people present in a frame within a structured SVM framework [19]. For the remaining of this section we assume that human interactions are symmetrical and leave the asymmetrical case for Section 6.2.

Structured SVM. In a general structured prediction problem, the objective is to learn, in a supervised way, a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ from inputs $\mathbf{x} \in \mathcal{X}$ to complex outputs $\mathbf{y} \in \mathcal{Y}$ given a set of training example pairs $(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_N, \mathbf{y}_N)$. The mapping is usally defined in terms of a scoring function $S(\mathbf{x}, \mathbf{y}; \mathbf{w})$ as:

$$\begin{aligned} \hat{\mathbf{y}} &= \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} S(\mathbf{x}, \mathbf{y}; \mathbf{w}) && (1) \\ S(\mathbf{x}, \mathbf{y}; \mathbf{w}) &= \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) && (2) \end{aligned}$$

where $\Phi(\mathbf{x}, \mathbf{y})$ is a combined feature map of inputs and outputs that codifies the underlying structure of the output space. Because $S(\mathbf{x}, \mathbf{y}; \mathbf{w})$ provides a numerical value, we can think of the parameters $\mathbf{w}$ as weights that favor certain configurations in $\Phi(\mathbf{x}, \mathbf{y})$. Another way of interpreting the output value of S is as a score that measures how compatible a specific structured label $\mathbf{y}$ is with respect to an input vector $\mathbf{x}$. Therefore, from this point forward, we refer to S as a scoring or compatibility function. To learn the weights $\mathbf{w}$, the problem is set using a soft-margin SVM formulation:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2} \parallel \mathbf{w} \parallel^2 + C \textstyle\sum_{i=1}^N \xi_i \\ \text{subject to} \quad & \langle \mathbf{w}, \delta\Phi_i(\mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \qquad (3) \\ & \forall i, \, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i, \, \xi_i \geq 0 \end{aligned}$$

where $\delta\Phi_i(\mathbf{y}) \equiv \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y})$ and ξ_i is a slack variable. The constrain in (3) makes explicit that the difference between the score obtained by using the correct label $\mathbf{y}_i$ and the score of any other label has to be larger than or equal to the loss $\Delta(\mathbf{y}_i, \mathbf{y})$. This is known as margin re-scaling [18]. The number of constraints defined in (3) can be extremely large (depending on the range of the output space), and in practice a full optimization using all the constraints is intractable. A solution for this problem is to optimize only a subset of these constraints such that the resulting solution is still accurate. The constraints are selected by finding the label $\mathbf{y}$ that maximizes $\mathbf{w}^T\Phi(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}_i, \mathbf{y})$ for each input example i. This is known as finding the most violated constraint and is described in [19].

5.1 Formalizing the problem

The structured learning problem of interactions is posed in the following terms: in each frame there is a set of upper body detections $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_M]$. Each detection $\mathbf{x}_i = [l_x^i \;\; l_y^i \;\; s^i \;\; \mathbf{h}^i \;\; \mathbf{V}^i]$ has information about its upper left corner location (l_x, l_y), scale (s), head orientation SVM scores ($\mathbf{h} \in \mathbb{R}^D$) and local context SVM classification scores ($\mathbf{V} \in \mathbb{R}^{KD}$). $\mathbf{V}$ is obtained by classifying the local context descriptor associated with this detection using the interaction classifiers previously learned for each discrete head pose. Here, $D = 5$ and $K = 5$ correspond to the number of discrete head orientations and number of interaction classes (including the no-interaction class) respectively. Associated with each frame is a label $\mathbf{Y} = [y_1 \ldots y_M \; \theta_1 \ldots \theta_M \; y_c]$. This label is formed by an interaction class label $y_i \in \{0, .., K-1\}$ and head orientation label $\theta_i \in \{1, ..., D\}$ for each detection (where 0 represents the no-interaction), and a configuration label y_c that serves as an index for one of the valid pairings of detections. For example, for three detections there are four valid configurations: {(1), (2), (3)}, {(1), (2,3)}, {(1,3), (2)} and {(1,2), (3)}, where (i, j) indicates that detection i is interacting with detection j. Note, head pose appears both in $\mathbf{X}$ and $\mathbf{Y}$. However, $\mathbf{X}$ is composed from the raw measurements of the pose classifiers, whilst $\mathbf{Y}$ has the ground truth label of the pose.

5.2 Defining the scoring function

The compatibility scoring function $S(\mathbf{X}, \mathbf{Y}; \mathbf{w})$ is defined as the sum of various potentials. These potentials, defined below, take into account the local and global properties we seek to model, and their separate and combined effects are analyzed in Section 6.1.4.

Head orientation potential (Ψ_h). The first potential function deals with scores obtained by the head pose classifier. As these scores are obtained from an SVM classifier, a higher positive value for a given head pose indicates a greater confidence in the classification. By adding the scores of each bounding box i given a specific labeling we have $\Psi_h(\mathbf{X},\mathbf{Y}) = \sum_i^M h^i_{\theta_i}$. The labeling that maximizes this function corresponds to selecting for each bounding box the labels θ_i that have the highest score $h^i_{\theta_i}$. Two additions are made to this function. First, because we will combine its output with the outputs of two other potential functions, we need to scale the SVM scores $\mathbf{h}$ so that the influence of each potential function is balanced. Second, we add a bias term that acts as an automatic way of measuring the confidence of the head pose classifier. These additional terms are treated as parameters of the potential function and will be learned during a training stage. Taking these changes into consideration, the final potential function is defined as:

$$\Psi_h(\mathbf{X},\mathbf{Y};\boldsymbol{\alpha}) = \sum_{i=1}^{M}(\alpha^0_{\theta_i} h^i_{\theta_i} + \alpha^1_{\theta_i}) \qquad (4)$$

where $\alpha^0_{\theta_i}$ and $\alpha^1_{\theta_i}$ are the scaling and bias terms and $\boldsymbol{\alpha} = [\alpha^0_1 \cdots \alpha^0_D\, \alpha^1_1 \cdots \alpha^1_D]$. The head orientation labels also play a role in the other two potential functions. In this way we could choose a head orientation that doesn't have the highest SVM score, if by doing so the combined score of the potential functions is higher.

