Finding Group Interactions in Social Clutter

Finding Group Interactions in Social Clutter

Ruonan Li
Parker Porfilio
and
Todd Zickler

TR-01-13

Computer Science Group
Harvard University
Cambridge, Massachusetts

Finding Group Interactions in Social Clutter

Ruonan Li
Harvard University
ruonanli@seas.harvard.edu

Parker Porfilio
Brown University
parker_porfilio@brown.edu

Todd Zickler
Harvard University
zickler@seas.harvard.edu

Abstract

We consider the problem of finding distinctive social interactions involving groups of agents embedded in larger social gatherings. Given a pre-defined gallery of short exemplar interaction videos, and a long input video of a large gathering (with approximately-tracked agents), we identify within the gathering small sub-groups of agents exhibiting social interactions that resemble those in the exemplars. The participants of each detected group interaction are localized in space; the extent of their interaction is localized in time; and when the gallery of exemplars is annotated with group-interaction categories, each detected interaction is classified into one of the pre-defined categories. Our approach represents group behaviors by dichotomous collections of descriptors for (a) individual actions, and (b) pair-wise interactions; and it includes efficient algorithms for optimally distinguishing participants from by-standers in every temporal unit and for temporally localizing the extent of the group interaction. Most importantly, the method is generic and can be applied whenever numerous interacting agents can be approximately tracked over time. We evaluate the approach using three different video collections, two that involve humans and one that involves mice.

1. Introduction

Social interactions are common, but they rarely take place in isolation. Conversations and other group interactions occur on busy streets, in crowded cafes, in conference halls, and in other types of social gatherings. In these situations, before a computer vision system can *recognize* distinctive group interactions, it must first *detect* them by distinguishing between participants and by-standers and by localizing them in time. This paper addresses this spatio-temporal detection problem for cases in which the agents in a large gathering can be reasonably detected and tracked.

We consider group interactions broadly as distinctive space-time structural co-occurrence of individual actions. These occur in a variety of places and over a variety of times scales. We might want to find in a cocktail party, for example, all three-person conversations dominated by one person for a sustained period of time. On a busy street, we could search for all cases in which two passersby exchange a

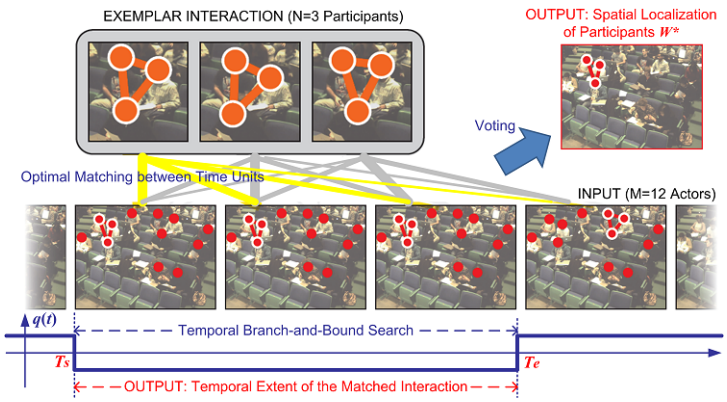

Figure 1. Detecting and localizing interactions in social clutter. Given an exemplar video of an N-person social interaction, we seek to find similar interactions in a long input video with $M > N$ approximately-tracked people. For each temporal frame in the exemplar, the N best-matching participants are identified separately in each temporal unit of the input, and the matches are assigned scores. Matching scores are accumulated over time through voting that is insensitive to tracking errors and changes in action rates, and this produces a spatial localization of the N participating people. Their interaction is then localized in time using an efficient branch-and-bound search.

"hello". In a collection of hockey games, we might want all instances of a "three-on-one", and in nature we might be interested in localizing instances of distinctive group interactions among populations of animals, insects, or bacteria. Each of these cases would likely require distinct algorithms for detecting and tracking the agents, and each would benefit from action descriptors that are tuned for that setting. But beyond this, all of these scenarios can be abstracted as collections of (possibly fragmented and noisy) trajectories with accompanying time-varying action descriptors, and this is the abstraction on which we operate.

As depicted in Fig. 1, our approach is based on matching. Given an exemplar video of a distinctive group interaction involving a small handful of N agents, we detect and localize instances of similar interactions within a long video of a larger gathering of $M \geq N$ agents. We represent a group interaction as an ensemble of two types of time-varying descriptors: per-agent descriptors that encode the appearance and/or motion of each agent, and relative pairwise descriptors that encode the appearance and/or motion of each agent relative to another. Matching an exemplar interaction amounts to searching through space and time for ensembles that are similar in some sense. This approach avoids generating explicit semantic descriptions of group interactions, and it is advantageous when one lacks the vocabulary to precisely describe a class of interactions, or when they cannot easily be broken down according to a pre-defined grammar. To use our matching approach for recognition, we simply match an input video against a labeled gallery of exemplars and then extract a class label or ranked list of labels from the resulting scored matches.

In designing our detection system we face two main challenges. First, tracks may be fragmented and noisy, and we expect the presence of outlying fragments caused by false detections. We want an approach that can succeed in spite of these. Second, we expect that the same type of interaction can occur over different temporal extents and at variable rates within its temporal extent, so we want an approach insensitive to these "within-class" variations. We address these challenges using a voting-based approach, depicted in Fig. 1. First, the social descriptor-ensemble at each exemplar time unit is compared separately to each time unit of the input video, and the best-matching N participants in each

unit are identified along with their matching score (yellow and gray lines in Fig. 1). Second, weighted votes are accumulated from these unit-wise matches to obtain a final estimate of the N participants. Third and finally, the temporal extent of the interaction is determined through an efficient branch-and-bound search. Our designs for these three processing stages are tightly connected to each other and to our representation for interactions. Optimal unit-wise matching is made possible by our restriction to second order (individual and pairwise) action descriptors, and a metric learning procedure serves the dual role of improving voting (step two) and enabling efficient branch and bound search (step three).

Substantial progress has been made toward detecting activities of a single actor [13, 24, 22, 16, 7]. For analyzing interacting groups, previous approaches have considered cases in which: 1) there are no bystanders [11, 10, 3, 19, 21]; the interaction of interest is *a priori* localized in time [17, 4]; or both of these simultaneously [12, 20, 15]. A notable exception is [1], which like us, addresses the problem of localizing interactions in long videos that contain bystanders, albeit with a less flexible representation (more on this in Sec. 4).

We evaluate our approach using three different datasets: 1) the UT-Interaction Dataset [21]; 2) a new database of videos from an "interactive classroom" in which students self-organize in small group discussion (e.g. [5]); and 3) the Caltech Resident-Intruder Mouse dataset [2].

