EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Hands-on: deformable pose and motion models for spatiotemporal localization of fine-grained dyadic interactions

Coert van Gemeren*, Ronald Poppe and Remco C. Veltkamp

Abstract

We introduce a novel spatiotemporal deformable part model for the localization of fine-grained human interactions of two persons in unsegmented videos. Our approach is the first to classify interactions and additionally provide the temporal and spatial extent of the interaction in the video. To this end, our models contain part detectors that support different scales as well as different types of feature descriptors, which are combined in a single graph. This allows us to model the detailed coordination between people in terms of body pose and motion. We demonstrate that this helps to avoid confusions between visually similar interactions. We show that robust results can be obtained when training on small numbers of training sequences (5–15) per interaction class. We achieve AuC scores of 0.82 with an IoU of 0.3 on the publicly available ShakeFive2 dataset, which contains interactions that differ slightly in their coordination. To further test the generalization of our models, we perform cross-dataset experiments where we test on two other publicly available datasets: UT-Interaction and SBU Kinect. These experiments show that our models generalize well to different environments.

Keywords: Interaction detection, Dyadic interactions, Spatiotemporal localization, Social behavior, Video analysis

1 Introduction

We focus on detecting fine-grained human interactions in videos. The automated localization of such social behaviors has a wide range of applications in video search, automated video captioning, and surveillance. Modeling human interactions from videos is motivated by these applications and has gained significant research interest in recent years. Starting from the analysis of individuals performing particular actions in isolation [1, 2], to actions constrained by the context in which they occur [3], researchers have begun to direct their attention to the analysis of human-human interactions [4, 5] and group actions [6].

In this work, we focus on two-person (*dyadic*) interactions such as shaking hands, passing objects, or hugging. The type of interaction in which people engage informs us of the relationship between them, their activity, and the social and cultural setting in which the interaction takes place. For example, we can use this information to differentiate between friendly and hostile interactions or to determine whether a person in an elderly home is a staff member, family member, or unrelated visitor.

Detecting such social interactions in videos involves several subtasks. First, we need to localize the people involved. Second, we need to identify who interacts with whom. Third, for each interaction, we need to determine the start and end. And finally, we need to assign the correct class label. These subtasks present challenges. Variations in appearance due to factors such as lighting conditions or differences in clothing can make localization of people more difficult. In addition to these environmental factors, the variation in viewpoints from which interactions can be observed makes the introduction of a single visual representation of the interaction challenging. Important parts of the interaction can be occluded from certain viewpoints, which make it harder to determine who are interacting.

We also face challenges in dealing with the performance of the movement. The visual appearance of different interactions can be similar, which might cause confusions in the classification of interactions. Finally, dealing with temporal

* Correspondence: C.J.vanGemeren@uu.nl
Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, 3584 CC Utrecht, Netherlands

variations between instances of the same interaction presents difficulties in marking their start and end points.

A large body of work has emerged to address these challenges. In recent years, significant progress has been achieved to make each of these subtasks more robust and accurate [7]. However, solving each subtask independently is unlikely to give the best results. Errors made early on, for instance in the person detection, impact the final classification because each step depends on the previous.

To overcome this issue, we propose a solution that addresses all subtasks of the interaction classification problem simultaneously. We look at the body configurations of the people involved. The solution we propose benefits significantly from information on limb positions and movements. We share this view with Jhuang et al. [8], who have shown that the precision in action classification improves with greater accuracy of the limb estimations. Van Gemeren et al. [9] demonstrate that pose or movement alone is typically not sufficient to distinguish between similar interactions. In the example in Fig. 1, the poses during the handshake and passing an object look similar, whereas the motion is different. On the other hand, the movement of passing an object and a fist bump can both be characterized by two right hands moving toward each other. In this case, the poses are different. Therefore, we model both pose and motion of the body parts that are representative for the interaction. Our models are based on the deformable parts models (DPM) introduced by Felzenszwalb et al. [10]. To model the temporal extent of the interactions, the detection responses are accumulated over time to generate spatiotemporal localization tubes.

We focus on human interactions that have a moment of coordinated contact between the two individuals. Our work is aimed at distinguishing between interactions that vary subtly. Therefore, we model the characteristic pose and motion of relevant body parts at a fine-grained level. The output of our method is a set of spatiotemporal localization tubes with an assigned interaction label. Our end-to-end framework for spatiotemporal interaction localization aids in the automated analysis of videos. Instead of just recognizing that a certain interaction takes place in the video, we recover where and when it takes place.

We apply our method in a variety of settings. To this end, we train our models on a modest number of training videos. We run cross-dataset experiments to test the generalization of our trained models.

We make three main contributions. First, we present a novel spatiotemporal DPM that supports both pose and motion features per part, which enables us to train fine-grained models that can detect interactions which vary only subtly. Second, we introduce an end-to-end framework to localize human interactions from video in both space and time. Third, we show the efficacy of our work in spatiotemporal localization experiments both on a single dataset, and in a cross-dataset scenario.

In the next section, we will discuss related work, followed by a detailed explanation of our method in Section 3. In Section 4, we detail our experiments, after which we discuss the results in the Section 5. We conclude in Section 6.

2 Related work

Detecting and classifying human actions in videos has attracted a lot of interest over the past decade. The type of actions that have been considered typically involve a single person performing actions such as walking, running, and hand waving [1]. More recently, researchers have started to analyze the behavior of multiple people, in groups or pairs [6, 11]. We focus on the localization of two person interactions that are characterized by the coordination of the movements and poses of both individuals. We distinguish between approaches that classify interactions directly from distributions of image features and those that first detect faces or bodies and then classify the interaction.

The first efforts that were made consider gross body movements and employed bag of visual words (BoVW)

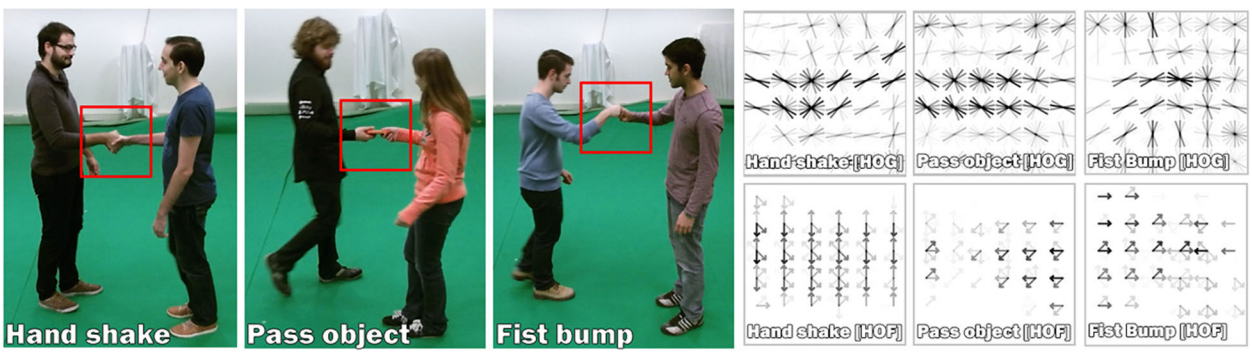

Fig. 1 Interactions that differ slightly in pose (HOG) or movement (HOF). The red bounding boxes show the areas of the right hands, with part descriptors shown on the right

or vector of locally aggregated descriptors (VLAD). Local features are pooled in a region and a mapping is learned from feature distributions to semantic action labels [12]. To be more robust and to take into account the local movement of image features, Wang et al. [13] introduce dense trajectories of keypoints. A dense trajectory is defined as a sequence of keypoint locations over time. At each point in the trajectory, local gradient and motion features are calculated, using histograms of oriented gradients (HOG), histograms of optical flow (HOF), and motion boundary histograms (MBH) descriptors. These features can be encoded with a BoVW, VLAD, or using Fisher vectors (FV) [14].

The main advantage of this approach is that accurate localization of a person is not required. The approach is successful in creating discriminative dictionaries of the movements in the scene which implicitly represent the interactions. There is no explicit link between the low-level movement and human body parts. Without such pose information, discriminative patterns of image movements are modeled implicitly, for instance using co-occurring spatiotemporal codewords [15]. To localize an interaction, additional steps need to be taken such as clustering dense trajectories [16, 17]. When a sufficient number of trajectories can be clustered, the volume created by the set roughly encompasses the interaction. This approach is less reliable in the presence of other motions, for example when multiple people interact in close proximity.

Instead of starting from low-level image features, another line of approach is to first detect faces or bodies [5, 18]. Given two close detections, interactions can then be classified based on extracted features within the detection region [19]. Recent work in this area has employed gross body movement and proximity cues for the detection of interactions. Patron-Perez et al. [5] use this two-stage approach to classify human interactions in unsegmented videos. The drawback is that classification is suboptimal when the person localization fails, for example when people partly occlude each other. Yang et al. [20] improve classification in these cases by building detectors for various types of physical interactions such as hand-hand and hand-shoulder touches. The relative distance between individuals has been further explored by Sener and İkizler [21], who formulate interaction detection as a multiple-instance learning problem because not all frames in an interaction are considered informative. Sefidgar et al. [22] use the same reasoning to create a model based on discriminative key frames and consider their relative distance and timing within the interaction. In this paper, we focus on physical interactions and analyze them at the level of body parts.