Local context potential (Ψ_ℓ). This function takes into account the local context of each person in the frame. We use the previously learned SVM local context interaction classifiers to compute scores $v^i_{y_i\theta_i}$, which represent the score of labeling bounding box i with an interaction label y_i if the label for the head orientation is θ_i. We construct it in a similar way to the head orientation potential by adding scaling and bias terms to get:

$$\Psi_\ell(\mathbf{X},\mathbf{Y};\boldsymbol{\beta}) = \sum_{i=1}^{M}(\beta^0_{y_i\theta_i} v^i_{y_i\theta_i} + \beta^1_{y_i\theta_i}) \qquad (5)$$

where $\boldsymbol{\beta} = [\beta^0_{01} \cdots \beta^0_{(K-1)D}\, \beta^1_{01} \cdots \beta^1_{(K-1)D}]$. Analogous to the head orientation potential, the bias term measures the confidence in the different local context SVM classifiers. For example, these weights could codify that the SVM classifier for hand shakes is more reliable when the discrete head orientation of a person is a profile rather than when it is frontal.

Global context potential (Ψ_g). The third part of the scoring function deals with the global context of the interaction. It is at this point that the relative spatial relations between people are codified and where the configuration label is used. We want to assign higher scores to labels that are congruent with the spatial configuration of people in the scene. This means, for example, that if two people are far from each other and are labeled as interacting (i.e. are part of the set of pairs defined by the configuration label), labeling them as hugging should result in a low score. Also, we want to learn how these spatial relations depend on the head orientation of the people. To encode these constraints, we divide the space around a person's bounding box into $R = 6$ discrete regions as shown in Fig. 5. The size of the regions is proportional to the bounding box size. Given a pair of people (i, j) obtained from the configuration label, each person is taken in turn as the central one to divide the frame space and to compute the discrete relative location of the other. The result of this process is encoded by using an indicator vector $\boldsymbol{\delta}$ with six elements (corresponding to the six discrete spatial regions). We use the notation $\boldsymbol{\delta}_{ij}$ to represent the relative location of person j with respect to person i and $\boldsymbol{\delta}_{ji}$ for the opposite case. A way of scoring a labeling for this pair is to weight the vector $\boldsymbol{\delta}_{ij}$ depending on the interaction label of person i and his head orientation, and correspondingly weighting $\boldsymbol{\delta}_{ji}$. In general we need to define a set of spatial weights for each combination of interaction class and head orientation. We will use the notation $\boldsymbol{\gamma}_{y\theta}$ to denote a vector of spatial weights (corresponding to the six discrete spatial regions) for a given interaction class y and discrete head orientation θ. Effectively what these weights do is to encode common spatial configurations between people for each interaction class. Using the defined notation we can express the global context score by

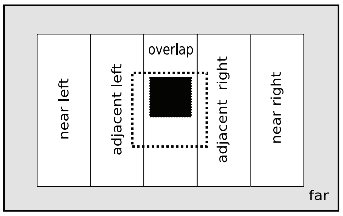

Fig. 5. Discrete set of relative spatial relations between people in a frame. The dotted square represents an upper body bounding box and the black square the head area.

$$\Psi_g(\mathbf{X},\mathbf{Y};\gamma) = \sum_{(i,j)\in P_{y_c}, y_i\neq 0, y_j\neq 0} (\boldsymbol{\gamma}^T_{y_i\theta_i}\boldsymbol{\delta}_{ij} + \boldsymbol{\gamma}^T_{y_j\theta_j}\boldsymbol{\delta}_{ji}) \qquad (6)$$

where P_{y_c} is the set of valid pairs defined by configuration index y_c and $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_{11} \cdots \boldsymbol{\gamma}_{(K-1)D}]$. Combining the potential functions we arrive at the complete scoring function:

$$S(\mathbf{X},\mathbf{Y}) = \Psi_h + \Psi_\ell + \Psi_g \qquad (7)$$

where we have omitted the potential function's parameters for clarity. In Section 5.4 we describe an efficient algorithm to compute the label that maximizes the score function and explain how to learn the parameters $\boldsymbol{\alpha} \in \mathbb{R}^{2D}, \boldsymbol{\beta} \in \mathbb{R}^{2KD}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{(K-1)DR}$.

5.3 The loss function

A key element of a structured learning framework is to define a suitable loss function for the problem in consideration. Here, the loss function should penalize not only wrong assignments of interaction and head pose labels but configuration labels as well. Additionally, it should penalize a label mismatch between detections that are labeled as interacting. Annotation is provided for all of these elements (interaction class, head pose, UB region) on the training and test data, as described in Section 6.1. Taking these elements into consideration, the loss function is defined as:

$$\begin{aligned}\Delta(\mathbf{Y}, \hat{\mathbf{Y}}) = w_h \sum_{i=1}^{M} \Delta_h(\theta_i, \hat{\theta}_i) + w_\ell \sum_{i=1}^{M} \Delta_\ell(y_i, \hat{y}_i) + \\ w_c \left[\sum_{(i) \in I_{y_c}} \Delta_c(i) + \sum_{(i,j) \in P_{y_c}} \Delta_c(i,j) \right]\end{aligned} \tag{8}$$

where $\mathbf{Y}$ is the ground truth labeling, $\hat{\mathbf{Y}}$ is a labeling hypothesis, I_{y_c} and P_{y_c} represent the sets of independent detections and valid pairs spanned by the configuration y_c and w_h, w_ℓ, w_c are positive weights. As with the scoring function, we have divided the loss function into three parts so as to make clear the contribution of different kinds of labeling errors. The first part, Δ_h, measures the error of assigning incorrect head orientation labels. This error is proportional to the difference between the true orientation θ_i and the predicted $\hat{\theta}_i$ (i.e. the penalty for labeling a head orientation as frontal-right when the true orientation is profile-right is less than when labeling it profile-left). We define the loss Δ_h to be 0 if the predicted head orientation is correct, 1 if the discrete pose is adjacent to the true pose and 2 otherwise. The second part of the loss function, Δ_ℓ, takes into account incorrect individual interactions labels. We set this loss to be equivalent to the zero-one loss (Δ_{01}, where $\Delta_{01}(a, b) = 0$ if $a = b$ and 1 otherwise). The last part of the loss function deals with errors derived from an incorrect configuration label assignment, and we set it as:

$$\Delta_c(i) = \begin{cases} 1 & \text{if } (i) \notin I_{\hat{y}_c} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$\Delta_c(i,j) = \begin{cases} 2 & \text{if } (i,j) \notin P_{\hat{y}_c} \\ 1 & \text{if } (i,j) \in P_{\hat{y}_c},\ \hat{y}_i \neq \hat{y}_j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Consider a frame with three people, two of them interacting. A candidate label that assigns an incorrect interaction label to a person who is not interacting will result in a loss of 1 from Δ_ℓ. If this error occurs instead in the labeling of one of the people who are interacting, then the loss will be 2 (1 for the incorrect label in Δ_ℓ plus 1 for assigning different labels to interacting people in Δ_c). The weights w_h, w_ℓ, w_c are set to 1, but this could be changed to assign higher penalties to mistakes made in a specific component of the label. Note, the scoring function is defined as a sum of unary and pairwise terms over tracks. In the following section we take advantage of this structure to find the most violated constraint.

