Supplemental Material: 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

1. Details of Branch-and-Bound Localization

To apply branch-and-bound algorithm to minimize the quality function (6) in the main paper, 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 [2], can be derived as in Algorithm 1. 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

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

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

$$I(t) = \sum_{t'=1}^{t} \min(0, D^*(t)) \tag{2}$$

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

$$\hat{f}(T_{s,low}^{l,i}, T_{s,upp}^{l,i}, T_{e,low}^{l,i}, T_{e,upp}^{l,i}) = I(T_{e,upp}^{l,i}) - I(T_{s,low}^{l,i}). \tag{3}$$

2. Additional Experiment Results

2.1. Computational Cost for the Classroom Interaction Database

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.

2.2. Caltech Resident-Intruder Mouse Dataset

We also tested the approach on Caltech Resident-Intruder Mouse Dataset [1], 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 [1]. Note that in all videos are pair-wise interactions without ‘by-standers’ (*i.e.* $M = N$), our experiment on this dataset is not meant to distinguish the participants,

```
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 1: Branch-and-bound search for temporal localization.

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

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 [1] does (See [1] for details). Differently from [1], 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 2 shows the results using the error metric (frame-wise accuracy) defined in [1]. 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 2. Accuracies for the proposed method and the baselines on Caltech Resident-Intruder Mouse Dataset. (%)

	Trajectory_1	Trajectory_2	STIP	Both
[1] w/o. context	52.3	52.3	29.3	53.1
[1] 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 [1] 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 [1], we implemented the approach in [1] using trajectory-based features and one-level auto-context classifier. As shown by the result in Fig. 1, our approach is particularly better at localizing longer interactions though [1] demonstrates its advantage under a shorter feature window on shorter interactions.

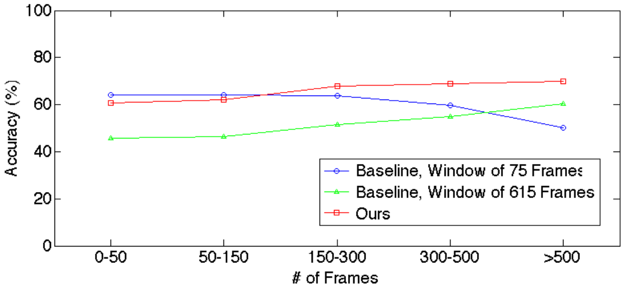

Figure 1. Accuracy comparison for varying length of interactions between [1] and our approach.

References

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

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