The availability of body pose and, especially, body movement information has been found to increase action classification performance [8]. Poses and movements of specific body parts characterize interactions [23]. For instance, in a fist bump, the arms extend toward each other and the knuckles of the right hands meet as they touch (see Fig. 1). This requires the lower arm to be in a particular pose and to move forward. The positions of the limbs were first used by Bourdev et al. [24] to detect people engaged in specific actions in still images. Kong et al. [25] combine this with motion, forming units of interaction. They use attributes such as "outstretched hands" and "leaning forward torso" and consider their co-occurrences to characterize interactions.

Some of the attributes might not be informative, such as the positioning of the feet when performing certain greetings. Kong and Fu [26] consider only those body parts that characterize the interaction. Their method pools BoVW responses in a coarse grid. This allows them to identify specific motion patterns relative to a person's location, but the level of detail of the analysis is limited by the granularity of the patches and the accuracy of the person detector. We also focus on physical interactions, but analyze them at a finer scale by modeling the precise pose and movements of specific body parts.

Part-based models such as the deformable parts model (DPM) [10] can be used to detect people in an image and localize their body parts. These models employ part detectors and impose spatial constraints between these parts. DPMs are sufficiently flexible to describe articulations of the body [27]. This enables the detection of key poses representative of an action [28]. Different body parts can occur at different depths within an image due to out-of-plane rotation. The resulting differences in depth make the affected parts more difficult to detect. Allowing parts to scale independently from each other can counter this and has been explored by Dubout and Fleuret [29], who also modeled scale deformations. This results in a significant improvement over using image pyramids.

Yao et al. [30] use DPMs and focus on human-object interactions. To capture the movement related to a key pose, they connect the output of a DPM to a set of motion templates. This formulation works well for the representation of coarse movements, but the motion templates are not connected to specific parts of the DPM model. Tian et al. [31] have extended DPMs for action detection to model changes in pose over time, using spatiotemporal descriptors [32]. Parts are deformable in both space and time but, again, are not connected to specific body parts. As such, they cannot model detailed motion or pose of a specific limb. Van Gemeren et al. [9] use interaction-specific DPMs to locate people in characteristic poses. They then describe the coordinated movement in the region in between DPM detections. As there can be significant variation in how people pose, this two-stage approach strongly relies on the accuracy of the pose detection.

DPMs show promising results in action detection. They are capable of modeling proximity and orientations. For *fine-grained* interactions such as those in social encounters, we need to more effectively model the coordination between the people involved. To this end, we introduce a spatiotemporal DPM that encodes the fine-grained movements and poses of specific limbs from both individuals involved in the interaction. This enables us to detect close proximity interactions with subtle differences in pose and movement.

3 Method

In this section, we introduce our spatiotemporal localization model for fine-grained interaction localization. We also detail the procedures for training such a model on a small set of video examples, and to detect interactions in unsegmented videos.

For human interactions, there is a moment where both the pose and motion are coordinated in a way that is characteristic for the interaction: the *epitome*. At the epitome, we model the two-person interactions with a novel DPM formulation. This formulation is based on Yang and Ramanan [27], who detect articulated poses in images using a flexible mixture-of-parts model. They model the pose as a tree of patches connected with springs. Each patch consists of a set of representations (mixtures) tuned to different angles and rotations. This is a variation of the original DPM [10] in which parts appear on multiple layers, but are not modeled as mixtures. Yang and Ramanan demonstrate accurate detections of limb configurations for a wide variety of human poses. We base our formulation on this model, but adapt four key properties to make it suitable for the localization of coordinated human interactions in videos.

First, Yang and Ramanan model body parts in different orientations using multiple mixtures per part. In our work, we consider articulations potentially characteristic for an interaction and therefore model body parts in specific poses. For example, during a handshake, we model the upper arms, the lower arms, and the hands in the pose that is most characteristic for the given interaction. This representation is reminiscent of the poselets introduced by Bourdev et al. [24], though they only consider the interaction pose, whereas we consider both the pose and its corresponding motion.

Second, not all parts of the body contribute equally to the characterization of an interaction. For some parts, the pose is an important cue, while for others the movement is more important. Therefore, for every part, we model either pose or motion features, or a combination of both. This allows us to focus on those aspects that characterize an interaction.

Third, we model the spatial relation between the body parts and movements of both persons involved in the interaction simultaneously. We also model the orientation and the distance between the two persons during the interaction. Because we focus on fine-grained interactions that are coordinated in the pose and movements of particular limbs, we represent the corresponding parts of both persons in a single tree structure.

Finally, we model the cumulative response space of the part convolutions as a four-dimensional data structure that represents scale, time, and 2D space. Dubout and Fleuret [29] have shown that modeling scale and 2D feature space as a single three-dimensional response space improves accuracy, because individual parts can be scaled independently as opposed to the fixed scales for all parts in the original DPM formulation. We follow this rationale and add time to the response space as a fourth dimension.

With these extensions to [27], we can model fine-grained interactions that differ only slightly. We use this novel spatiotemporal formulation to detect interactions between two individuals from video in scale, time, and space.

3.1 Model formulation

To model the pose and motion at the epitome of each positive training example, we define a graph $G = (V, E)$, with V a set of K body parts and E the set of connections between pairs of parts [27]. The body parts we consider may be compound parts consisting of multiple skeleton joints, such as a *torso*, *right lower arm*, or *left upper leg*. A body part is formed by the smallest area surrounding a given set of skeleton joints, including a margin. For example, the *right upper arm* part is contained in the bounding box surrounding the *right shoulder* and *right elbow* joints, shown as yellow dots in Fig. 2. Each body part i $(1 \le i \le K)$ is centered on location $l_i = (s_i, t_i, y_i, x_i)$ within a spatiotemporal feature pyramid. y_i and x_i represent the 2D location in the feature space of frame t_i, at layer s_i of the feature pyramid, at which the part occurs.

The scoring for a part configuration in image I is given by:

$$S(\boldsymbol{I}, \boldsymbol{l}) = \vartheta_s\Big(\sum_{i \in V} \boldsymbol{w}_i \cdot \phi_i(\boldsymbol{I}, \boldsymbol{l}_i)\Big) + \sum_{ij \in E} \boldsymbol{w}_{ij} \cdot \psi(\boldsymbol{l}_i - \boldsymbol{l}_j) \quad (1)$$

The first term models the part appearance with a convolution of image feature vector $\phi_i(\boldsymbol{I}, \boldsymbol{l}_i)$ with trained detector w_i. After convolution, the result is resized by ϑ_s. For clarity, we omit that the scores are defined by the dot product between a part and a sub-window of a feature pyramid computed from the input image. The second term contains the pairwise deformations between parts $\psi(\boldsymbol{l}_i - \boldsymbol{l}_j) = [ds\ ds^2\ dt\ dt^2\ dy\ dy^2\ dx\ dx^2]$, with $ds = s_i - s_j$, $dt = t_i - t_j$, $dy = r_i\ s_i\ y_i - r_j\ s_i\ y_j$ and $dx = r_i\ s_i\ x_i - r_j\ s_j\ x_j$, the relative location and scale of part i

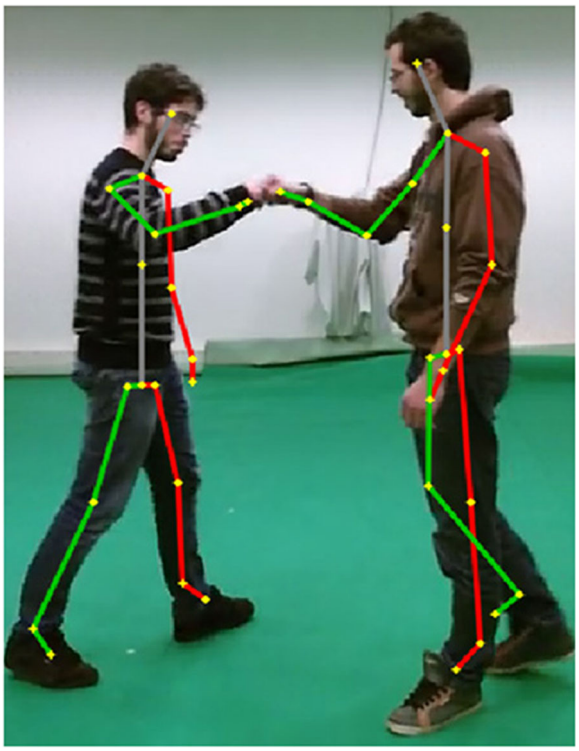

Fig. 2 Interaction with superimposed pose data. Green: right side, red: left side

with respect to part j [27]. Note that the distances dx and dy for part i and j are defined with respect to a root level factor r, scaled by s which is derived from the pyramid layer at which the deformation takes place. We allow each part to have its own spatial resolution. r compensates for the resolution difference between i and j that occurs when parts have a different spatial resolution. In practice, we only allow two different resolutions. This allows us to quickly find candidate detections for some parts at a coarse level, on which a localization of fine-grained pose and motion cues at double the resolution can be detected for other parts. s on the other hand compensates for the scale difference between i and j, that occurs because we scaled responses $\boldsymbol{w_i} \cdot \boldsymbol{\phi}_i(\boldsymbol{I}, \boldsymbol{l}_i)$ by ϑ_s.