5.4 Inference and Learning

We use the SVM^{struct} package [31] to learn the weights $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$ in a supervised way (i.e. a set of example pairs $(\mathbf{X}, \mathbf{Y})$ is given). First, the scoring function (7) is arranged to the form of (2), making explicit that it is a linear combination of weights and elements of the feature mapping Φ:

$$\begin{aligned} S(\mathbf{X}, \mathbf{Y}) &= \boldsymbol{\alpha}^T \phi_h + \boldsymbol{\beta}^T \phi_\ell + \boldsymbol{\gamma}^T \phi_g \\ &= \underbrace{\begin{bmatrix} \boldsymbol{\alpha}^T & \boldsymbol{\beta}^T & \boldsymbol{\gamma}^T \end{bmatrix}}_{\mathbf{w}^T} \underbrace{\begin{bmatrix} \phi_h \\ \phi_\ell \\ \phi_g \end{bmatrix}}_{\Phi} \end{aligned} \tag{11}$$

where Φ is a feature vector composed of the head pose scores, local context scores and global context relations as were defined in Section 5.2.

For inference, given a new example $\mathbf{X}$ we want to find the label that maximizes $\mathbf{w}^T \Phi(\mathbf{X}, \mathbf{Y})$. Similarly during training, the label that maximizes $\mathbf{w}^T \Phi(\mathbf{X}_i, \mathbf{Y}) + \Delta(\mathbf{Y}_i, \mathbf{Y})$ is needed in order to compute the most violated constraint. Because we are dealing with videos this process has to be repeated thousands of times (once for each frame). This section describes an efficient method for inference with polynomial complexity, which can be easily extended to find the most violated constraint.

In general, for a frame with M people, K interaction classes (including a no-interaction class) and D head orientation classes, the complexity of exhaustive search over possible labels is $N_m \times K^M D^M$ labels, where N_m is the number of configurations for M people (see Section 5.1), i.e. the number of possible labels per configuration increases exponentially with the number of people. For example, consider the task of finding the label that maximizes the scoring function when there are four people present in a frame. In this setting there are 10 valid configurations and the score must be computed for each one of the $10 \times 5^8 \approx 3.9$ million possible labels.

However, due to the structure of the scoring function this search can be made more efficient and the complexity reduced to $O(M^2 K^2 D^2)$, as will now be shown. Assume for the moment that we know the configuration label y_c. Given the configuration label, there are only two options for each detection i: either it's labeled as independent or as part of a pair (i, j).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

If it's labeled as independent, the only parts of the scoring function that contribute to the score for that detection are Ψ_h and Ψ_ℓ. In this case, finding its best interaction and head pose label corresponds to maximizing the following function:

$$f_{ind}^{i}(\mathbf{x}_i, y_i, \theta_i; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \underbrace{\alpha^0_{\theta_i} h^i_{\theta_i} + \alpha^1_{\theta_i}}_{\Psi^i_h} + \underbrace{\beta^0_{y_i\theta_i} v^i_{y_i\theta_i} + \beta^1_{y_i\theta_i}}_{\Psi^i_\ell} \quad (12)$$

Alternatively, if it's labeled as part of a pair (i, j), then we have to take into account the contribution of the global context potential. Therefore, the best interaction and head pose labels are the ones that maximize:

$$f_{pair}^{(i,j)}(\mathbf{x}_i, \mathbf{x}_j, y_i, \theta_i, y_j, \theta_j; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \Psi^i_h + \Psi^j_h + \Psi^i_\ell + \Psi^j_\ell + \underbrace{\boldsymbol{\delta}_{ij} \boldsymbol{\gamma}^T_{y_i\theta_i} + \boldsymbol{\delta}_{ji} \boldsymbol{\gamma}^T_{y_j\theta_j}}_{\Psi^{(i,j)}_g} \quad (13)$$

In general, for a specific configuration y_c, the labeling that maximizes the scoring function is given by:

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmax}} \left[\sum_{i \in I_{y_c}} f^i_{ind} + \sum_{(i,j) \in P_{y_c}} f^{(i,j)}_{pair} \right] \quad (14)$$

where $\mathbf{Y}$ is maximized keeping the value of y_c constant and I_{y_c}, P_{y_c} are defined as above.

So far we have assumed that y_c is known, but, when doing inference, this also needs to be estimated. The process to estimate the complete label $\hat{\mathbf{Y}}$ that maximizes the scoring function is carried out by the following steps. First we compute the $\max$ and argmax of f_{ind} for every detection in the frame. Then we compute the $\max$ and argmax of f_{pair} for every possible pairing of people in the frame. Note that the unary costs Ψ^i_h and Ψ^i_ℓ, already evaluated for computing the independent costs, can be reused when computing the pair costs. Finally, we test all possible configurations and choose the one that maximizes (14) using the previously computed maxima. The first step requires us to compute KD values for each detection, while the second requires K^2D^2 values for each pair. In total we need to compute $MKD + \frac{M(M-1)}{2}K^2D^2$ values, which is no longer exponential in M. Observe that the first two steps only need to be run once and not for every different configuration (a configuration only defines the values of I_{y_c} and P_{y_c} in Equation (14)). Therefore, computing the score for a specific configuration only involves adding previously computed values. This is a crucial point, because the number of configurations is still combinatorial in M, allowing the inference to be efficient for $M \leq 10$.

During learning, this algorithm can be used to find the most violated constraint because the loss function decomposes in a similar way to the scoring function. By adding the corresponding loss components Δ_h, Δ_l, $\Delta_c(i)$ to f_{ind} and $\Delta_c(i, j)$ to f_{pair}, nothing else has to be modified, resulting in a significant speed up of the learning process.

6 Experiments

The individual performance of each stage of the described method has a direct influence on the final results. In this section we present several experiments both to evaluate each stage of our method and to identify places where future improvements could be made. We start by describing the dataset that we use in our experiments, followed by an evaluation of the upper body detector, track estimation process and head pose classifier. We then move to test the performance of the local context descriptor and the contribution of using global cues, comparing to our results with a baseline. Finally we test our method with a standard benchmark dataset (UT-Interaction) that contains six interaction classes.