2. Matching and Localizing Interactions

We consider a video as a sequence of T temporal units that occur at a frequency equal to or less than the frame-rate of the raw video data. The duration of these T units is typically between one and a few raw video frames, and it is determined by the application-appropriate choice for temporal resolution of atomic action descriptors (e.g., positions, velocities, accelerations, histograms of flow, space-time SIFT). We assume the existence of an application-specific detection and tracking system that outputs M space-time tracks, which can be time-varying points, bounding boxes, silhouettes, or something else. Due to agent entry and exit, occlusions, and other tracking errors, not all M tracks will persist over all T frames, and some of the M tracks may correspond to short-lived false detections. The value of M is thus the total number of trajectory fragments that are identified with distinct agents.

With each track we associate ensembles of two types of descriptors. There are TM per-time-unit d_I-dimensional descriptors $\{\mathbf{f}_{m,t}\}$ where $\mathbf{f}_{m,t}$ encodes the mth agent's activity at time unit $t \in [1, T]$; and $TM(M-1)$ pairwise d_P-dimensional descriptors $\{\mathbf{g}_{m,m',t}\}$ where $\mathbf{g}_{m,m',t}$ encodes at time t the motion and/or appearance of agent m relative to agent $m', m' \neq m$. Loosely speaking, $\mathbf{g}_{m,m',t}$ captures the "influence" that agent m' has over agent m at time t. This influence is not symmetric in general, so typically $\mathbf{g}_{m,m',t} \neq \mathbf{g}_{m',m,t}$. We use the notation $\mathcal{Q}_t \triangleq \{\mathbf{f}_{m,t}, \mathbf{g}_{m,m',t}\}$ for the ensembles of all M tracks at time t, and $\mathcal{Q} \triangleq \{\mathcal{Q}_t\}_{1 \leq t \leq T}$ for the ensembles harvested from the entire input video. As mentioned above, the dimensions and entries in the descriptor vectors $\mathbf{f}, \mathbf{g}$ will be application dependent, and we consider a variety of examples in our experiments. Each exemplar video is processed in the very same way as the input video, so that an exemplar of $N \leq M$ participants over S time units is represented at each time $s \in [1, S]$ by the ensemble $\mathcal{D}_s \triangleq \{\mathbf{f}^D_{n,s}, \mathbf{g}^D_{n,n',s}\}$. We use the analogous notation $\mathcal{D} \triangleq \{\mathcal{D}_s\}_{1 \leq s \leq S}$ for the ensembles collected from the entire exemplar.

Given a collection of exemplars and an input video, our matching strategy is as follows. For each exemplar $\mathcal{D}$, we search through the input $\mathcal{Q}$ for the optimal match, identifying the set of N participants and localizing their interaction in time. The tracks corresponding to this best detection are then removed from $\mathcal{Q}$, and the procedure is repeated to find the second-best match, and so on. This provides multiple

ranked detections for each exemplar. In the end, we have a pool of space-time localizations from the input, with each of these "detected interactions" associated through similarity scores to one or several exemplars. To classify a detected interaction, we simply apply the majority of the category labels to the top-ranked exemplars associated with it. The remainder of this section describes in detail the process of locating the single best match for one exemplar.

2.1. Matching between Temporal Units

The first step in our framework is to separately compute the correspondence between the N exemplar agents at each time $s \in [1, S]$ and the optimal subset of $N \leq M$ of input agents at each time $t \in [1, T]$. We represent this N-to-M correspondence by the $N \times M$ binary matrix W, where the nm-th entry w_{nm} is one only when the nth exemplar agent is matched to the mth input agent. Matches must be unique, so these matrices must have one non-zero entry in each row and at most one non-zero entry in each column: $W\mathbf{1} = \mathbf{1}$ and $W^T\mathbf{1} \leq \mathbf{1}$. We use the symbol $\mathcal{W}$ to represent the space of all such matrices, *i.e.*, $\mathcal{W} \triangleq \{W \in \{0,1\}^{N\times M} | W\mathbf{1} = \mathbf{1}, W^T\mathbf{1} \leq \mathbf{1}\}$.

The quality of a correspondence is measured by the similarity between the individual and pairwise descriptors of the N selected input agents and those of the N exemplar agents. We formalize this by defining

$$\hat{D}(\mathcal{Q}_t, \mathcal{D}_s, W) = \sum_{nm} w_{nm} d_I(\mathbf{f}_{m,t}, \mathbf{f}^D_{n,s}) + \sum_{nmn'm'} w_{nm} w_{n'm'} d_P(\mathbf{g}_{m,m',s}, \mathbf{g}^D_{n,n',t}), \tag{1}$$

to be the dissimilarity between two instantaneous ensembles under a particular matching matrix W. We use Mahalonobis distances to compare descriptors in this expression, so that $d_I(\mathbf{f}, \mathbf{f}') = (\mathbf{f} - \mathbf{f}')^T \Sigma_I (\mathbf{f} - \mathbf{f}')$ and $d_P(\mathbf{g}, \mathbf{g}') = (\mathbf{g} - \mathbf{g}')^T \Sigma_P (\mathbf{g} - \mathbf{g}')$, with $\Sigma_I \succeq 0$ and $\Sigma_P \succeq 0$ positive semi-definite matrices learned from exemplar videos as will be described in Sec. 3.

Our immediate objective is to find the matching matrix $W \in \mathcal{W}$ that minimizes the score $\hat{D}(\mathcal{Q}_t, \mathcal{D}_s, W)$. Letting $\mathbf{w}$ be the vector formed by stacking the columns of W, the optimization can be expressed as

$$\min_{\mathbf{w}} \mathbf{c}^T\mathbf{w} + \mathbf{w}^T H \mathbf{w}, \text{s.t. } w_{nm} \in \{0,1\}, W\mathbf{1} = \mathbf{1}, W^T\mathbf{1} \leq \mathbf{1}, \tag{2}$$

where $\mathbf{c}$ is a $MN \times 1$ vector of distances between individual descriptors, $d_I(\mathbf{f}_{m,t}, \mathbf{f}^D_{n,s})$, and H is a $MN \times MN$ matrix of distances between pairwise descriptors $d_P(\mathbf{g}_{m,m',t}, \mathbf{g}^D_{n,n',s})$'s. This problem has integer constraints and is generally not convex, so we instead solve

$$\min_{\mathbf{w}} (\mathbf{c} + \hat{\mathbf{c}})^T\mathbf{w} + \mathbf{w}^T(H + \hat{H})\mathbf{w}, \text{s.t. } w_{nm} \in \{0,1\}, W\mathbf{1} = \mathbf{1}, W^T\mathbf{1} \leq \mathbf{1}, \tag{3}$$

where $\hat{\mathbf{c}} = [\sigma_1, \sigma_2, \cdots, \sigma_{MN}]^T$, $\hat{H} = \text{diag}\{-\sigma_1, -\sigma_2, \cdots, -\sigma_{MN}\}$, and each σ_i is a sufficiently large number greater than $\sum_{j=1, j\neq i}^{MN} |H_{ij}| + H_{ii}$.