We do not allow scaling in this way for ds and dt. ds deals with deformations between responses from different pyramid layers. Allowing parts to move independently in this dimension causes mismatches between the lattices of part locations at different scales. This happens when the size difference between two consecutive layers is non-integral. Dubout and Fleuret solve this by approximating the root position of a part by rounding it to its closest integral position [29]. We take a slightly different approach. Instead of rounding the part location at the root position, we scale $\Sigma_{i \in V}\ \boldsymbol{w_i}\ \psi(\boldsymbol{l}_i - \boldsymbol{l}_j)$ by ϑ_s, which is the factor by which the original input in layer s of the feature pyramid was scaled. As a result, each feature response at a layer of the pyramid becomes a scale space response with the same spatial dimensions. This has the advantage that at each scale the spatiotemporal response space can be concatenated into a single four-dimensional spatiotemporal response matrix. $\boldsymbol{w_{ij}}$ encodes the deformation error of the connection between parts i and j. The deformation function $\psi(\boldsymbol{l}_i - \boldsymbol{l}_j)$ can now be applied on each dimension independently and in linear time with the number of part locations, the same way as it was applied in the original DPM implementation [10]. The chosen structure keeps the model suitable for cascade object detection [33] in which the detection process at a certain image location can be stopped early if a cumulative response score threshold is not met after processing a particular part.

The four key extensions to this model are defined as follows.

3.1.0.1 Class-specific part detectors We learn class-specific detectors that encode the characteristic articulations or movements of the body parts directly. Examples are a right-facing torso or an upwards moving lower arm. Therefore, we use a single detector per class, instead of a mixture of part detectors as in [27]. We base this choice on [24] who train articulated poses and obtain good results in a pose detection task. The main difference with [24] is that we train both pose and motion in this manner. So, in addition to a pose representing the epitome of the interaction, we simultaneously model the fine-grained movements of relevant body parts.

3.1.0.2 Multiple features Our model supports different types of features per part.

For part i with feature representations D_i, we replace the first term in Eq. 1 by:

$$\vartheta_s \left(\sum_{i \in V} \sum_{j \in D_i} b_{ij} \boldsymbol{w}_i^j \cdot \boldsymbol{\phi}_i^j(\boldsymbol{I}, \boldsymbol{l}_i) \right) \tag{2}$$

$\boldsymbol{\phi}_i^j(\boldsymbol{I}, \boldsymbol{l}_i)$ denotes a feature vector of type j for part i. $\boldsymbol{w}_i^j$ is the trained detector for part i and feature type j. Parts can have different combinations of features D_i. In this work D_i is HOG, HOF, or MBH, but it is not limited to these features. In fact, the DPM inference algorithm is well suited to incorporate a learned feature extractor such as convolutional neural networks (**CNN**) [34, 35]. As such, our formulation is different from Yao et al. [30], who require one HOG template and a set of HOF templates per body part. In contrast, our model allows us to focus on those features that are characteristic for a

specific body part and interaction class. We explicitly also consider features that are calculated over time such as HOF descriptors. Bias b_{ij} denotes the weight for each feature type D_i, to emphasize the gradient properties of the pose (HOG) or the movement direction (HOF or MBH) of part $\boldsymbol{w}_i^j$.

3.1.0.3 Two-person interaction As there are two persons involved in a dyadic interaction, we combine their body parts in the same graph (see Fig. 3). Each actor's body parts form a sub-tree in this $(2K+1)$-node graph. The torso parts of both actors are connected through a virtual root part of the graph. This part does not have an associated part detector but it allows us to model relative distances between people, similar to Patron-Perez et al. [5] and Sener and İkizler [21]. We enforce that the size of the virtual root part is equal to the size of the entire dyad of bodies, regardless of the locations and sizes of the associated part detectors. This compensates for the fact that during localization we are interested in a tube that encompasses the entire interaction, as opposed to just a hull surrounding the modeled parts. Finally, a combined graph allows us to stop the detection process at that location early when only one suitable person is found in the dyad.

3.1.0.4 Independent spatiotemporal part modeling Each part can be independently scaled and spatiotemporally transformed, to best fit the interaction that is being modeled. This results in an extremely flexible deformable parts model capable of capturing fine-grained differences between different interaction types.

3.2 Training

For each interaction class, we learn a deformable model from a set of training sequences. We describe a sequence of length n as $\boldsymbol{X} = \{(\boldsymbol{I}_i, y_i, \boldsymbol{p}_i)\}_{i=1}^{n}$ with $\boldsymbol{I}_i$ an image frame, y_i the interaction label of frame i, and $\boldsymbol{p}_i$ a vector containing the 2D joint positions of the two people performing the interaction.

We train the interaction model in three steps. First, we determine the epitome for each training sequence. Second, we learn the initial body part detectors and assemble the body parts into the initial pose and motion model. Third, we simultaneously update the epitome and the body part detectors.

3.2.0.1 Epitome detection At the epitome, the people engaged in an interaction pose in a way that is representative for the interaction. We first find the epitome frame for each positive training sequence. We define the epitome as the pose with the smallest pair-wise differences to other training sequences. We thus find a set of similar frames, one per sequence. To determine these epitome frames, we iterate over the training sequences in a random order. At each iteration, we compare the pose in the current frame, the *seed pose*, to all poses in the other sequences' frames. We assign the frame with smallest Euclidean distance between the seed pose and the other frames' poses as the epitome. This results in a set of epitomes of which we can calculate the cumulative pose difference. We select the set of epitomes with the smallest cumulative pose difference.

We normalize the distances between the joints based on the spine length to compensate for differences in scale. Furthermore, we translate the set of compared joints so they overlap as much as possible with the seed pose. We can efficiently calculate the scaling and the translation with a 2D adaptation of the Kabsch algorithm [36]. For translation, we have a set consisting of k two-dimensional points that represent body joints: $K = ((x_1, y_1), \ldots, (x_k, y_k))$, with mean $(\bar{x}, \bar{y})$. We translate these points such that their mean is at the origin, giving

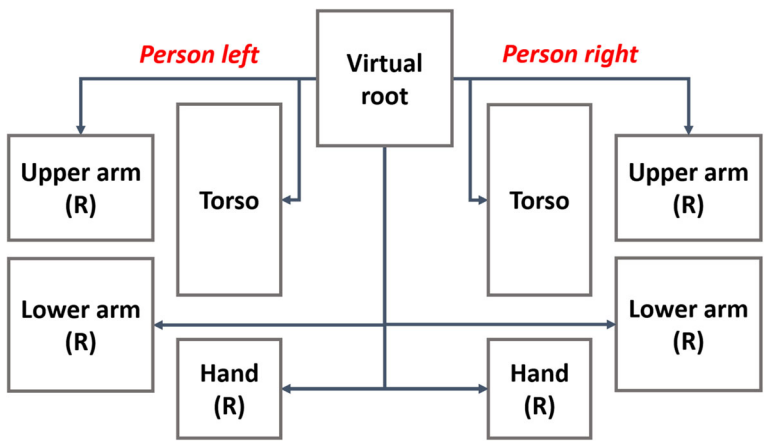

Fig. 3 Graph of the part configurations of our model. Both persons share the same virtual root part which encompasses the entire dyad

points $(x_i - \bar{x},\ y_i - \bar{y})$ $(1 \le i \le k)$. After translation, we perform uniform scaling by s:

$$s = \sqrt{\frac{\Sigma_{i=1}^{k}(x_i - \bar{x})^2 + (y_i - \bar{y})^2}{k}} \tag{3}$$

We do not perform rotation normalization because the orientation of the limb is informative of the interaction.

Based on the summed joint distances with the best seed frame, we find corresponding epitome frames in other sequences of the same interaction. These sequences are labeled as *prime* if the normalized joints distance is below 0.5, and *inferior* otherwise. We thus obtain two sets of sequences with epitome frames: one set where the poses are much alike and one where the differences between the poses are bigger.

3.2.0.2 Initial model learning After finding the epitomes for each positive example of a certain interaction class, we use the prime sequences to train body part detectors $\boldsymbol{w}_i^j$. We specify beforehand, for each part, the type D_i, spatial resolution r and temporal extent τ. Parameter r indicates the cell size. For movement descriptors, τ dictates how many frames around the epitome are used.

For the body parts of both individuals in the interaction, we optimize $\phi_i^j(\boldsymbol{I}, \boldsymbol{l}_i)$ and b_{ij} using dual coordinate descent SVM (DCD SVM) solvers [37]. We create a set of detectors for each part. After this positive optimization round, we perform a round of negative hard data mining [10] for each detector. We harvest negative examples in random frames of the Hannah dataset [38]. This allows us to optimize each detector to maximize its response score for target interactions, while minimizing the scores of patches that do not depict any interaction of interest.