6.1 TV Human Interaction dataset (TVHI)

The TVHI dataset is composed of 300 video clips compiled from 23 different TV shows [32]. 200 of the clips contain one of four interactions: *hand shake*, *high five*, *hug* and *kiss* (each appearing in 50 videos). Negative examples (clips that don't contain any of the interactions) make up the remaining 100 videos. The length of the video clips ranges from 30 to 600 frames. The interactions are not temporally aligned (i.e. a clip containing a hand shake might start with people walking towards each other or directly at the moment of the hand shake). There is a great degree of variation between different clips and also in several cases within the same clip. Such variation comprises the number of actors in each scene, their scales and the camera angle, including abrupt viewpoint changes at shot boundaries.

Annotations. To have a ground truth for the training and the evaluation of the methods developed in this paper, every frame of each video is annotated with the following information: the location of the people present (with bounding boxes framing their upper bodies), their discrete head orientations, and their interaction label. This was done for those people whose upper body size was within a certain range. This range goes from far shots that show the whole body to medium shots where only the upper body is visible and is equivalent to 50-350 pixels in our videos. Also the pairs of people who are interacting, if any, are annotated in each frame.

Data split. For the purposes of avoiding, as much as possible, biases towards specific actors, scenarios or shooting styles during training and testing, the dataset is split evenly into two groups, each containing videos of mutually exclusive TV shows. Each group, contains 25 video clips of each interaction and 50 negative clips.

6.1.1 Upper body detector and track estimation

The aim of this section is to evaluate the precision and consistency with which we can locate people. To this end we compare the performance of using the raw upper body detections (UBD) and of using automatically computed tracks.

In order to train and test the track classifier, we computed tracks in each video from the raw detections obtained. Using the dataset's bounding box annotations, the generated tracks were split into positive and negative examples for training. We trained two track classifiers, one for each split of the dataset (see previous section). During the testing phase, a classifier trained with examples from one half of the dataset is used to classify the tracks of the other half. This same setting of training classifiers for each partition of the data is used in all the following experiments.

There are approximately 40k upper body ground truth annotations in our dataset. In a typical object recognition evaluation framework [33], a detection is considered a true positive if the value of the $overlap = intersection/union$ of its bounding box with a ground truth annotation is ≥ 0.5. Because we need to estimate the head orientation inside a detection's bounding box, this measure is too loose, and we consider a true positive only if the overlap is ≥ 0.7. In order to assess the relative performance raw UBD vs UBD after tracking, all bounding boxes belonging to tracks classified as true tracks are treated as positive upper body detections, which are then marked as true or false positives according to their overlap with the ground truth annotations.

Fig. 6 shows the precision-recall curves computed using the bounding boxes obtained from the raw upper body detector and the bounding boxes belonging to tracks generated using only clique partitioning (CP) or the KLT-CP combination (see Section 3). Also displayed are the results obtained by the CP tracks generation method of [5], for comparison. It can be seen that at both overlap thresholds the computation of tracks increases the precision. There is a marked improvement over [5] due to the addition of a track classifier for the selection of correct tracks instead of the heuristic approach of [5]. Computing tracks also increases the recall because, after linking the detections during the track generation process, any gaps in the resulting tracks (which usually correspond to missed detections) are filled by interpolation.

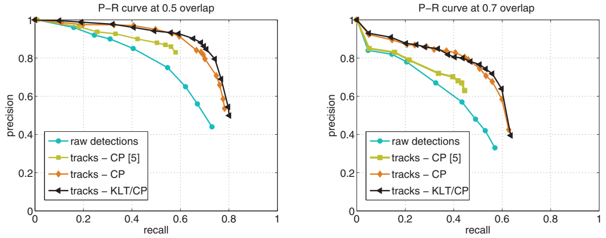

Fig. 6. Performance of the upper body detector and tracks estimation. Precision-recall curve of the upper body detector (raw), generated tracks using the CP method of [5] and our new approach (CP only or KLT-CP combination). Results are given for two overlap thresholds. Computing tracks improves the precision and recall in both cases.

The previous experiment showed results at the person detection level, but it doesn't give an idea of the quality or usefulness of the computed tracks for interaction detection. With the upper body ground truth annotations, we can compute 967 tracks of people in the whole dataset. When using the KTL-CP method before track classification, 2099 tracks are generated, 821 of which are true positives. After classification we end up with a total of 644 correctly classified tracks and 78 false positives. Two other results are relevant in the context of the experiments presented in Section 6.1.4. First, of the 300 videos in our dataset, 58 contain at least one false positive track (19.3%). Second, simultaneous tracks of both people performing an interaction are potentially available (using the ground truth) for 178 of the 200 videos that contain interactions. Of these 178, the KLT-CP track estimation and classification methods obtain 79 (44.4%). How these two results influence the structured prediction method is discussed in Section 6.1.4.

6.1.2 Head pose classifier

As mentioned in Section 3, we learn a set of five one-vs-all linear SVM classifiers (corresponding to five discrete head orientations) using HOG descriptors. For training, we manually crop head regions from disjoint TV shows' frames, which are then normalized to 80×80 pixels. Several transformations are applied to this initial set, including small rotations, scale deformations, additive and Gaussian noise and jitter in order to create a final training set of $\approx$ 20k examples.

To evaluate the accuracy of the head pose classifier, we use the manual annotations of the dataset to select all automatically computed tracks that overlap with the ground truth. From the bounding boxes of these tracks we extract a head region (Fig. 3a), which is then classified. In total we classify 42,762 bounding boxes. Fig. 7 shows the confusion matrix between head poses and the per-class percentage of correctly classified head orientations (the discrete label of a head region is taken as the highest SVM score) before and after the smoothing. Using smoothing increases the overall accuracy from 69.05% to 72.03%, while the lowest performance occurs in bounding boxes extracted from hug videos; this is probably because the heads present in these videos are sometimes partially occluding each other. Confusion occurs between head poses that are close in angle distance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

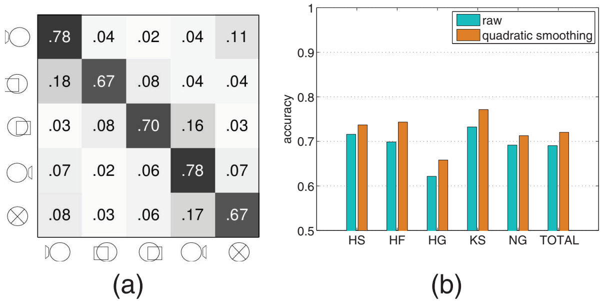

Fig. 7. Performance of the discrete head pose classifier. (a) Confusion matrix. (b) % of correctly classified head poses, before and after applying a quadratic smoothing, for each interaction class: hand shake (HS), high five (HF), hug (HG), kiss (KS), negative (NG).