Note that $\hat{H}$ imposes a negative strictly dominant diagonal to H and the quadratic term $\hat{H} + H$ is strictly negative definite. Therefore, (3) is a concave programming in the convex unit hypercube $[0,1]^{N\times M}$ and will achieve its minimum at one of the feasible vertices. The feasible vertices, meanwhile, are exactly the feasible solutions of (2), and at these vertices, the values of the objective of (3) are equal to those of (2) due to the cancellation brought by $\hat{\mathbf{c}}$. Therefore, by solving the much more efficient Problem (3) we obtain the exact solution for the original Problem (2). We solve (3) using the CVX toolbox [9].

For notational convenience, we define $D(\mathcal{Q}_t, \mathcal{D}_s) \triangleq \min_{W\in\mathcal{W}} \hat{D}(\mathcal{Q}_t, \mathcal{D}_s, W)$ to be the similarity between ensembles $\mathcal{Q}_t$ and $\mathcal{D}_s$, and $W^{t,s} \triangleq \arg\min_{W\in\mathcal{W}} \hat{D}(\mathcal{Q}_t, \mathcal{D}_s, W)$ to be the optimal instantaneous matching matrix that yields this similarity.

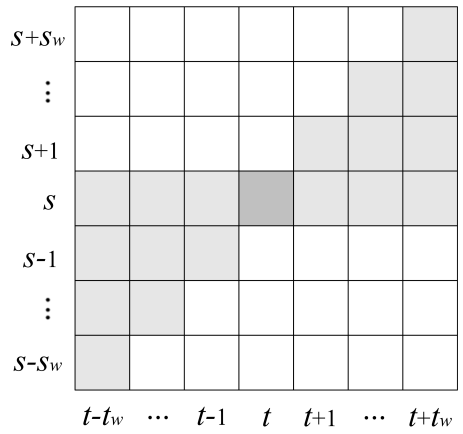

Figure 2. The temporal neighborhood used in to compute (4). See Sec. 2.2 for details.

2.2. Voting for Participant Identification

The next step is to accumulate participant information from noisy instantaneous matches $W^{t,s}$, with the goal of identifying a single optimal matching matrix, denoted $W^* \in \mathcal{W}$, that identifies a single consistent set of $N < M$ participants over the duration of the interaction being matched. We achieve this through voting, with the intuition being that the optimal matching W^* will occur relatively frequently among the instantaneous matches $\{W^{t,s}\}$. Each per-unit match casts a weighted vote, and to tally these votes we maintain two arrays both sized of $|\mathcal{W}|$. Each element of the first array counts the number of votes for a particular matching matrix, and the corresponding element in the second array maintains a cumulative sum of the weights for that matching matrix.

The weight of each vote is determined by two factors. The first is the dissimilarity between the descriptor-ensemble of the exemplar and that of the matched input agents $D(\mathcal{Q}_t, \mathcal{D}_s)$. The second is a measure of temporal consistency, with the intuition being that if the N-subset of agents is matched at temporal pair (t, s) is correct, the same N-subset of agents should be matched for other pairs (t', s') in small temporal neighborhoods of the exemplar and input video. We measure this using the ℓ_1 distance between matching matrices: $||W^{t,s} - W^{t',s'}||_1$. These two factors are combined to provide a vote's weight as

$$v(W^{t,s}) = \sum_{(t',s') \in \mathcal{N}(t,s)} (\|W^{t,s} - W^{t',s'}\|_1 + 1) D(\mathcal{Q}_{t'}, \mathcal{D}_{s'}), \tag{4}$$

where $\mathcal{N}(t, s)$ is a temporal neighborhood of (t, s) in which we enforce the consistency and it is depicted in Fig. 2, where the pair (t, s) is shown in black square and the neighborhood is shown as shaded area.

As a result, the voting procedure is shown in Algorithm 1, where in the last two steps we find among those matching matrices which receive a substantial number of supports from instantaneous matchings the best matching W^* with the lowest average dissimilarity to the exemplar. This idea is also illustrated in Fig. 1, where a thick matching line indicates a strong similarity (low weight v), and the agents receiving the lowest average weight are selected as participants.

2.3. Branch-and-Bound Temporal Localization

Our third step is to determine the starting time T_s and ending time T_e ($1 \leq T_s < T_e \leq T$) of the interaction. For this purpose, after the participants are determined through the best matching W^*, we

1. Clear both accumulator arrays;
2. For each $t \in [1, T]$, $s \in [1, S]$, increment the count for the matching matrix $W^{t,s}$ by 1, and increase the sum of weights
   in the companion array corresponding to $W^{t,s}$ by $v(W^{t,s})$;
3. Identify a subarray of matrices receiving more than $\frac{S}{2}$ counts,
   and normalize the sum of weights in the companion subarray
   by corresponding counts;
4. Report the matching matrix W^* to be the one in the subarray receiving the minimum normalized sum of weights.

Algorithm 1: Voting procedure for identify the participants (*i.e.*, the best overall matching W^*).

recompute for all (t, s) pairs the dissimilarities under this best matching $\hat{D}(\mathcal{Q}_t, \mathcal{D}_s, W^*)$, between the interaction of the individuals selected by W^* at time t and the exemplar at time s. We then compute $D^*(t) = \min_s \hat{D}(\mathcal{Q}_t, \mathcal{D}_s, W^*)$, the minimal dissimilarity of the input interaction by the selected participants at time t to the entire exemplar, and $s^*(t) = \arg\min_s \hat{D}(\mathcal{Q}_t, \mathcal{D}_s, W^*)$, the time in the exemplar at which the input at time t exhibits this maximum similarity.

If the N selected agents in the input perform the same interaction during $T_s \leq t \leq T_e$ as those in the exemplar during $1 \leq s \leq S$, they will be visually similar and temporally aligned: The minimum-scores $D^*(t)$ will be small for $T_s \leq t \leq T_e$, and each minimum-score time $s^*(t)$ will be in the same relative location in $[1, S]$ as t is in $[T_s, T_e]$. Our aim is to design an objective function that encodes preferences for both to enable efficient temporal search for the optimal T_s and T_e. As interactions occur at variable rates within their temporal extent, we use a temporal pyramid to efficiently measure alignment in a way that also respects these variations. The pyramid contains L levels indexed by $l \in [0, 1, \cdots, L-1]$ and equal-length cells at the lth level indexed by $i \in [0, 1, \cdots, 2^l - 1]$. The indicator $\mathbf{1}(t \in \mathcal{C}(T_s, T_e, l, i))$ is one whenever t occurs in the ith cell of the lth level of the pyramid over $[T_s, T_e]$, and $\mathbf{1}(s \in \mathcal{C}(1, S, l, i))$ is the analogous indicator for the exemplar. Then, when considering an input interval $[T_s, T_s]$ we measure alignment for each time-pair (t, s) using

$$k(t, T_s, T_e, s, 1, S) \triangleq \sum_{l=0}^{L-1} \sum_{i=1}^{2^l} \mathbf{1}(t \in \mathcal{C}(T_s, T_e, l, i)) \mathbf{1}(s \in \mathcal{C}(1, S, l, i)). \tag{5}$$