3.2.0.3 Epitome and model refinement We then optimize deformation parameters $\psi(\boldsymbol{l}_i - \boldsymbol{l}_j)$ in Eq. 1, as well as its features $\phi_i^j(\boldsymbol{I}, \boldsymbol{l}_i)$ in Eq. 2, in a latent manner. We apply the model (Eq. 2) on the complete set of positive training examples. This results in an updated positive example set that accounts for more variation, while the negative impact of the less-than-perfect training data is largely avoided.

We assemble the interaction model from the individual part detectors. The placement of the parts is based on the joint locations of the poses at the epitome. Once the initial model is constructed by anchoring all interaction model parts at their relative positions, we apply it to both prime and inferior training sequences of the particular class to detect new latent positive interaction examples. We search for the highest scoring frame in each sequence to add to the positive example set. This frame is the new epitome for that positive example. Given that the initial epitome frames are selected solely based on pose, this step allows us to better represent the motion of the body. The resulting positive example set is used to optimize $\phi_i^j(\boldsymbol{I}, \boldsymbol{l}_i)$, $\psi(\boldsymbol{l}_i - \boldsymbol{l}_j)$ and b_{ij} using the DCD SVM solvers. Examples of trained models are shown in Fig. 4.

3.2.0.4 Platt scaling of the classification output A common problem with the classification scores of SVM solvers is that the output is susceptible to large numbers of false positives. This is because max margin methods, such as SVMs, tend to produce scores distributed around the − 1 to 1 range. We would like to compress the scores of the false positives at the lower end of the SVM classification scores, and those of the true positives at the higher end. To achieve this, we estimate the posterior probabilities as a sigmoid loss function using Platt calibration. We calibrate the sigmoid loss function by testing the model

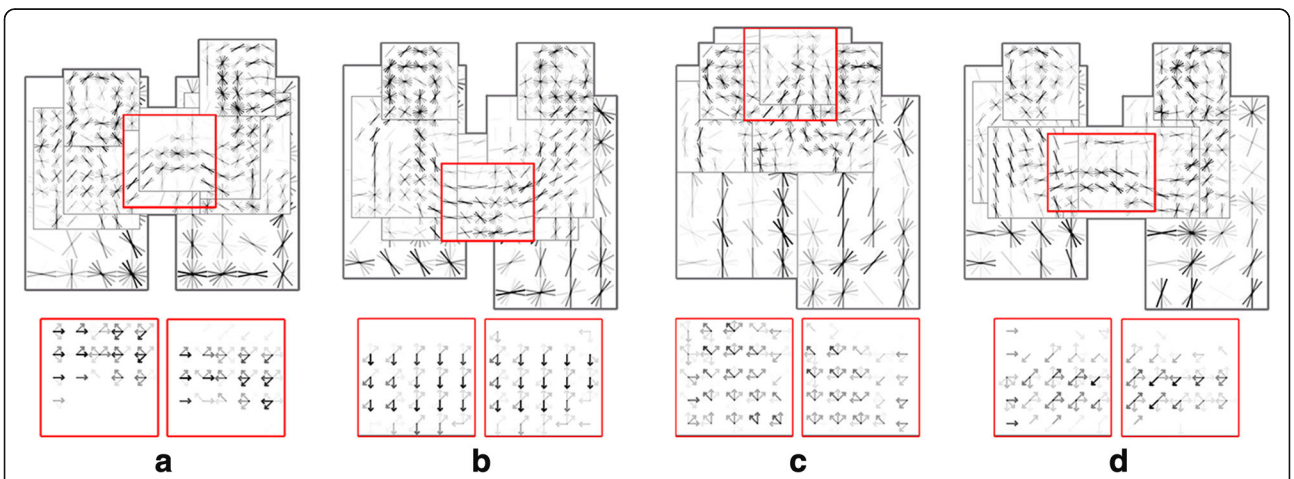

Fig. 4 Trained interaction models. The parts outlined in red represent the gradient features of the right hands. Below the part model, we show the motion features of these parts. **a** Fist bump. **b** Handshake. **c** High five. **d** Pass object

on a set of cross-validation examples taken out of the training set before optimizing the model. The minimal amount we take out is four training examples. The resulting scores are split into positive and negative candidates by determining, for each candidate location in the training example, whether it was a true or false positive. True positives correspond to those locations within the annotated spatiotemporal ground truth. We use the resulting score sets to calibrate the posterior probabilities.

Because we look at the locations of the candidates with respect to the spatiotemporal ground truth of the positive examples, we can keep track of the positive scores of the candidates at the start, epitome, and end of each interaction. Naturally, the scores at the epitome are very close to 1.0 after scaling. The scores at the start and end of each interaction are significantly lower. We use these scores to estimate the duration of the interaction.

3.3 Spatiotemporal localization

A trained model can be used to detect an interaction as a spatiotemporal tube within a video that contains one or more occurrences of the interaction of interest. We first create a set of candidate epitome detections and construct the localization tubes from these candidates.

To speed up the process, we consider only every fourth frame. We first generate a feature pyramid of the types in D to detect interactions at various scales. The number of layers in the pyramid depends on the frame size of the video. In detection, we halve this size every ten layers, until it is smaller than the size on which the model can perform detections. We use dynamic programming to make the feature descriptions of a given type available to all relevant model parts. For each scale, we thus only calculate the feature description and model responses once. The pyramids deal with feature types that take into account multiple frames, such as HOF, by creating descriptions for τ consecutive frames at each layer.

We generate a set of spatiotemporal detection candidates spanning the entire video. We scale each candidate score by applying the sigmoid function obtained from Platt calibration. Finally, we remove spatially overlapping detections at the same temporal location, using non-maximum suppression. Each of the remaining candidates represents a potential epitome for the interaction of interest with a scaled response score $S(\boldsymbol{I}, \boldsymbol{l})$ (Eq. 1).

The next step is to construct tubes from the set of candidate detections. The highest scoring candidates are likely to correspond to the epitome of the interaction. Non-prototypical interaction poses and movements are likely to give a lower response to the interaction localization model. Consequently, we can find candidates that are part of the interaction before and after the epitome at nearby locations in the spatiotemporal response function. We therefore look for subsets of consecutive high-scoring detections that have significant spatial overlap to create the localization tubes (Fig. 5).

Our response function is four-dimensional. Because we apply the deformation function on each dimension, we can find spatiotemporal blobs that represent the spatiotemporal extent of each interaction. During the calibration of the Platt scaling function, we recorded the scores at the edges of the spatiotemporal ground truth, after scaling. We use the mean scores of the start and end of the ground truth tubes to determine when a tube starts and when it ends, with respect to the detected epitome location.

In the response space, we aim to extract tubes that contain the interaction of interest. Temporally, the response space contains gaps because we only test every fourth frame. First, we create a continuous response space by interpolating all the response scores between consecutive frames. Second, we find the best scoring detection and look at the size of the bounding box. We consider this detection as the epitome frame and store the detection's score and size. Instead of relying on the candidate detections, which occur only every fourth frame, we use the continuous response space to find the location of the interaction in each frame by interpolating between detection frames. We then add neighboring frame segments backwards and forwards in time. Iteratively, we project the bounding box to a neighboring frame. Within this area, we center the bounding box on the highest value in the response space. We continue this process as long as the overlap between two subsequent bounding boxes is at least 50% and the score of the newly added segment is higher than the mean tube edge scores obtained during training.

After creating a tube, we remove all candidates that spatially overlap with it for more than 50%. The tube creation process stops when all candidates have either been converted to tubes or have been erased. As a final step, we remove tubes that are shorter than τ frames.

Using the response space instead of directly relying on detection candidates is beneficial for three reasons. First, there can be missing detections, for example as a result of background motion, or partial occlusions. This situation complicates linking detections over time. Using the response space, missing detections can be bridged by interpolation, as neighboring detections ensure that the sub-volume is sufficiently covered. Second, the start and end of a tube does not have to match with the tested frames that have temporal gaps between them. By interpolating between these frames, we can more accurately find the start and end of the interactions. Third, we can detect interactions with varying durations, even those that differ significantly from those seen during training. This is a desirable characteristic, especially for cyclic interactions such as handshakes.