6.1.3 Evaluating the local context descriptor

In this experiment we analyse the contribution of each component of the local context descriptor when learning interaction classifiers. In particular we test the effect of *(i)* not using head orientation information *vs* adding it, either by manual annotation or by automatic classification; *(ii)* changing the descriptor: using only motion, only gradients, or both.

To be able to compare the results obtained, all of the experiments follow the next steps. From each clip, we manually selected five consecutive frames, which are inside the temporal region where the interaction is happening. From these, local context descriptors are extracted from tracks of people performing the interaction. The same process is applied to the negative videos. As described in Section 6.1, the dataset is divided into two sets for training and testing. The descriptors of each set are used in turn to train a one-vs-all linear SVM classifier for each interaction in a supervised way.

Table 1 shows the results obtained for different settings of the local context descriptor (LCD). Several things can be concluded from this representation. First, it can be readily observed that the use of head orientation improves the average classification accuracy when correctly estimated, but errors when automatically classifying the head orientation reduce it. This improvement can be consistently noted independently of the local context descriptor used (motion, gradients or both).

Table 2 gives a closer look at the per-class accuracy of the LCD classifier in the case where the head pose was manually annotated. An easily distinguishable characteristic of the classifier, made clear from this table, is that the use of motion features alone has better performance when classifying high fives and kisses, while a combination of both works better for hugs. This is very intuitive because hugs contain minimal motion in contrast to the other actions. The bad performance of hand shake and high fives when using only gradient information could be explained by the coarseness of the descriptor, which results in learning gradients that are too general to be distinctive in these cases. This is particularly evident for hand shakes where the pose of the body is a common pose that can be easily confused. Note, increasing the number of cells in the grid was found not to improve the results. The increased size of the descriptor, combined with a reduced number of pixels for training each cell, led to worse classification results.

TABLE 1
Avg. accuracy for different settings of the Local Context Descriptor (LCD) and how the head pose was computed (rows).

Head Pose \LCD	Motion	Gradients	Both
manual	41.7 %	43.2 %	48.9 %
auto	41.0 %	39.7 %	46.8 %
no	35.3 %	31.7 %	40.4 %

TABLE 2
Accuracy per class when using manual head pose.

LCD	HS	HF	HG	KS	NG
motion	42.7 %	64.3 %	22.4 %	14.5 %	64.5 %
gradient	24.5 %	33.3 %	38.2 %	39.8 %	80.5 %
both	29.6 %	61.7 %	30.6 %	39.8 %	83.0 %

6.1.4 Evaluating global descriptors for video retrieval

In this experiment, a comparison is made in the context of video retrieval, between independently classifying the upper body bounding boxes in a frame or employing structured learning for their joint classification. We also analyze, in the case of using structured prediction, the contribution of each potential function. Based on the previous results, we use a local context descriptor that is composed both of motion and gradients. For the purposes of retrieval the task is defined with positives at the clip level. So, for example, the perfect retrieval score (average precision =1) for hand-shakes would be if all the hand-shake clips in the test data are retrieved first, and similarly for the other interactions. For this we need a score for each clip in order to rank them.

Baseline. As a baseline for comparison with our method, we use STIP features [7] in a standard Bag of Words (BoW) approach. From a training set of videos, we sampled approximately 60,000 of these features. The descriptors of these features (composed of histograms of gradients and histograms of optic flow) were then clustered using k-means to compute two vocabularies of 1000 and 2000 visual words respectively. Each feature in a video is assigned the label of the closest cluster center (in Euclidean distance), and the whole video is represented by a normaliszed histogram of visual word occurrences. These histograms are used to learn a linear multi-class SVM classifier in a supervised way. We employed the SVM^{multiclass} package provided by Joachims [34].

TABLE 3
Average precision results for the video retrieval task.

	Tracks	Head pose	Struct	HS	HF	HG	KS	AVG
1	manual	manual	-	0.4625	0.4381	0.5415	0.4911	0.4833
2	manual	manual	Ψ_ℓ, Ψ_g	0.4132	0.4306	0.6608	0.6857	**0.5476**
3	manual	auto - fixed	-	0.4781	0.3643	0.5711	0.4463	0.4649
4	manual	auto - fixed	Ψ_ℓ, Ψ_g	0.4374	0.3531	0.6032	0.4910	0.4712
5	manual	auto - struct	$\Psi_\ell, \Psi_h, \Psi_g$	0.3727	0.4457	0.6699	0.5414	**0.5074**
6	auto [5]	auto - fixed	-	0.3981	0.2745	0.3267	0.2613	0.3151
7	auto	auto - fixed	-	0.3898	0.4156	0.5463	0.3205	0.4181
8	auto [5]	auto - fixed	Ψ_ℓ, Ψ_g	0.3517	0.2569	0.3769	0.3250	0.3276
9	auto	auto - fixed	Ψ_ℓ, Ψ_g	0.3737	0.4470	0.5088	0.3473	0.4192
10	auto	auto - struct	$\Psi_\ell, \Psi_h, \Psi_g$	0.3935	0.4582	0.4699	0.3760	**0.4244**
11	Baseline SVM - 1000 codewords			0.3668	0.2171	0.2485	0.2668	0.2748
12	Baseline SVM - 2000 codewords			0.3204	0.2791	0.2541	0.2067	0.2651

Scoring videos. For a retrieval task, we need a way of evaluating the performance of our method when applied to all the frames of a video clip. We propose to compute in each video i an interaction score v_k^i for each interaction class k (this score represents the likelihood that a given interaction is present in the video). These interaction scores are based on the classification of each track extracted from the clip. In each frame, a bounding box belonging to a track is classified either independently (by using only its local context) or by using the structured learning framework of Section 5.

Each track j in video i is assigned a score t_k^{ij} for each interaction class k as the ratio between the number of its bounding boxes that were classified as interaction k (denoted b_{jk}) and the total number of its bounding boxes that were classified as any interaction, i.e. $t_k^{ij} = b_{jk} / \sum_{k'}^{K} b_{jk'}$; this excludes bounding boxes labeled as no-interaction ($k = 0$) and bounding boxes that are marked as occluded during the track estimation stage. We don't consider the length of the track as a normalizing factor because, as part of the pre-processing stage, short tracks are eliminated and the average length of the remaining tracks inside a clip is roughly similar. The interaction scores for clip i are computed by adding the tracks' scores:

$$v_k^i = \sum_{j=1}^{T} t_k^{ij} \quad k = 1, ..., K$$

where T is the number of estimated tracks in the clip and K is the number of interaction classes (in our case $K = 4$). The videos are ranked by using the scores v_k^i.