Let (t_s, t_e) be the true, unknown starting and ending times of the detected interaction in the input video, and suppose that the input descriptor-ensemble over this interval exactly matches that of the exemplar. To determine good estimates for the interval (t_s, t_e) we define a cost that is a product of the temporal alignment and visual similarity summed over the candidate interval:

$$f(T_s, T_e) \triangleq \sum_{t=T_s}^{T_e} k(t, T_s, T_e, s^*(t), 1, S)(D^*(t) - 1). \tag{6}$$

As will be described in the next section, we use metric learning to ensure that the dissimilarities $D^*(t)$ are driven toward 0 in the true interval $[t_s, t_e]$ and toward 2 otherwise. This means that the summand in (6) considered as a function of t assumes a negative value in the desired interval $t_s \leq t \leq t_e$ and a positive value otherwise, as denoted as $q(t)$ and depicted in the bottom of Fig. 1. This ensures that the function f achieves the global minimum if and only if the interval $[T_s, T_e]$ is exactly aligned to the

desirable interval $[t_s, t_e]$. As a result, $f(T_s, T_e)$ satisfies the "quality function" requirements described in [14] that enables the use of an efficient branch-and-bound search for the globally optimal interval (T_s, T_e) without the need for an exhaustive sliding window.

Specifically, we first specify the spaces where T_s and T_e may take a value. We denote the length of the shortest exemplar activity as T_{min}, then we assume $1 \leq T_s \leq T - T_{min} + 1$ and $T_{min} + 1 \leq T_e \leq T$. Additional constraint may be imposed, such as $T_{min} \leq T_e - T_s$. Given these information, the temporal branch-and-bound algorithm, as a companion to the 2-D case studied in [14], can be derived as in Algorithm 2. In this algorithm, $\hat{f}(T_{s,low}, T_{s,upp}, T_{e,low}, T_{e,upp})$ is a lower bound of the values of the quality function evaluated on all intervals enclosed in $[T_{s,low}, T_{s,upp}] \times [T_{e,low}, T_{e,upp}]$. To calculate this lower bound, we define

$$\hat{f}(T_{s,low}, T_{s,upp}, T_{e,low}, T_{e,upp}) = \sum_{l=0}^{L-1} \sum_{i=1}^{2^l} \hat{f}(T_{s,low}^{l,i}, T_{s,upp}^{l,i}, T_{e,low}^{l,i}, T_{e,upp}^{l,i}) \tag{7}$$

where $T_s^{i,l}, T_e^{i,l}$ are the boundaries of cell $\mathcal{C}(T_s, T_e, l, i)$. In other words, we use the summation of the lower bounds of all cells in the pyramid as the lower bound of the entire interval. The evaluation of $\hat{f}(T_{s,low}^{l,i}, T_{s,upp}^{l,i}, T_{e,low}^{l,i}, T_{e,upp}^{l,i})$, however, is a $\mathcal{O}(1)$ operation with the help of integral dissimilarities $I(t)$ of those negative group dissimilarities $D^*(t)$ over t. Specifically, let

$$I(t) = \sum_{t'=1}^{t} \min(0, D^*(t)) \tag{8}$$

which only needs to be computed once. Then the lower bound for the cell $\mathcal{C}(T_s, T_e, l, i)$ can be obtained as

$$\hat{f}(T_{s,low}^{l,i}, T_{s,upp}^{l,i}, T_{e,low}^{l,i}, T_{e,upp}^{l,i}) = I(T_{e,upp}^{l,i}) - I(T_{s,low}^{l,i}). \tag{9}$$

We have described the approach to locate the single best match for one exemplar. Though it operates on continuous tracks that are achievable in all experiments in Sec. 4, the process can also handle moderately broken tracks by setting the descriptor values of missing temporal units to be sufficiently large (or small) so as not to be matched with any exemplar agents. As long as the number of missing units is small, correct matches still dominate during voting. Then, $D^*(t)$ and $s^*(t)$ can be interpolated from adjacent units.

3. Descriptor Metric Learning

As mentioned in Sec. 2.1, we learn matrices Σ_I, Σ_P for the Mahalanobis distances $d_I(\mathbf{f}, \mathbf{f}')$ and $d_P(\mathbf{g}, \mathbf{g}')$, so that the learned metrics can: 1) enhance discrimination between exemplar categories by ensuring that distances are smaller when descriptors are drawn from roughly the same temporal location within a labeled exemplar of the same category, and larger otherwise; and 2) enhance the accuracy of temporal localization by ensuring that distances between labeled ensembles and unlabeled "background" ensembles are large. The combination of 1) and 2) leads to more accurate spatial localizations of participants (*i.e.* better W^* as discussed in Sec. 2.2), and induces the "quality function" conditions required for efficient temporal localization by branch-and-bound (Sec. 2.3). We achieve all of these benefits simultaneously by using an adaptation of the Large Margin Nearest Neighbor (LMNN) framework [23].

For each application scenario, we use a training set of exemplar videos—possibly having varying numbers of agents N—that are annotated with start/end times, category labels, N-agent correspondences

1. Initialize: Let $T_{s,low} = 1$, $T_{s,upp} = T - T_{min} + 1$, $T_{e,low} = T_{min} + 1$, and $T_{e,upp} = T$; Initialize priority queue Q as empty;

2. Do

   - If $T_{s,upp} - T_{s,low} \geq T_{e,upp} - T_{e,low}$
     $T^{(1)}_{s,low} \leftarrow T_{s,low}, T^{(1)}_{s,upp} \leftarrow T_{s,low} + \frac{T_{s,upp} - T_{s,low}}{2}, T^{(1)}_{e,low} \leftarrow T_{e,low}, T^{(1)}_{e,upp} \leftarrow T_{e,upp}$,
     $T^{(2)}_{s,low} \leftarrow T_{s,low} + \frac{T_{s,upp} - T_{s,low}}{2}, T^{(2)}_{s,upp} \leftarrow T_{s,upp}, T^{(2)}_{e,low} \leftarrow T_{e,low}, T^{(2)}_{e,upp} \leftarrow T_{e,upp}$;
     else
     $T^{(1)}_{s,low} \leftarrow T_{s,low}, T^{(1)}_{s,upp} \leftarrow T_{s,upp}, T^{(1)}_{e,low} \leftarrow T_{e,low}, T^{(1)}_{e,upp} \leftarrow T_{e,low} + \frac{T_{e,upp} - T_{e,low}}{2}, T^{(2)}_{s,low} \leftarrow T_{s,low}$,
     $T^{(2)}_{s,upp} \leftarrow T_{s,upp}, T^{(2)}_{e,low} \leftarrow T_{e,low} + \frac{T_{e,upp} - T_{e,low}}{2}, T^{(2)}_{e,upp} \leftarrow T_{e,upp}$;
   - If $T_{min} \leq T^{(1)}_{e,upp} - T^{(1)}_{s,low}$, push $(T^{(1)}_{s,low}, T^{(1)}_{s,upp}, T^{(1)}_{e,low}, T^{(1)}_{e,upp}, \hat{f}(T^{(1)}_{s,low}, T^{(1)}_{s,upp}, T^{(1)}_{e,low}, T^{(1)}_{e,upp}))$ into Q;
   - If $T_{min} \leq T^{(2)}_{e,upp} - T^{(2)}_{s,low}$, push $(T^{(2)}_{s,low}, T^{(2)}_{s,upp}, T^{(2)}_{e,low}, T^{(2)}_{e,upp}, \hat{f}(T^{(2)}_{s,low}, T^{(2)}_{s,upp}, T^{(2)}_{e,low}, T^{(2)}_{e,upp}))$ into Q;
   - Let $(T_{s,low}, T_{s,upp}, T_{e,low}, T_{e,upp})$ be the tuple in Q achieving the minimal $\hat{f}$;

   Until $T_{s,low} = T_{s,upp}, T_{e,low} = T_{e,upp}$.