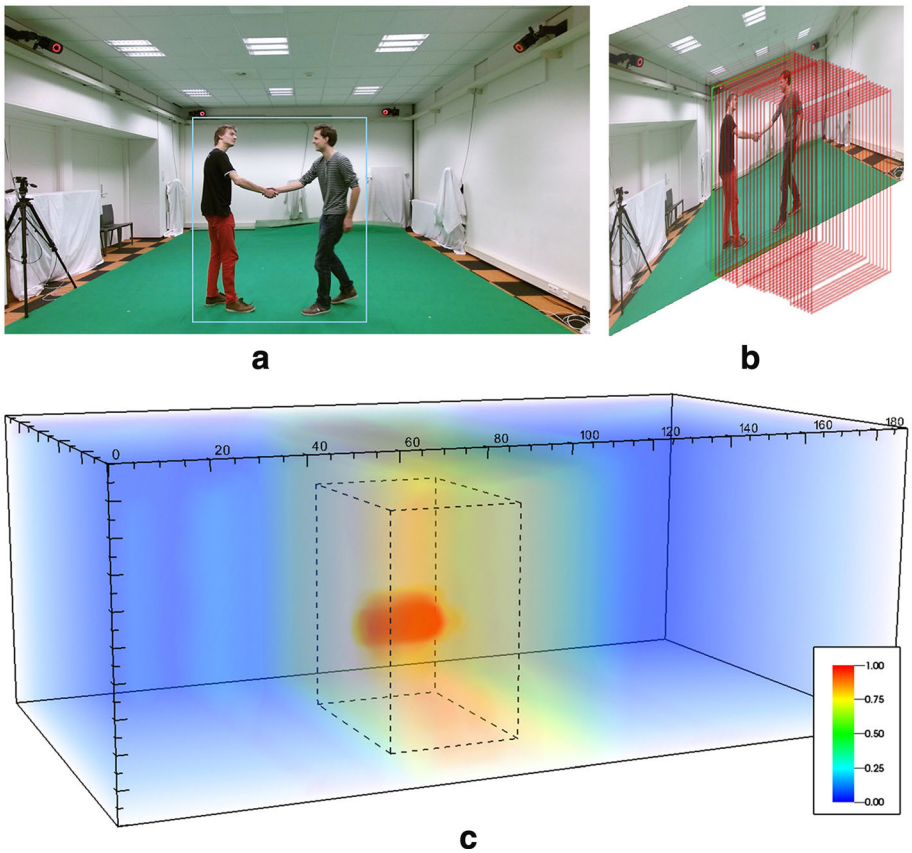

Fig. 5 Example frame of a handshake interaction (**a**). The Platt response space appears in (**c**). The localization tube is shown in (**b**), projected on the epitome frame. **a** Video frame with ground truth box (light blue). **b** Final detection tube. **c** Platt response space with the ground truth bounding volume (dashed)

4 Experiments

Research on interaction *recognition* considers assigning labels to video sequences that are segmented in both space and time. In contrast, we focus on the more challenging task of *spatiotemporal localization* from unsegmented videos. In addition to correct classification, this requires finding the subspace in the video containing interactions of interest.

To experiment with interactions that are visually similar, we use the recently introduced ShakeFive2[1] dataset [39]. We test the generalization of our models using cross-dataset experiments. We train our models on ShakeFive2, and apply them to the unsegmented videos of the interaction datasets UT-Interaction [4] and SBU Kinect [40].

4.1 Datasets

We briefly describe the three publicly available datasets that are used in this paper.

ShakeFive2 consists of 94 videos recorded at 30 frames per second, with a resolution of 1280×720 pixels. The videos feature five close proximity interactions: *fist bump*, *handshake*, *high five*, *hug*, and *pass object*. Each video contains one two-person interaction, recorded under controlled settings but with small variations in viewpoint (see Fig. 6). For each person in each frame, 2D joint position data obtained using Kinect2 is available. Interactions are labeled per frame.

UT-Interaction consists of two sets of 10 videos each, recorded at 30 frames per second with a resolution of 720×480 pixels. The first set features at most two interacting persons at each moment, while the second set contains multiple pairs of people interacting simultaneously. The following interactions are performed: *handshake*, *hug*, *kick*, *point*, *punch*, and *push*. No pose data is available and we use the bounding box data from [21] as ground truth.

SBU Kinect involves two actors performing one interaction per video in an indoor setting. The videos were recorded using a Kinect at 15 frames per second and a resolution of 640×480 pixels. The interactions are *hand shake*, *high five*, *hug*, *pass object*, *kick*, *leave*, *punch*, and *push*. Pose data, obtained with a Kinect, is provided but is not always accurate. From the 260 videos, we exclude 42 with incorrect pose data.

Fig. 6 Example frames from ShakeFive2, SBU Kinect, and UT-Interaction. Top row: handshake, bottom row: hug

While both ShakeFive2 and UT-Interaction are recorded at 30 fps, SBU Kinect has a frame rate of 15 fps. During localization, we can modify the temporal extent of the model by dividing τ by two. Combined with skipping half as many frames while processing a video, we can compensate for this difference in frame speed.

4.2 Performance measurements

We detect interactions in both space and time and use the average intersection over union of the ground truth G and detected tube P as in [17]. G and P are two sets of bounding boxes and θ is the set of frames in which either P or G is not empty. The overlap, expressed as the intersection over union (IoU), is calculated as follows:

$$\mathrm{IoU}(G,P) = \frac{1}{\|\theta\|}\sum_{f\in\theta}\frac{G_f \cap P_f}{G_f \cup P_f} \tag{4}$$

We evaluate different overlap thresholds σ for which IoU $(G, P) \geq \sigma$ and report the mean area under the curve (AuC), which is a measure for the average precision over all interactions in each fold and each interaction. In the evaluation, we compare against a ground truth which has bounding boxes of the full body. Therefore, we compare its spatial size to the size of the virtual root part, which was scaled to represent the spatial size of the interaction dyad during training. We use the relative location and scale of this part to account for the increased height of the ground truth if only part of the body is modeled.

To analyze confusions between pairs of classes, we use a difference mean average precision (d-mAP) multi-class confusion matrix [39]. Each score in this matrix indicates how much of the AuC for a given class is lost to another class.

4.3 Experimental setup

To test the performance of a model on a given video, we process every eighth frame. Our models have a temporal extent of $\tau = 15$ frames, which means there is a temporal overlap between candidates of seven frames. In an unsegmented video, we find all candidate detections as described in Section 3.3.

We consider two testing scenarios: *single class* (SC) and *multi-class* (MC). For single class localization, we apply a detector for a given interaction to test videos for that class only. This scenario measures the spatiotemporal localization accuracy. In the multi-class scenario, we use the detector on all available test sequences in a dataset. This allows us to test for confusions with other interactions. In this scenario, we measure the response from the model on interactions of the target class and distractor classes. False positives occur when responses from a distractor class score higher than the responses from the target class. This common situation will lead to a lower precision as we do not compare or filter these detections.

4.3.1 Baseline comparison

We compare our method to a baseline using a codebook consisting of dense trajectories [13] and Fisher vectors (FV). This approach has achieved state-of-the-art performance in the localization of individual human actions [17]. We largely follow [14]. During training, we obtain a set of dense trajectories over all relevant parts in all videos of a given interaction class. The vector length of each feature type (HOG, HOF, and MBH) in the dense trajectories is halved using principle component analysis. We encode all trajectories into a Gaussian mixture model (GMM) with $K = 256$ clusters and a codebook size of 256 k trajectory keywords. We use FV in a straightforward manner. The length of each Fisher vector is determined by $F = 2KD$, with D the number of dimensions of the concatenated feature descriptors from the dense trajectories. With the resulting codebook, we train a linear support vector machine (lSVM).

At test time, we encode a video with dense trajectories. We then create sets of trajectories that represent subsections of the video. We use a sliding window on every eighth

frame to determine the subsections. A dense trajectory is selected when it intersects with a subsection. We filter out sets that are empty or that contain the same trajectories. We use the codebook to encode subsections and classify them with the lSVM.

5 Results and discussion

We first evaluate the performance of our method on ShakeFive2 with different model configurations. Then we look at parameter settings, training strategies, and the amount of training data. Finally, we evaluate the performance of our models in two cross-dataset experiments.

5.1 Model feature configurations

We test the HOG, HOF, HOGMBH, HOGHOF, and HOGHOFMBH models on the ShakeFive2 dataset. We compare against the baseline of dense trajectories with Fisher vectors (FV). We refer to the five interactions as FB (fist bump), HS (handshake), HF (high five), HU (hug), and PO (pass object). We use a minimal overlap between the detected tube and the ground truth volume (Eq. 4) of 30% ($\sigma = 0.3$). The AuC is obtained using the mean of a four-fold cross-validation. In Fig. 7, we show IoU (*G*, *P)*-diagrams for varying σ (Eq. 4) for the four different models. Results for the SC and MC scenarios are shown in Table 1.

In the baseline FV experiment, we have purposely omitted using a feature pyramid. We select the number of dense trajectories suitable for the scale at which the interactions take place. This eliminates false detections at different scales. Even with this advantage, FV does not perform well in both the SC and MC scenarios. We believe this is mainly due to the limited amount of data that the codebook and the SVM are trained on. The fine-grained differences between the different interactions are not captured well by the dense trajectories. Given the superior performance on human action detection [17], the lower scores for the detection of fine-grained interactions suggest that not explicitly modeling the coordination between people is disadvantageous.

When tested only on videos of the same class, the HOGHOFMBH model outperforms all other model configurations. This demonstrates that interactions are more accurately detected by a combination of pose and motion information. When additional sequences of other interactions are tested (MC), we notice a drop for all models but only marginally for HOGHOFMBH. Especially when relying only on pose information alone in the HOG model, the confusion between interactions increases. The HOGMBH model does not really improve performance over the HOG model. And although the HOGHOF model improves performance slightly more, the most dramatic improvement is caused by the combination of all three feature types.