Results. A summary of the average precision (AP) results obtained for the video retrieval task is shown in Table 3. The table reports 12 experiments: the first 10 in which we test different components of our method, plus the two baselines described previously. The first column of the table specifies if the tracks in the video were computed from manually annotated upper bodies or by the automatic method described in Section 3. The second column refers to the method used to compute the head orientation. This can be done in three different ways: with manual annotations, using the head pose classifier directly (auto), or letting the structured prediction choose the head pose (struct). The third column refers to the potential functions used for structured prediction.

Despite the substantial challenges of the task, the results obtained fall within those generated by state-of-the-art methods in single-action recognition that use similar datasets [7], [9], [10], [35]. It can be observed that the baseline multi-class SVM classifiers (experiments 11 and 12) performed poorly even compared to the fully automatic method. This could be because of the small number of training examples (25 per interaction class). Coupled with the size of the training set, other problems for learning representative interaction models based on BoW might arise from the variance in the dataset, background and camera motion, and from the fact that the interaction doesn't necessarily occur in the whole video.

The first part of the table (experiments one and two) shows the results of the ranking under perfect conditions, namely all the tracks and head poses are correct (using manual annotations). In this case the structured learning method gives the largest improvement. Experiments three to five report results when the head pose is computed automatically. It can be observed that this reduces the average precision compared to the manual case, as is expected. Using structured prediction without the head orientation potential (Ψ_h) only performs slightly better than the independent method. Even if the global context potential (Ψ_g) can help improve the classification in some cases, introducing a wrong head orientation will result in worse results. By including Ψ_h in the structured prediction framework, we add some robustness against this kind of error. The results of using the fully automatic method for generating tracks and computing head orientations are shown in the third part of the table. Although the AP is not as high as with the manual case, it is important to note that

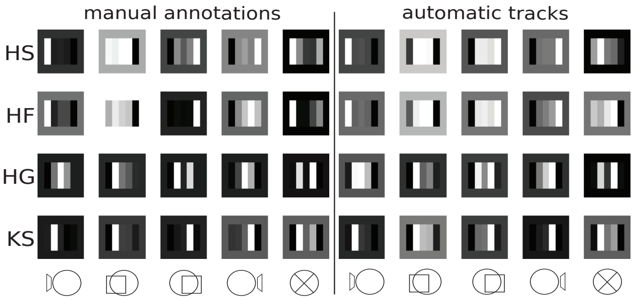

Fig. 8. Spatial weights (γ, see Fig. 5) learned using training data from manual annotations and automatically generated tracks, corresponding to experiments two and ten in Table 3. Lighter intensity indicates a higher weight.

by using our modified method for generating tracks (KLT-CP) we have obtained a significant improvement from the results reported in [5] (AP = 0.3276 for the automatic case using structured prediction). Several reasons could account for the small improvement of 0.63% showed by using structured prediction in the fully automatic case. One observation arising from the results shown in Section 6.1.1 is that, although the KLT-CP process for generating tracks is more reliable than the one used in [5], we only end up with tracks of both people interacting in a video less than 50% of the time. This reduces the beneficial effects of using global context. Also, there are still a number of tracks created from false positive detections that aren't eliminated by the track classifier. These tracks have a negative influence on the classification of correct tracks due to the joint nature of the classification produced by the structure prediction method. An illustration of the global context weights (γ) learned for experiments two and ten is shown in Fig. 8. It can be seen that the learned weights correspond to the spatial arrangements that we would expect for each interaction class, i.e. people appear closer to each other when hugging or kissing and further away for hand shakes and high fives. The location of people is also correlated with the head orientation.

Fig. 9 shows the best ranked videos returned for each interaction, generated by the fully automatic method using structured prediction (corresponding to experiment 10 of Table 3). Looking at the results it can be seen that the top ranked video is correct for all interaction classes, and that in two classes,*hand shake* and *high five*, the first five ranked videos are correct. However, each interaction class has its problems. Hand shakes are difficult to classify because, as shown in Section 6.1.3, they rely mostly on local motion information. The natural motion of the arms of a person can be easily confused with the motion of a hand shake. Adding camera, background and other independent motion (e.g. people walking near by) complicates the task even more. The same problems can be observed for high fives. Hugs and kisses tend to be confused with each other due to their very similar appearance and spatial information. It is very common to observe the frames immediately before a hug happens to be classified as kisses (here we see the effects of the global context for the classification).

6.2 UT-Interaction dataset

The UT-Interaction dataset was first used in [15] and has become a benchmark for human interaction recognition. The dataset contains examples of six interactions: *hand shake*, *hug*, *kick*, *point*, *punch* and *push*. It is divided into two sets of 10 video sequences. From each set, 60 shorter clips have been extracted containing a single interaction (six clips were extracted from each longer sequence). To compare our approach with previous works we evaluate the classification accuracy of our method on these shorter clips following the framework described in [36] for this task, which is a 10-fold leave-one-out cross validation (where the clips extracted from one of the sequences are left out and training is performed on the clips extracted from the other nine sequences). The clips corresponding to the class *point* only contain one person; because we are interested in evaluating our method for pair-wise interactions we exclude these clips for our experiments.

Modifications. Some minor modifications were made to the human detection and representations in order to apply them to this dataset. First, this dataset contains entire humans (not just upper bodies) viewed from slightly above. The upper body detector employed for the TVHI Dataset is not suitable for this viewpoint and does not detect a person reliably when their head dips below their shoulders, which is the case in several interactions in this dataset (e.g. kick, punch, push). Instead, a head detector is used in conjunction with a full person detector [37], [38].

Second, the local context region is expanded, from covering only the region around the upper body, to include the lower part of the body as well. Therefore we moved from a 8×8 grid to a 12×16 one.