3. Output: $T_s \leftarrow T_{s,low}, T_e \leftarrow T_{e,low}$.

Algorithm 2: Branch-and-bound search for temporal localization.

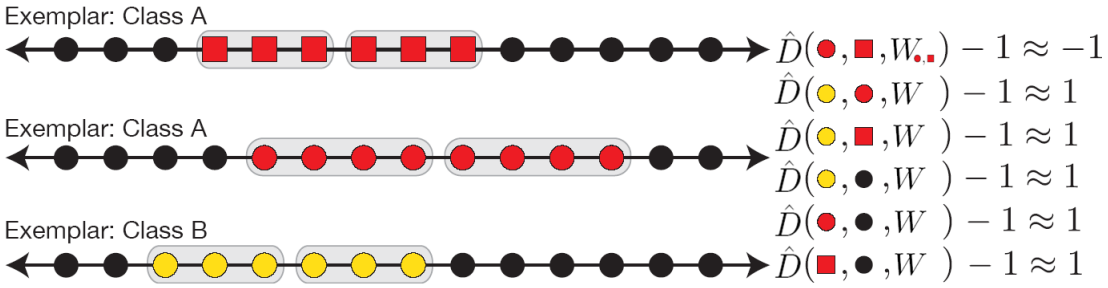

Figure 3. Constraints used in discriminative metric learning. Each row is an annotated two-cell exemplar with markers representing instantaneous descriptor-ensembles at each time unit. For discrimination between interaction categories, distances between ensembles of the same class (red circles and red squares) should be small whenever they occur in the same cell number; and distances for different classes (red vs. yellow) should be large. For effective and efficient temporal localization, distances between ensembles at labeled times and unlabeled "background" times (black circles) should be large, and all distances should be offset by -1.

between exemplars of the same category. We use unlabeled time units in the videos as "background" samples. Intuitively, the learned metrics should satisfy the six types of constraints shown in Fig. 3. This figure depicts three different exemplar videos in which a subset of time units have been labeled as being distinctive interactions of two different classes. In this example, each labeled exemplar is shown as being divided into two cells; these correspond to the lowest level of the temporal pyramid described in Sec. 2.3. The first three constraints in the list enhance discrimination between categories, while the last three enhance the accuracy of temporal localization. Offsetting all of the distances by -1 ensures that the summand in (6) assumes proper negative values as required for branch-and-bound search.

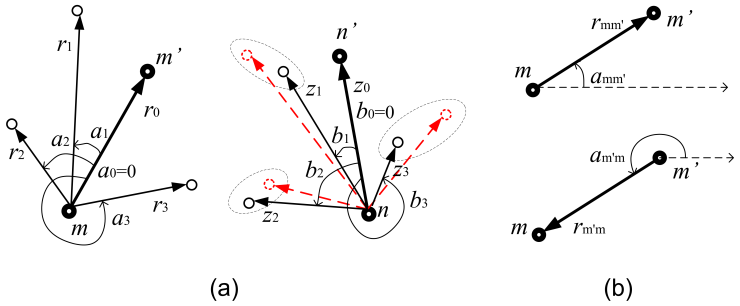

Figure 4. For the classroom dataset, pairwise descriptors for groups comprised of (a) three or more participants, and (b) two participants. See text for details.

We enforce these constraints through an LMNN framework by constructing two collections from our database exemplars. The collection $\mathcal{P}$ contains all pairs of instantaneous interaction ensembles that are of the same category (red circles and red squares in Fig. 3) and occur roughly in the same temporal location within the interaction instances (*i.e.*, in the same cell of the lowest level of the temporal pyramids), together with their "ground-truth" matchings. The collection $\mathcal{M}$ is comprised of ordered triples (h, k, l) in which ensemble h is the same category as ensemble k and ensemble l is either of a different category or background. Having defined these two collections, each Mahalanobis metric is found by solving

$$\begin{aligned} &\min_{\Sigma_I, \Sigma_P} \sum_{(u,v)\in\mathcal{P}} \hat{D}(\mathcal{D}_u, \mathcal{D}_v, W_{u,v}) + \gamma \sum_{(h,k,l)\in\mathcal{M}} \xi_{h,k,l}, \\ &\text{s.t.} \hat{D}(\mathcal{D}_h, \mathcal{D}_l, W) - \hat{D}(\mathcal{D}_h, \mathcal{D}_k, W_{h,k}) \geq 2 - \xi_{h,k,l}, \Sigma_I \succeq 0, \Sigma_P \succeq 0, \xi_{h,k,l} \geq 0, \end{aligned} \tag{10}$$

where $W_{u,v}$ is the "ground-truth" matching for pair (u, v) and W is an arbitrary matching[1]. The minimization over either Σ_I or Σ_P is exactly a LMNN problem [23], and we apply LMNN multiple times to learn a distinct pair (Σ_I, Σ_P) for each value of N that exists in the training set.

4. Experiments

We evaluate our approach on three datasets, two that involve humans and one that involves mice. The datasets are very different from one another, with distinct types of individual and pairwise descriptors that are appropriate for that environment. In all experiments we use four-level temporal pyramids for the interactions and we set the time unit to be half the duration of the cells in the lowest level. Neighborhood sizes t_w and s_w are taken as a quarter of the length of a cell on the bottom of the pyramid[2].

4.1. Classroom Interaction Database

We collected and annotated a new database of videos capturing students' behaviors over five hour-long sessions in an interactive classroom. As shown in the left-most images of Fig. 7, the students are seated in a regular lecture hall and are observed by a camera array with non-overlapping fields of

[1]It is useful to add to the collection $\mathcal{M}$ additional triples in which l is derived from same-category same-cell pairs but with permuted incorrect matching matrices. In this case $\hat{D}(\mathcal{D}_h, \mathcal{D}_l, W)$ in (10) is replaced by $\hat{D}(\mathcal{D}_h, \mathcal{D}_l, \bar{W}_{h,l})$, where $\bar{W}_{h,l}$ is the permuted incorrect matching.

[2]When the neighborhood extends out of video boundary, we only consider the cells within the boundary and normalize the vote by the number of cells actually involved.