There are differences in performance between interactions. Handshakes are detected robustly by all our models. Hugs are detected significantly worse than other interactions. Especially when the model does not employ all feature descriptors, the performance is very bad. This can be attributed to the inaccurate pose information in the training data as a result of frequent occlusions. Despite modeling pose and motion, there are still significant visual similarities between, for instance, a hug and a distractor class. Because the subjects approach each other, their gross body movement is also similar. As the limb placement is not very well captured by the hug model, we can understand why a false positive like the right image in Fig. 8 occurs. Additionally, training the parts as poselets for each individual will prove very difficult even with sufficient amounts of properly annotated data. Because of the minimal distance between the subjects during the hug, there is little coordination between them, which makes it hard to gather useful pose features during training.

5.2 Confusions

In the multi-class setting, we investigate how often interactions are confused. Table 2 presents the d-mAP scores on ShakeFive2 for HOG and HOGHOFMBH. For the HOG model, there are many confusions. Pose information alone is not sufficient to distinguish between interactions

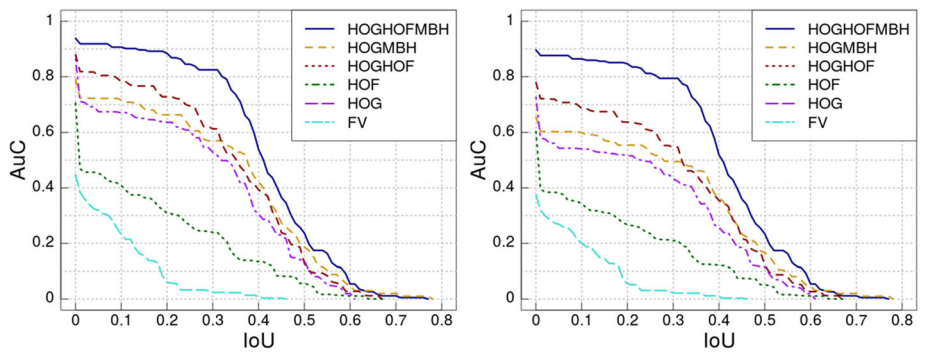

Fig. 7 Mean AuC over all interactions using four-fold cross-validation in the SC (left) and MC (right) scenarios of ShakeFive2

Table 1 Mean AuC using four-fold cross-validation on ShakeFive2 with $\sigma = 0.3$

	SC/MC	FB	HS	HF	HU	PO	Avg.
FV	SC	0.02	0.04	0.02	0.03	0.02	0.03
HOG	SC	0.53	0.66	0.76	0.01	0.70	0.53
HOF	SC	0.18	0.64	0.30	0.05	0.03	0.24
HOGHOF	SC	0.74	0.92	0.83	0.10	0.47	0.61
HOGMBH	SC	0.70	0.65	0.83	0.05	0.63	0.57
HOGHOFMBH	SC	1.00	0.95	0.95	0.43	0.79	0.82
FV	MC	0.01	0.03	0.02	0.03	0.01	0.02
HOG	MC	0.43	0.51	0.76	0.01	0.49	0.44
HOF	MC	0.15	0.62	0.21	0.05	0.02	0.21
HOGHOF	MC	0.64	0.80	0.60	0.10	0.36	0.50
HOGMBH	MC	0.58	0.54	0.83	0.04	0.49	0.50
HOGHOFMBH	MC	0.96	0.93	0.94	0.43	0.70	0.79

that differ slightly in temporal coordination: handshake, fist bump, and pass object. The number of confusions for the HOGHOFMBH model is much lower. The additional motion information can be used to avoid misclassification between visually similar interactions.

5.3 Amount of training data

The HOGHOFMBH models achieve good localization performance despite being trained on a small number of example sequences. Here, we test the performance of the model when trained on different numbers of sequences. In different cross-validation configurations, we train on examples from a single fold and test on all other folds. By switching training and test folds, we can test our models on all examples in a single fold, after training them on all other folds. We evaluate both assignment configurations for four folds, and we evaluate two folds. Our scheme results in 15, 10, and 5 training examples and 20, 50, and 75 test examples, respectively.

Table 2 d-mAP scores for the HOG (left) and HOGHOFMBH (right) models on ShakeFive2. In columns the true class, in rows the tested class

	FB	HS	HF	HU	PO		FB	HS	HF	HU	PO
FB		0.26	0.17	0.11	0.17	FB		0.02	0.01	0.00	0.02
HS	0.11		0.02	0.03	0.17	HS	0.01		0.00	0.01	0.04
HF	0.14	0.13		0.13	0.13	HF	0.00	0.01		0.02	0.00
HU	0.52	0.52	0.52		0.51	HU	0.18	0.19	0.19		0.18
PO	0.17	0.24	0.02	0.01		PO	0.09	0.14	0.05	0.05	

Figure 9 shows the AuC. Performance goes up with the number of training examples. The difference between 15 and 10 sequences is very small and suggests that saturation occurs at a low number of training sequences. This is advantageous as obtaining training sequences with pose data might be difficult, especially when many interactions are considered.

5.4 Cross-dataset evaluation

In this section, we investigate how well our models generalize by testing on different datasets. We train models using all available examples from ShakeFive2.

5.4.1 UT-Interaction

We evaluate the HOGHOFMBH models on the UT-Interaction dataset. When we apply our model to an environment not seen before, the lSVM classification scores are significantly lower. This results in a negative shift on the Platt scale, which causes both positive and negative detections to fall on the lower end of the sigmoid function, resulting in bad performance. We solve this by rescaling the response scores with the Platt algorithm using a leave-one-out cross-validation on the UT-Interaction data set. We report the mean AuC with a minimal overlap between the detected tube and the ground truth volume (Eq. 4) of 10% ($\sigma = 0.1$), because the cut-off point is lower for these results.

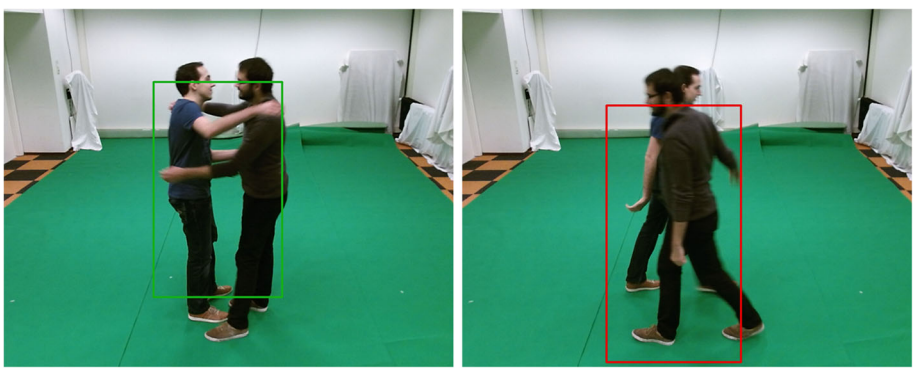

Fig. 8 True positive hug localization (score 0.94) (left). False positive localization of a hug with two persons about to cross each other after a distractor interaction (score 0.36) (right)

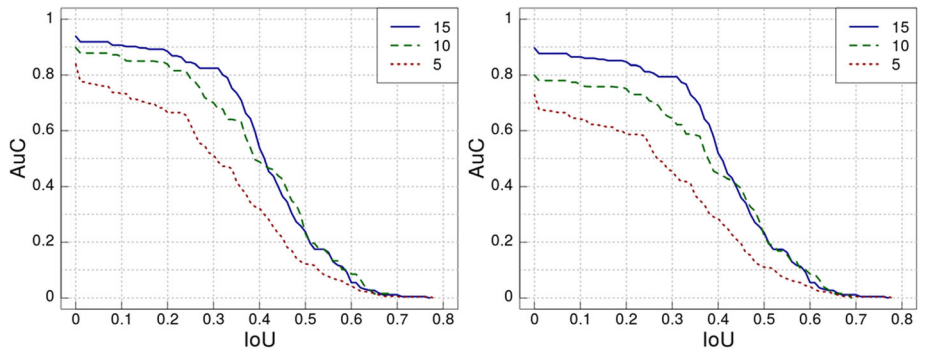

Fig. 9 Mean AuC over all interactions using four-fold cross-validation with different numbers of training sequences, for SC (left) and MC (right)

Ours, and previously reported results are summarized in Table 3. A direct comparison with other works is difficult for a number of reasons. First, we report localization results only for handshake and hug, the two interactions shared between ShakeFive2 and UT-Interaction. Second, we report spatiotemporal localization results, whereas other works consider a recognition scenario. In the latter setting, volumes segmented in space and time are classified. Third, we train our models on a different dataset.

Our cross-dataset results show promising scores for both the SC and MC evaluations. Not only is our metric strict on the measurement of spatiotemporal overlap with the ground truth, it also takes into account whether or not the tube label is of the correct interaction class. From our results, we can conclude that we are capable of localizing interactions of interest with reasonable accuracy, although we do obtain a moderate number of false positive localizations for visually similar interactions.

We note higher scores for hug than for handshake. This is because most interactions occur with similar interpersonal distance, except for hug. Approaching each other to perform a hug is visually similar to a handshake, as can be seen in Fig. 10. Therefore, it is easier to confuse a hug or a push for a handshake, than the other way around.