Third, because the method described in Section 5 was designed to handle symmetrical interactions (i.e. where the local context is the same for both people interacting), the formulation must be changed in order to take advantage of the global context information in asymmetric interactions like punching or kicking. To tackle the asymmetry problem, we define a new class for each one of the asymmetric interactions; these classes are named: *being kicked*, *being punched* and *being pushed*. Because now an interaction could be composed of people having different interaction labels, we remove from $\Delta_c(i,j)$ (Eq. 10) the penalty that inhibited this behaviour. Instead, a *compatibility* term is added to the global context potential Ψ_g (Eq. 6). This extra term learns weights for pairings of interaction labels that were seen often in the training examples, resulting in the extended potential:

$$\Psi'_g = \Psi_g + \sum_{(i,j)\in P_{y_c}, y_i\neq 0, y_j\neq 0} \rho_{y_i y_j} \qquad (15)$$

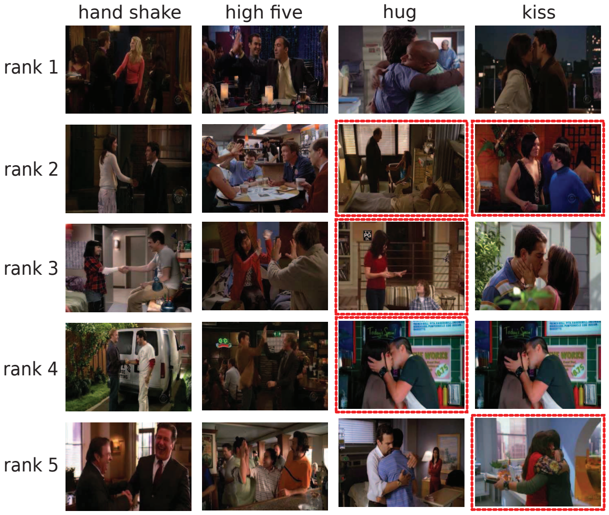

Fig. 9. Highest ranked videos divided by interaction class obtained using the structured prediction method with automatically computed tracks. The red squares indicate false positives.

Finally, with the purpose of following the same training procedure described for the TVHI Dataset, we have annotated the short clips with upper bodies and their respective head pose and interaction label (including the extra interaction classes and a no-interaction class). During testing we follow the scoring scheme described in Section 6.1.4, with the difference that bounding boxes labeled as one of the extra interaction classes vote for their complementary class (e.g. *being punched* votes for the class *punch*).

Results. The computation of automatic tracks recovered 97% of the people performing the interaction in the first set and 96% in the second. In total there were only four false positive returned tracks. Classification accuracy results per interaction class are shown in Table 4. This table includes results for our automatic method when using only the local context (LC) and when using the full structured approach (FULL). Also shown are the top two results reported in [36] for comparison (the baseline uses a bag of words of Cuboid features [39] combined with an SVM classifier). It can be seen that our method performs well compared with the state of the art, achieving 84% and 86% classification accuracy for sets one and two respectively. In analysing the results, we noticed that the interactions *punch* and *push* are often confused (this was also reported in [36]) as well as *being kicked*, *being punched* and *being pushed*. The confusion in the added classes is often corrected when using the structured method because the interaction class of both people is taken into account. This is not the case when only the local context is employed, which explains the low results obtained in this case. Also worth noticing is that, in our approach, a no-interaction class is defined which increases the number of possible labels for a video clip.

The results also show a marked difference between the classification accuracy obtained for each set, which is more evident in the baselines. Set two has a less homogeneous background than set one and small camera motions. Around half of the clips of set two contain parts of other people not involved in the interaction (this is because the clips were cropped from larger videos containing multiple people). These characteristics of set two introduce extra noise for methods that use spatio-temporal features, and could partly explain the decrease of performance of the baseline. Another, more subtle difference, is the way some interactions are performed. In several instances of the interaction *punch* in set one, the person *being punched* doesn't react (stays almost still). This is an explanation for why in set one the full model does not improve on the local context for punch, but it does in set two where the person being punched *does* react.

7 Conclusions and Future Work

In this paper we have presented a method for learning two-person human interactions from TV shows. Our method combines local and global information using a structured prediction approach. Both local and global descriptors of the interaction make use of people's head orientation, which we compute automatically. We added robustness to errors in the head pose estimation by incorporating it as part of the structured model. A more robust method for generating people's tracks (KLT-CP) was described and showed that both modifications result in improvements over previous results. We outlined an efficient way of doing inference on our model, which reduces substantially the search space over the number of possible labels (which is exponential in the number of people if using a naive brute force method). We described how our method can handle asymmetric interactions and showed experiments on a benchmark dataset.

A clear improvement can be obtained by using a better upper body detector and more reliable tracks – this would reduce the gap between the manual and automatic results. This is of particular importance if the method were to be used in more crowded surveillance scenarios. In these environments, tracking approaches like the one presented in [40] have shown a very good performance. Also, our model looks for interactions on a frame by frame basis, and no explicit effort is made to maintain any kind of temporal consistency of the labels across consecutive frames of a video. A natural extension to this work would be to incorporate this temporal consistency into the structured model. Introducing long-term temporal features (e.g. spatio-temporal global context) could also help to eliminate ambiguity inherent in a frame by frame classification.

Finally we note that the reaction to an action can be more varied than the action itself. For example, the action of *punching* is fairly constrained but the *reaction* to the punch can take many forms (e.g. moving the

TABLE 4
Classification Accuracy, UT-Interaction Dataset.

SET 1				
Class	Baseline	BIWI [41]	ours LC	ours FULL
Hand shake	80 %	70 %	90 %	100 %
Hug	90 %	100 %	50 %	100 %
Kick	90 %	100 %	50 %	80 %
Punch	70 %	70 %	60 %	60 %
Push	80 %	90 %	100 %	80 %
AVG	82 %	86 %	70 %	84 %

SET 2				
Class	Baseline	BIWI [41]	ours LC	ours FULL
Hand shake	80 %	50 %	80 %	90 %
Hug	80 %	90 %	100 %	90 %
Kick	60 %	100 %	50 %	90 %
Punch	70 %	80 %	10 %	70 %
Push	40 %	40 %	90 %	90 %
AVG	66 %	72 %	66 %	86 %

head, falling away, moving backwards). A more flexible modeling of multimodal interactions constitutes an interesting direction for future research.

References

[1] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *Conference on Computer Vision and Pattern Recognition*, 2005.

[2] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, “Progressive Search Space Reduction for Human Pose Estimation,” in *Conference on Computer Vision and Pattern Recognition*, 2008.

[3] A. Kläser, M. Marszalek, C. Schmid, and A. Zisserman, “Human Focused Action Localization in Video,” in *SGA*, 2010.

[4] B. Benfold and I. Reid, “Guiding visual surveillance by tracking human attention,” in *British Machine Vision Conference*, 2009.

[5] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. Reid, “High Five: Recognising human interactions in TV shows,” in *British Machine Vision Conference*, 2010.

[6] I. Laptev and P. Perez, “Retrieving Actions in Movies,” in *International Conference on Computer Vision*, 2007.

[7] M. Marszalek, I. Laptev, and C. Schmid, “Actions in Context,” in *Conference on Computer Vision and Pattern Recognition*, 2009.

[8] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action MACH. A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition,” in *Conference on Computer Vision and Pattern Recognition*, 2008.

[9] J. Liu, J. Luo, and M. Shah, “Recognizing Realistic Actions from Videos ”in the Wild”,” in *Conference on Computer Vision and Pattern Recognition*, 2009.