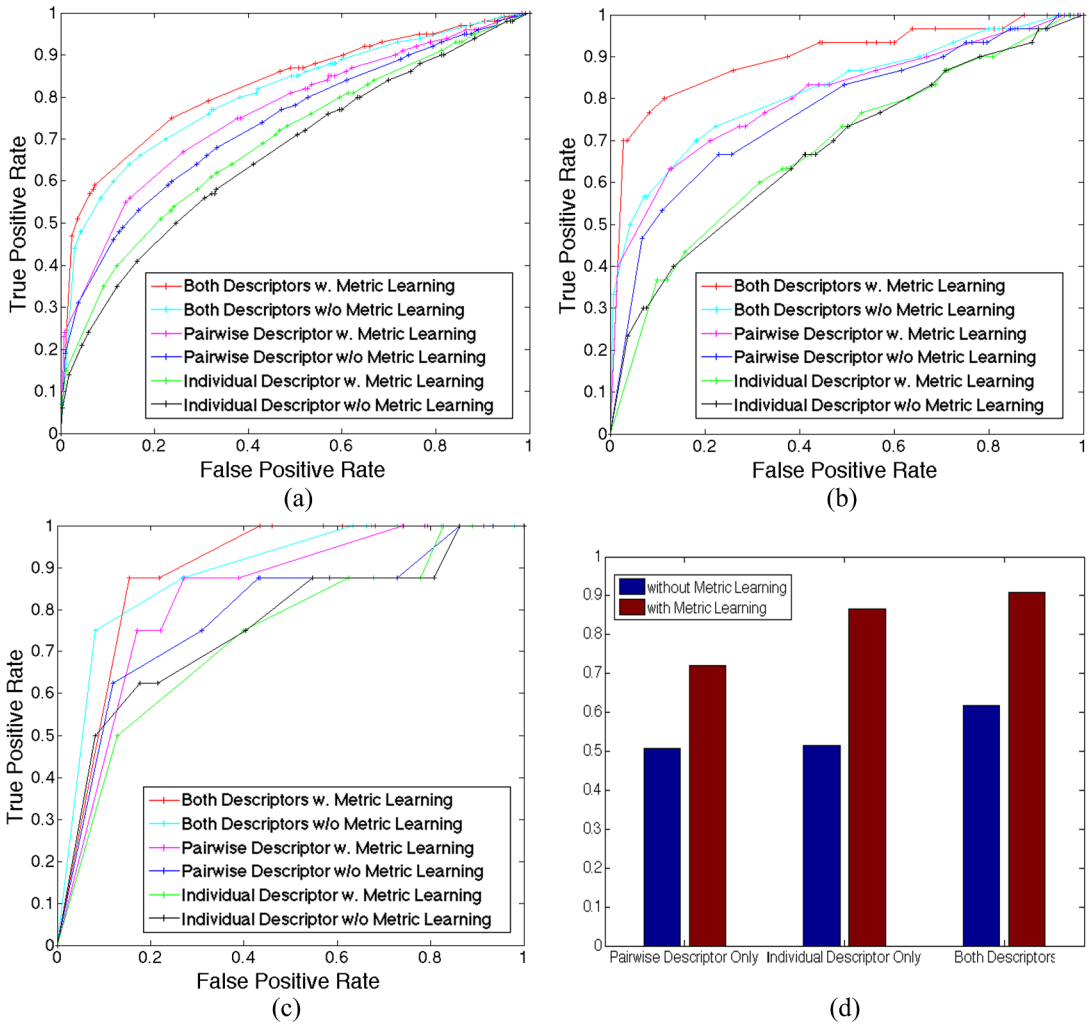

Figure 5. (a)(b)(c) ROC curves for identifying the participants of an two-person, three-person, and four-person interactions using the proposed approach and baselines. (d) Temporal localization accuracies using the proposed approach with and without metric learning, using individual and/or pairwise descriptors.

view. The classroom is "interactive" because at various times throughout the lecture students are invited to engage in ad-hoc group discussions about problems provided by the instructor (see, e.g., [5]). The ad-hoc groups can form within and across seating rows, and detecting them is a challenge because the number of by-standers is much larger than the number of participants (M is between 10 and 20 while N is between 2 and 4), video quality is limited (low light, 15fps), and the visual cues for interaction are quite subtle. The ability to automatically detect such interactions is important for education researchers, however, since it can help in understanding how students self-organize into groups, and which geometric configurations of groups lead to improved educational outcomes [5].

We applied an OpenCV face detector and generated long tracks of the bounding boxes using a combination of OpenCV mean-shift tracking and optical flow. In consultation with education experts, we manually identified the participants and start/end times of all two-person, three-person, and four-person interactions, obtaining 254 two-person, 112 three-person, and 16 four-person interactions in total. We

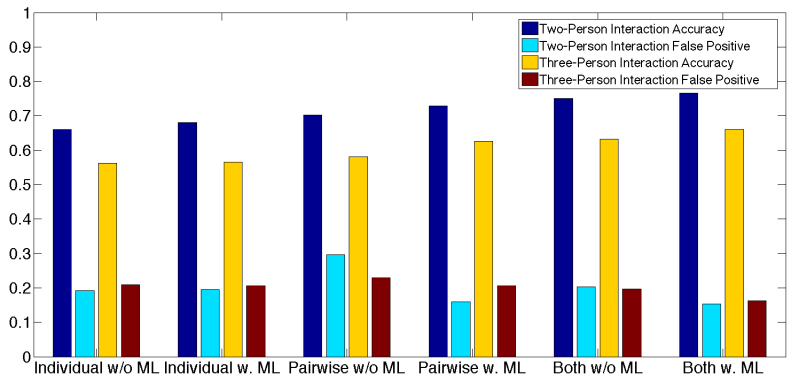

Figure 6. Average classification accuracies and false positives for two-person and three-person interactions (Individual and/or pairwise descriptors, with or without metric learning (ML)).

defined interaction categories based on the geometric configurations of the participants: three categories for 2-person interactions (same row; different rows with left agent in front; different rows with right agent in front) and four categories for 3-person interactions. Samples of these exemplars can be found in the columns (c1)-(c3) of Fig. 7. The annotated interactions range from a few seconds to tens-of-seconds in length. Since the raw videos arise from five different hour-long session, we adopt a leave-one-session-out evaluation scheme in partitioning training samples (exemplars) from test samples (inputs). Also, for each split of the data we manually eliminate the false detections and tracks in the exemplars, while leaving them present in the test samples.