The d-mAP scores on the UT-Interaction data in Table 4 show that the hug interaction is rarely confused with other interactions. The pose of the bodies with respect to each other seems to be sufficiently different compared to the other interactions. There are more confusions in the classes where the two persons have an extended arm, such as handshake and push.

Our method has more difficulties with set two for the handshake interaction, which contains more occlusions when people walk behind each other. These cause occasional confusions between interaction classes. We note that for the hug interaction, we obtain higher scores with set two than with set one. We believe this is caused by the camera angle. Set two is filmed with a vertical orientation that is comparable to the perspective in ShakeFive2, while set one is filmed from a higher perspective (see rightmost top and bottom images Fig. 6).

Table 3 Single-class (SC) and multi-class (MC) AuC for UT-Interaction (left). Classification accuracies reported on UT-Interaction (right). Reported values are for $\sigma = 0.1$

	Set	HS	HU	Avg.	Method	Avg. (%)
SC	#1	0.64	0.65	0.70	Raptis and Sigal [28]	100
	#2	0.59	0.93		Ryoo [18]	85
MC	#1	0.54	0.52	0.61	Sener and İkizler [21]	100
	#2	0.46	0.93		Zhang et al. [15]	100

5.4.2 SBU Kinect

Table 5 summarizes the performance on SBU Kinect [40]. We have tested the "noisy" variation of this dataset using the HOGHOFMBH model after recalibrating the Platt scaling using four-fold cross-validation.

We obtain near perfect scores for handshake in the SC and MC scenario even though we did not train on this dataset. For higher values of σ, our method remains close to perfect up to $\sigma = 0.3$. This is partly because the videos of SBU Kinect are cropped right to the moment the interaction takes place. As there is no start or end to the interactions, there are fewer false positives due to overestimations in the temporal domain.

We compare our results on spatiotemporal localization to reported classification scores. Yun et al. [40] use pose features and obtain 75, 61, and 85% recognition accuracy for the handshake, hug, and pass object interactions, respectively. If we interpret our MC results as a measure for classification accuracy, our performance is slightly better for handshake but lower for hug and pass object. Still, our method does not require prior segmentation in time and space.

The d-mAP scores for SBU Kinect in Table 6 reveal that handshake (HS) hardly has any confusions. Hug (HU) has a moderate number of confusions, mainly with pass object (PO) and punch (PC). Pass object (PO) has a high number of confusions with all other classes. On closer inspection of

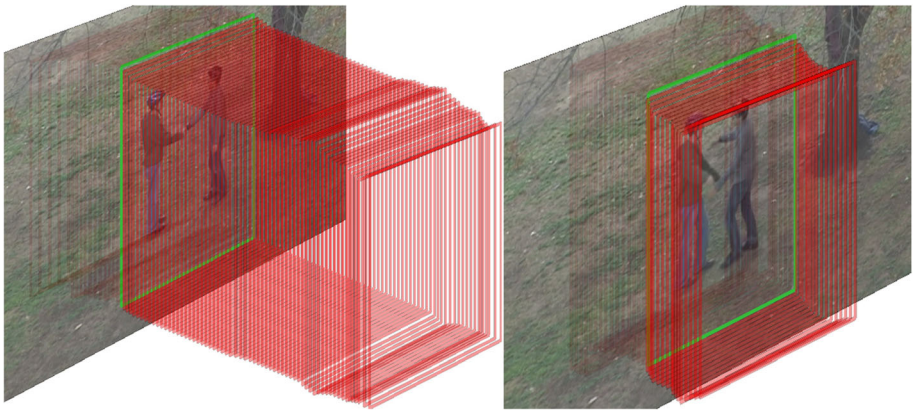

Fig. 10 True positive handshake localization (score 0.65, spatiotemporal overlap with ground truth 42%) (left). False positive localization of a handshake when two persons are just about to hug (score 0.67) (right)

the videos of both SBU Kinect and ShakeFive2, we notice that there is a significant difference in the type of object that is passed on. This introduces a bias during training that eventually reduces generalization. It seems this can be solved by using more varied training data.

6 Conclusions

We have introduced a novel spatiotemporal deformable part model for the localization of two-person interactions. We combine pose and motion by means of HOG, HOF, and MBH to represent the fine-grained body part coordination of two persons. Our models localize the epitome of the interaction, which we expand temporally to form tubes that cover the duration of the interaction. With this approach, we are the first to address spatiotemporal localization of interactions. We cannot only say whether an interaction has occurred, but also recover its spatial and temporal extent.

We train an interaction model from only a few videos with pose information. We find that models that combine HOG, HOF, and MBH features perform best on the ShakeFive2 dataset, which contains interactions that vary subtly. We achieve AuC scores of 0.82 with an IoU of 0.3. In the presence of visually similar interactions, a combination of pose and motion information reduces the number of misclassifications. The generalization of our approach to different settings is demonstrated in cross-dataset experiments on the UT-Interaction and SBU Kinect datasets.

Our method is appealing for several reasons. First, we can perform spatiotemporal localization on unsegmented videos. Second, we require only a modest number of training examples (10–15) to learn robust models. Third, our model formulation is flexible enough to incorporate different features and part configurations, so other interaction classes can be easily trained.

Despite its good performance, the method has some limitations. Most importantly, the temporal extent of the interaction is difficult to estimate with our method because we train our models on the epitome of an interaction, which covers only a small part of it. We rely on the degradation of the detection score with respect to the epitome to estimate the duration of the interaction. This is not the optimal way of modeling an interaction's start and end point. It may be helpful to explicitly model these poses in a mixture model that represents the onset and ending, as well as the epitome as separate model components.

Second, we rely on pose information during training. When this information is inaccurate, the trained model is suboptimal and produces more false positives. Eliminating the dependency of pose data could help to avoid this issue, while at the same time making our approach applicable to a variety of datasets. Another source of false positives is commonly occurring detections from multiple overlapping interaction models. We could eliminate duplicate detections by comparing the detections of the interaction models.

Table 4 d-mAP scores for the HOGHOFMBH models on UT-Interaction. In columns the true class, in rows the tested class. Addition interactions from UT-Interaction include kick (KI), point (PT), punch (PC), and push (PS)

	HS	HU	KI	PT	PC	PS
HS		0.21	0.22	0.17	0.13	0.16
HU	0.08		0.08	0.10	0.07	0.11

Table 5 Single class (SC) and multi-class (MC) AuC for SBU Kinect at $\sigma = 0.1$

	SC/MC	HS	HU	PO
HOGHOFMBH	SC	0.99	0.63	0.21
HOGHOFMBH	MC	0.87	0.20	0.06

Table 6 d-mAP scores for the HOGHOF models on SBU Kinect. In columns the true class, in rows the tested class. Addition interactions from SBU Kinect include kick (KI), leave (LV), punch (PC), and push (PS)

	HS	HU	KI	LV	PC	PS	PO
HS		0.06	0.03	0.00	0.01	0.05	0.05
HU	0.38		0.47	0.22	0.53	0.38	0.58
PO	0.76	0.83	0.70	0.81	0.79	0.83	

Third, our method uses hand-crafted features, such as HOG, HOF, and MBH. These features may produce sub-optimal region proposals for the body parts of the model. We can improve them by using techniques that do not rely on hand crafted features, such as Faster R-CNN [35].

Finally, our models only handle a single perspective. We would like to include multiple perspectives into our models to improve viewpoint independence.

Together, we envision that these improvements bring closer the automated spatiotemporal localization of a broad range of social interactions in unconstrained videos. This will allow for the automated analysis of TV footage and web videos. Moreover, we aim at the application of our work in dedicated social surveillance settings such as in public spaces and elderly homes.

7 Endnotes

[1]ShakeFive2 is publicly available from http://www.projects.science.uu.nl/shakefive

Abbreviations

AuC: Area under the curve; BoVW: Bag of visual words; CNN: Convolutional neural network; DCD: Dual coordinate descent; d-mAP: Difference mean average precision; DPM: Deformable parts model; FB: Fist bump; FV: Fisher vectors; GMM: Gaussian mixture model; HF: High five; HOF: Histogram of optical flow; HOG: Histogram of oriented gradients; HS: Handshake; HU: Hug; IoU: Intersection over union; KI: Kick; lSVM: Linear support vector machine; LV: Leave; mAP: Mean average precision; MBH: Motion boundary histogram; MC: Multi class; PC: Punch; PO: Pass object; PS: Push; PT: Point; SC: Single class; SVM: Support vector machine; VLAD: Vector of locally aggregated descriptors

Acknowledgements

Not applicable.

Funding

This project was funded by public-private ICT research community COMMIT and NWO TOP grant ARBITER.