[10] X. Wu, C. W. Ngo, J. Li, and Y. Zhang, “Localizing Volumetric Motion for Action Recognition in Realistic Videos,” in *ACM International Conference on Multimedia*, 2009.

[11] N. Oliver, B. Rosario, and A. Pentland, “Graphical Models for Recognizing Human Interactions,” in *International Conference on Neural Information and Processing Systems*, 1998.

[12] S. Park and J. K. Aggarwal, “A hierarchical Bayesian network for event recognition of human actions and interactions,” *Multimedia Systems*, vol. 10, no. 2, pp. 164–179, 2004.

[13] ——, “Simultaneous tracking of multiple body parts of interacting persons,” *Computer Vision and Image Understanding*, vol. 102, no. 1, pp. 1–21, 2006.

[14] M. S. Ryoo and J. K. Aggarwal, “Recognition of Composite Human Activities through Context-Free Grammar based Representation,” in *Conference on Computer Vision and Pattern Recognition*, 2006.

[15] ——, “Spatio-Temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities,” in *International Conference on Computer Vision*, 2009.

[16] F. Yuan, V. Prinet, and J. Yuan, “Middle-Level Representation for Human Activities Recognition: the Role of Spatio-temporal Relationships,” in *ECCV Workshop on Human Motion*, 2010.

[17] J. K. Aggarwal and M. S. Ryoo, “Human Activity Analysis: A Review,” *ACM Computing Surveys*, vol. 43, no. 3, 2011.

[18] B. Taskar, C. Guestrin, and D. Koller, “Max-margin Markov networks,” in *Neural Information Processing Systems Conference*, 2003.

[19] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large Margin Methods for Structured and Interdependent Output Variables,” *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[20] M. Blaschko and C. Lampert, “Learning to Localize Objects with Structured Output Regression,” in *European Conference on Computer Vision*, 2008.

[21] ——, “Object localization with global and local context kernels,” in *British Machine Vision Conference*, 2009.

[22] C. Desai, D. Ramanan, and C. Fowlkes, “Discriminative models for multi-class object layout,” in *International Conference on Computer Vision*, 2009.

[23] Y. Wang and G. Mori, “A Discriminative Latent Model of Object Classes and Attributes,” in *European Conference on Computer Vision*, 2010.

[24] T. Lan, Y. Wang, W. Yang, and G. Mori, “Beyond actions: Discriminative models for contextual group activities,” in *Neural Information Processing Systems Conference*, 2010.

[25] J. C. Niebles, C. W. Chen, and L. Fei-Fei, “Modeling temporal structure of decomposable motion segments for activity classification,” in *European Conference on Computer Vision*, 2010.

[26] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, “Pose Search: retrieving people using their pose,” in *Conference on Computer Vision and Pattern Recognition*, 2009.

[27] B. Lucas and T. Kanade, “An Iterative Image Registration Technique with an application to Stereo Vision,” in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[28] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Carnegie Mellon University, Tech. Rep. CMU-CS-91-132, 1991.

[29] M. Everingham, J. Sivic, and A. Zisserman, “Taking the bite out of automatic naming of characters in tv video,” *Image and Vision Computing*, vol. 27, no. 5, pp. 545–559, 2009.

[30] B. Benfold and I. Reid, “Colour Invariant Head Pose Classification in Low Resolution Video,” in *British Machine Vision Conference*, 2008.

[31] T. Joachims, T. Finley, and C. Yu, “Cutting plane training of structural SVMs,” *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[32] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman, “TV Human Interaction Dataset,” http://www.robots.ox.ac.uk/~vgg/data/tv_human_interactions, 2010.

[33] M. Everingham, L. Van Gool, C. K. I. Villiams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes (VOC) Challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[34] T. Joachims, “Multi-class support vector machine,” http://svmlight.joachims.org/svm_multiclass.html, 2008.

[35] A. Gilbert, J. Illingworth, and R. Bowden, “Fast Realistic Multi-Action Recognition using Mined Dense Spatio-temporal Features,” in *International Conference on Computer Vision*, 2009.

[36] M. S. Ryoo, C.-C. Chen, J. K. Aggarwal, and A. Roy-Chowdhury, “An Overview of Contest on Semantic Description of Human Activities 2010,” in *International Conference on Pattern Recognition Contests*, 2010.

[37] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, 2010.

[38] P. Felzenszwalb, R. Girshick, and D. McAllester, “Discriminatively Trained Deformable Part Models, Release 4,” http://people.cs.uchicago.edu/~pff/latent-release4/, 2010.

[39] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *VS-PETS*, 2005, pp. 65–72.

[40] B. Benfold and I. D. Reid, "Stable Multi-Target Tracking in Real-Time Surveillance Video," in *Conference on Computer Vision and Pattern Recognition*, 2011.

[41] D. Waltisberg, A. Yao, J. Gall, and L. Van Gool, "Variations of a Hough-Voting Action Recognition System," in *International Conference on Patter Recognition contest on Semantic Description of Human Activities*, 2010.

Alonso Patron-Perez is a postdoctoral researcher at the University of Oxford from where he received a D.Phil. degree in 2011. Previously he was awarded a degree in Computer Systems Engineering from the Technological Institute of Merida and the MSc in Mathematics from the Autonomous University of Yucatan, Mexico in 2004 and 2006 respectively. His research interests focus on different topics of computer vision and machine learning.

Marcin Marszalek Marcin Marszalek received his MSc degree in computer science form Warsaw University of Technology and his PhD degree from Institut National Polytechnique de Grenoble in 2008. He was a postdoctoral researcher at INRIA Grenoble and at the University of Oxford until 2010. He has produced a number of technical publications in highest ranked conferences and journals in computer vision. He is currently employed by Google and works for YouTube

Ian Reid is a Professor of Engineering Science at the University of Oxford. He received a BSc in Computer Science and Mathematics with first class honours from University of Western Australia in 1987 and was awarded a Rhodes Scholarship in 1988 in order to study at the University of Oxford, where he obtained a D.Phil. in 1991. Since then he has been employed in the Robotics Research Group conducting research in computer vision, including holding an EPSRC Advanced Research Fellowship (1997-2000), and has been a University Lecturer since 2000. In 2005 he was awarded title of Reader and in 2010 the title of Professor. His research interests include active vision, visual navigation, visual geometry, human motion capture and intelligent visual surveillance, with an emphasis on real-time aspects of the computations. He has published 140 papers on these topics in major journals and refereed conferences.

Andrew Zisserman is the Professor of Computer Vision Engineering at the Department of Engineering Science, University of Oxford.