We use a coarse representation of the head pose as the individual descriptor. Specifically, we compute the Histogram of Oriented Gradient (HOG) feature within each temporal unit and each detection box, and train nine one-versus-all SVMs on these HOG features to estimate the likelihood of nine head poses (front, left, lower-left, lower-front, lower-right, right, back-right, and back-left) for a new face in the input. The nine-dimensional likelihood vector serves as our individual descriptor. Meanwhile, we derive the pairwise descriptor for three or more individuals based on the geometrical configurations of the bounding boxes. As shown in the left panel of Fig. 4(a), for a pairwise descriptor of target m relative to target m' among five targets, we compute the distances r_i between all others and m, and the relative angles a_i between the connecting vectors and $\overrightarrow{mm'}$, and combine all these geometric quantities into a pairwise descriptor $\mathbf{g}_{m,m',t}$. When computing $\mathbf{g}_{n,n',s}$ in the input (right panel of Fig. 4(a)), we align $\overrightarrow{nn'}$ against $\overrightarrow{mm'}$ and predict the locations of the three individuals (shown in red), and compute the true distances z_i and relative angles b_i by locating the nearest individuals to the predicted locations. This pairwise representation achieves invariance under similarity transforms. For two-person interaction, we simply use the distance and the relative angles against the right horizontal axis (Fig. 4(b)).

We begin by looking at accuracy of detection, where we ignore the inferred interaction categories and simply measure the systems ability to detect when an interaction has occurred. Fig. 5 (a–c) show detection ROC curves for different group sizes with various parts of the system turned off. This includes using only one of the individual or pairwise descriptors, and using metric learning (optimized Σ_I, Σ_P) or not (Σ_I and Σ_P set to identity matrices). Using all parts of the system yields the best results, and we note that performance improves as the number of participants N increases. The latter is due to the fact that interaction patterns are more salient when more pairwise information is available.

Next, we study classification performance for 2-person and 3-person interactions, where we measure

the accuracy of inferred interaction categories. (Due to the small number of 4-person interactions in our dataset, we did not define categories for them.) As before we do this with various parts of the system turned off, and Fig. 6 shows the average true positive rates versus false positives when further classifying detected interactions into the three or four categories. Again we see that performance improves when more parts of the system turned on. We can also draw a contrast with the detection results of Fig. 5 where the pairwise features are substantially more important than the individual ones. This difference diminishes in Fig. 6, likely because we are measuring performance on correctly-detected interactions, where head pose provides stronger evidence than spatial configuration. An improved head-pose estimator or a more sophisticated description of body pose can be expected to further improved classification performance. Finally, we investigate the temporal localization performance, for which we compute the ratio of the intersection to the union of the estimated interval and the annotated interval, and we show the averages in Fig. 5 (d).

Fig. 7 shows some successes and failures of detection and matching. The second row shows a false detection (blue dashed box), where two people are not interacting but exhibit head poses similar to those of an interaction. In the fourth row, a three-person interaction is correctly identified even though the third associated exemplar is from a different category (two looking right). In the other rows, two-person and three-person interactions are correctly detected and matched with exemplars.

To evaluate the efficiency of the algorithm, we replace the optimal matching method with an exhaustive enumeration of all possible matchings. We also apply temporal sliding windows at eight scales ranging from half to twice of the exemplar length, stopping using the remaining scales whenever the current window achieves the same quality function value as the branch-and-bound. We show the average computation time for one match between an exemplar and an input on a 8-core 2.8GHz Macintosh in Table 1, where we see clear savings for the proposed approach.

Table 1. Computational cost comparison for the proposed matching approach and baselines (in seconds).

# of Participants	2	3	4
Exhaustive+Sliding Window	17.2	60.4	253.2
Exhaustive+Branch and Bound	12.6	27.6	59.7
Optimal Pairing+Sliding Window	12.4	23.2	40.8
Proposed	8.0	19.8	32.3

4.2. UT-Interaction Dataset.

For comparison to the state-of-art, we evaluate our approach on the UT-Interaction dataset [21]. We follow the protocol defined in previous work [21, 1]: 20% of available interaction annotations are used as exemplars for training, and the remaining (non-annotated) sequences are used for testing. For individual descriptors, we use 32-dimensional histogram of spatio-temporal features developed by [6] in each unit and each bounding box, which is constructed by applying PCA to a k-means-clustered, 500-word vocabulary. For pairwise descriptors, we use the difference between two 32-dimensional histograms computed for each of the two humans. The optical flow is computed using OpenCV, and histograms are comprised of 8 directions and 4 magnitudes. Training examples are manually examined to ensure error-less per-human bounding boxes, and for testing, we use an off-the-shelf human detector [8], and associate the detected boxes across frames to form continuous tracks.

Table 2 compares recognition accuracy and false-alarm rates to those of previous work [21, 1]. For our system, we consider one database exemplar at a time, compute its maximal response over the input video,

Figure 7. Examples of social interaction detection and matching on the classroom interaction database. Each row is an example of detecting a salient interaction from an input. (a) the input; (b) detected social interaction; (c-1) to (c-3) top three associated database exemplars that support the detection. (Blocked faces correspond to students who did not consent to their images appearing in publications.)

Table 2. Classification accuracies and false positive (FP) rates for the proposed method and baselines on the UT-Interaction dataset.

	Accuracy ([21], [1], ours)	FP Rate ([21], [1], ours)
Hug	(0.875, 0.904, 1.00)	(0.075, 0.055, 0.00)
Kick	(0.750, 0.775, 0.875)	(0.138, 0.108, 0.063)
Point	(0.625, 0.663, 0.750)	(0.025, 0.025, 0.088)
Punch	(0.500, 0.632, 0.750)	(0.201, 0.154, 0.138)
Push	(0.750, 0.782 , 0.750)	(0.125, 0.101, 0.138)
Shake Hands	(0.750, 0.789, 1.00)	(0.088, 0.060, 0.00)
Average	(0.708, 0.758, 0.854)	(0.108, 0.083, 0.071)

and claim a true positive only when both the class-label and the identified participants are simultaneously correct. Otherwise a false positive is indicated for that exemplar class. By these measures, our approach

provides improved accuracy and competitive false positive rates. Next we study detection in terms of both temporal localization and participant identification. For temporal localization, we follow the protocol of [1] by indicating a true-positive when there is correct classification and more than a 50% ratio between the intersection and union of the estimated temporal interval and the ground-truth. We achieve a slightly smaller area under ROC curve than the two baselines, as shown in Table 3, but point out that differences are hard to interpret because the temporal boundaries are somewhat ambiguous for the consecutively-executed interactions in the dataset. To assess participant identification, we enforce a stricter true-positive criterion that requires 100% correct identification instead of the 50% value used in [1] and our system still outperforms the method of [1]. We attribute this to the fact that we explicitly discriminate interactions and participants in the form of tracks of bounding boxes, while [1] does not do so but simply explains an input using a non-discriminative generative model.

Table 3. Area under ROC curve for the proposed method and the baselines on UT-Interaction dataset.

	[21]	[1]	ours
Temporal Localization	0.91	0.94	0.89
Participants Identification	N/A	0.87	0.93

As before, we disable components of our system to explore the effectiveness of combining both individual and pairwise descriptors, and using metric learning. The performance comparison is show in Table 4. It is interesting to see the pairwise descriptor plays a more crucial role for this dataset: A significant performance drop arises when we only consider individual action descriptors.

Table 4. Classification accuracies and false positive (FP) rates comparison on UT-Interaction dataset for evaluating the effectiveness of different components of the proposed approach: Individual and/or pairwise descriptors, with or without metric learning (ML).