Availability of data and materials

The datasets supporting the conclusions of this article are available from the following online repositories:

1. ShakeFive2 Dataset. A collection of Human interactions with accompanying skeleton metadata. http://www.projects.science.uu.nl/shakefive (Accessed December 15, 2017)
2. UT-Interaction Dataset. ICPR contest on Semantic Description of Human Activities (SHDA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html (Accessed December 15, 2017)
3. SBU Kinect Dataset. Two-person Interaction Detection Using Body-Pose Features and Multiple Instance Learning. http://www3.cs.stonybrook.edu/~kyun/research/kinect_interaction/ (Accessed December 15, 2017)

About the authors

Coert van Gemeren received a M.Sc. in Cognitive Artificial Intelligence from Utrecht University, the Netherlands. In 2012, he started his PhD on the topic of human interaction recognition in video. His research interests include computer vision, object detection, computational modeling, and the application of artificial intelligence in real-life settings.
Ronald Poppe received a PhD in Computer Science from the University of Twente, the Netherlands. In 2009, 2010, and 2012, he was a visiting researcher at the Delft University of Technology, Stanford University and University of Lancaster, respectively. He is currently an assistant professor at the Information and Computing Sciences department of Utrecht University. His research interests include the analysis of human behavior from videos and other sensors, the understanding and modeling of human (communicative) behavior, and the applications of both in real-life settings. In 2012 and 2013, he received the most cited paper award from the "Image and Vision Computing" journal, published by Elsevier.
Remco C. Veltkamp is full professor of Multimedia at Utrecht University, the Netherlands. His research interests are the analysis, recognition and retrieval of, and interaction with, music, images, and 3D objects and scenes, in particular the algorithm and experimentation aspects. He has written over 150 refereed papers in reviewed journals and conferences, and supervised 15 PhD theses. He was director of the national project GATE—Game Research for Training and Entertainment and is now director of the Utrecht Center for Game Research, www.gameresearch.nl.

Authors' contributions

CvG and RP set up the research. CvG performed the experiments. CvG, RP, and RV contributed to writing. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 26 May 2017 Accepted: 14 February 2018
Published online: 01 March 2018

References

1. C Schuldt, I Laptev, B Caputo, *Recognizing Human Actions: A Local SVM Approach*, Proceedings International Conference on Pattern Recognition (ICPR) (2004), pp. 32–36
2. JK Aggarwal, MS Ryoo, Human activity analysis: a review. ACM Computing Surveys (ACM) **43**(3), 16 (2011) 1–16:43
3. M Marszałek, I Laptev, C Schmid, *Actions in Context*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2009), pp. 2929–2936
4. M. S. Ryoo, J. K. Aggarwal, UT-Interaction Dataset, ICPR Contest on Semantic Description of Human Activities (SDHA), 2010. http://cvrc.ece.utexas.edu/SDHA2010. Accessed 13 Dec 2017
5. A Patron-Perez, M Marszałek, I Reid, A Zisserman, Structured learning of human interactions in TV shows. IEEE Trans. Pattern. Anal. Mach. Intell **34**(12), 2441–2453 (2012)
6. W Choi, S Savarese, Understanding collective activities of people from videos. IEEE Trans. Pattern. Anal. Mach. Intell **36**(6), 1242–1257 (2014)
7. R Poppe, A survey on vision-based human action recognition. Image Vis. Comput. **28**(6), 976–990 (2010)
8. H Jhuang, J Gall, S Zuffi, C Schmid, MJ Black, *Towards Understanding Action Recognition*, Proceedings IEEE International Conference on Computer Vision (ICCV) (2013), pp. 3192–3199
9. C van Gemeren, RT Tan, R Poppe, RC Veltkamp, *Dyadic Interaction Detection from Pose and Flow*, Proceedings Human Behavior Understanding Workshop (ECCV-HBU) (2014), pp. 101–115
10. PF Felzenszwalb, RB Girshick, DA McAllester, D Ramanan, Object detection with discriminatively trained part-based models. IEEE Trans. Pattern. Anal. Mach. Intell **32**(9), 1627–1645 (2010)

11. T Lan, Y Wang, W Yang, SN Robinovitch, G Mori, Discriminative latent models for recognizing contextual group activities. IEEE Trans. Pattern. Anal. Mach. Intell **34**(8), 1549–1562 (2012)
12. MJ Marín-Jiménez, E Yeguas, N Pérez de la Blanca, Exploring STIP-based models for recognizing human interactions in TV videos. Pattern Recogn. Lett. **34**(15), 1819–1828 (2013)
13. H Wang, A Kläser, C Schmid, L Cheng-Lin, Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vis **103**(1), 60–79 (2013)
14. J Sanchez, F Perronnin, T Mensink, J Verbeek, Image classification with the Fisher vector: Theory and practice. Int. J. Comput. Vis **105**(3), 222–245 (2013)
15. Y Zhang, X Liu, M-C Chang, W Ge, T Chen, *Spatio-Temporal Phrases for Activity Recognition*, Proceedings European Conference on Computer Vision (ECCV) (2012), pp. 707–721
16. B Ni, P Moulin, X Yang, S Yan, *Motion Part Regularization: Improving Action Recognition Via Trajectory Selection*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2015), pp. 3698–3706
17. JC van Gemert, M Jain, E Gati, CGM Snoek, *APT: Action Localization Proposals from Dense Trajectories*, Proceedings British Machine Vision Conference (BMVC) (2015), p. A117
18. MS Ryoo, JK Aggarwal, Stochastic representation and recognition of high-level group activities. Int. J. Comput. Vis **93**(2), 183–200 (2011)
19. MS Ryoo, *Human Activity Prediction: Early Recognition of Ongoing Activities from Streaming Videos*, Proceedings IEEE International Conference on Computer Vision (ICCV) (2011), pp. 1036–1043
20. Y Yang, S Baker, A Kannan, D Ramanan, *Recognizing Proxemics in Personal Photos*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2012), pp. 3522–3529
21. F Sener, N İkizler-Cinbis, Two-person interaction recognition via spatial multiple instance embedding. J. Vis. Commun. Image Represent. **32**(C), 63–73 (2015)
22. YS Sefidgar, A Vahdat, S Se, G Mori, Discriminative key-component models for interaction detection and recognition. Comp. Vision Image Underst. (CVIU) **135**, 16–30 (2015)
23. J Wang, Z Liu, Y Wu, J Yuan, *Mining Actionlet Ensemble for Action Recognition with Depth Cameras*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2012), pp. 1290–1297
24. L Bourdev, S Maji, T Brox, J Malik, *Detecting People Using Mutually Consistent Poselet Activations*, Proceedings European Conference on Computer Vision (ECCV) - Part V (2010), pp. 168–181
25. Y Kong, Y Jia, Y Fu, Interactive phrases: semantic descriptions for human interaction recognition. IEEE Trans. Pattern. Anal. Mach. Intell **36**(9), 1775–1788 (2014)
26. Y Kong, Y Fu, Close human interaction recognition using patch-aware models. IEEE Trans. Image. Process **25**(1), 167–178 (2015)
27. Y Yang, D Ramanan, Articulated human detection with flexible mixtures of parts. IEEE Trans. Pattern. Anal. Mach. Intell **35**(12), 2878–2890 (2013)
28. M Raptis, L Sigal, *Poselet Key-Framing: A Model for Human Activity Recognition*, Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013), pp. 2650–2657
29. C Dubout, F Fleuret, *Deformable Part Models with Individual Part Scaling*, Proceedings British Machine Vision Conference (BMVC) (2013), p. 28.1
30. BZ Yao, BX Nie, Z Liu, S-C Zhu, Animated pose templates for modeling and detecting human actions. IEEE Trans. Pattern. Anal. Mach. Intell **36**(3), 436–452 (2014)
31. Y Tian, R Sukthankar, M Shah, *Spatiotemporal Deformable Part Models for Action Detection*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2013), pp. 2642–2649
32. A Kläser, M Marszałek, C Schmid, *A Spatio-Temporal Descriptor Based on 3D–Gradients*, Proceedings British Machine Vision Conference (BMVC) (2008), pp. 995–1004
33. PF Felzenszwalb, RB Girshick, DA McAllester, *Cascade Object Detection with Deformable Part Models*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2010), pp. 2241–2248
34. R Girshick, F Iandola, T Darrell, J Malik, *Deformable Part Models Are Convolutional Neural Networks*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2015), pp. 437–446
35. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Advances in Neural Information Processing Systems (NIPS), 2015.
36. W Kabsch, A discussion of the solution for the best rotation to relate two sets of vectors. Acta, Crystallogr, A **34**(5), 827–828 (1978)
37. JS Supancic III, D Ramanan, *Self-Paced Learning for Long-Term Tracking*, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR) (2013), pp. 2379–2386
38. A Ozerov, J Vigouroux, L Chevallier, P Pérez, *On Evaluating Face Tracks in Movies*, Proceedings International Conference on Image Processing (ICIP) (2013), pp. 3003–3007
39. C van Gemeren, R Poppe, RC Veltkamp, *Spatio-Temporal Detection of Fine-Grained Dyadic Human Interactions*, Proceedings Human Behavior Understanding Workshop (ECCV-HBU) (2016), pp. 116–133 http://www.projects.science.uu.nl/shakefive. Accessed 13 Dec 2017
40. K Yun, J Honorio, D Chattopadhyay, TL Berg, D Samaras, *Two Person Interaction Detection Using Body-Pose Features and Multiple Instance Learning*, Proceedings Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2012), pp. 28–35