	Individual only	pairwise only	Both
Accur. w. ML	0.688	0.813	0.854
Accur. w/o ML	0.647	0.750	0.771
FP Rate w. ML	0.125	0.096	0.071
FP Rate w/o ML	0.163	0.113	0.083

4.3. Caltech Resident-Intruder Mouse Dataset

We also tested the approach on Caltech Resident-Intruder Mouse Dataset [2], which contains long video sequences recording pair-wise interactions between two mice. Behaviors are categorized into 12 different mutually exclusive action types, plus an 'other' category indicating no behavior of interest is occurring. A video typically lasts around 10 minutes at 25fps with a resolution of 640x480 pixels. Every video frame is labeled with one of the thirteen ground-truth categories, resulting in a segmentation of the videos into action intervals. For more details please refer to [2]. Note that in all videos are pairwise interactions without 'by-standers' (*i.e.* $M = N$), our experiment on this dataset is not meant to distinguish the participants, but to demonstrate that our approach can be directly used for a traditional task of temporal segmentation and classification without any changes.

We exactly follow the training/testing partitions provided by the dataset. We extract the spatio-temporal interest points (STIP) based appearance features and compute trajectory-based features from the tracks provided with the dataset as [2] does (See [2] for details). Differently from [2], we only compute STIP based features inside the bounding boxes enclosing the mice. The trajectory-based features (position, velocities, etc.) consists of those describing the motion of each individual mouse and

those describing the relative motion between two mice. We denote the former as T_ind and the latter as T_pair. For features arising only from trajectories, there can be two possible working modes: Using all trajectory-based features as individual descriptors (denoted as Trajectory_1), or using T_ind as individual descriptors and using those describing pairs' motions as pairwise descriptors (denoted as Trajectory_2). To classify an temporal interval in a test video that is successfully matched to one or more exemplars, we simply read the label of the top-scoring exemplar. Table 5 shows the results using the error metric (frame-wise accuracy) defined in [2]. It is evident that splitting the motion into individual ones and pairwise ones and learning separate metrics for them is advantageous. Motion trajectory information is much more important than local STIP-based features, which is not surprising given the limited articulation of the agents.

Table 5. Accuracies for the proposed method and the baselines on Caltech Resident-Intruder Mouse Dataset. (%)

	Trajectory_1	Trajectory_2	STIP	Both
[2] w/o. context	52.3	52.3	29.3	53.1
[2] w. context	58.3	58.3	43.0	61.2
Ours w/o. ML	45.6	49.4	18.8	50.9
Ours w. ML	54.5	66.0	31.7	62.9

It is observed in [2] that accuracy varies with the length of the interaction and with the length of window within which the local feature is computed. To investigate the performance of our approach on different lengths of the interaction as compared to [2], we implemented the approach in [2] using trajectory-based features and one-level auto-context classifier. As shown by the result in Fig. 8, our approach is particularly better at localizing longer interactions though [2] demonstrates its advantage under a shorter feature window on shorter interactions.

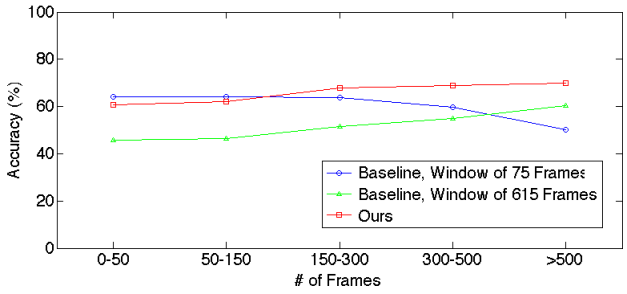

Figure 8. Accuracy comparison for varying length of interactions between [2] and our approach.

5. Conclusion.

We introduced a voting-based approach for detecting and localizing small-group interactions within larger social gatherings. The approach is based on matching against exemplars, and it avoids the need for any explicit semantic description of a group interaction. Since it operates on agent tracks, it is also quite flexible and can be applied in many different multi-agent scenarios, provided that the environment-specific individual descriptor and the environment-specific pairwise descriptor are properly defined. As practical detection and tracking continue to improve, we expect the opportunities for this type of analysis to expand.

We represent group interactions as collections of individual and pairwise descriptors (1st and 2nd order), and our results suggest that this is effective for groups of up to four agents. Higher-order interaction descriptors may play a more important role for larger interacting groups, and this may be a useful future research direction as new datasets become available. It may also be worth considering more flexible schemes for breaking an interaction into parts. We use a simple combination of descriptor collection and temporal pyramid, but one could imagine using a (learned) tree of space-time parts, analogous to how spatial parts-based models are used for object detection.

Acknowledgement The authors thank Ely Spears, Laura Tucker, Brian Lukoff and Eric Mazur for many helpful discussions. This research was supported by the National Science Foundation through awards IIS-0835338, IIS-0905243, and CNS-0708895.

References

[1] M. Amer and S. Todorovic. A chains model for localizing participants of group activities in videos. In *ICCV*, 2011.

[2] X. Burgos-Artizzu, P. Dollar, D. Lin, D. Anderson, and P. Perona. Social behavior recognition in continuous videos. In *CVPR*, 2012.

[3] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

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

[5] C. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69:970–977, 2001.

[6] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[7] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

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

[9] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, Apr. 2011.

[10] A. Hakeem and M. Shah. Learning, detection and representation of multi-agent events in videos. *Artificial Intelligence*, 171:586 – 605, 2007.

[11] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *ICCV*, 2001.

[12] S. Intille and A. Bobick. Recognizing planned, multiperson action. *CVIU*, 81:414 – 445, 2001.

[13] Y. Ke, R. Sukthankar, and M. Hebert. Volumetric features for video event detection. *IJCV*, 88(3):339 – 362, 2010.

[14] C. Lampert, M. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *PAMI*, 31(12):2129–2142, 2011.

[15] T. Lan, Y. Wang, W. Yang, S. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *PAMI*, 34(8):1549–1562, 2012.

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

[17] R. Li and R. Chellappa. Group motion segmentation using a spatio-temporal driving force model. In *CVPR*, 2010.

[18] R. Li, P. Porfilio, and T. Zickler. Finding group interactions in social clutter. Technical Report TR-01-13, Harvard School of Engineering and Applied Sciences, ftp://ftp.deas.harvard.edu/techreports/tr-01-13.pdf, 2013.

[19] V. Morariu and L. Davis. Multi-agent event recognition in structured scenarios. In *CVPR*, 2011.

[20] B. Ni, S. Yan, and A. Kassim. Recognizing human group activities by localized causalities. In *CVPR*, 2009.

[21] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[22] E. Shechtman and M. Irani. Space-time behavioral correlation. In *CVPR*, 2005.

[23] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2005.

[24] J. Yuan, Z. Liu, and Y. Wu. Discriminative video pattern search for efficient action detection. *PAMI*, 33(9):1728 – 1743, 2011.