Supplemental Material : A Unified Framework for Multi-Target Tracking and Collective Activity Recognition

Wongun Choi and Silvio Savarese

Electrical and Computer Engineering, University of Michigan, Ann Arbor, USA
{wgchoi,silvio}@umich.edu

1 Interaction Feature

In this paper, we model the interaction feature as a combination of three types of relative motion features, ψ_l, ψ_p, and ψ_a. Each of the feature vector encodes relative motion (distance and velocity), one's location in another's viewpoint, and co-occuring atomic action. All of them are represented as a histogram so as to capture a non-parametric statistics of interactions.

• ψ_l is a feature vector that captures the relative position of a pair of people. In order to describe the motion of one respect to the other, ψ_l is represented as a histogram of velocity and location difference between the two within a temporal window $(t - \triangle t,\ t + \triangle t)$.

• ψ_p encodes a person's location with respect to the other's viewpoint. First, we define the i^{th} target centric coordinate system for each time t by translating the origin of the system to the location of the target i and rotating the x axis along the viewing direction (pose) of the target i. At each time stamp t in the temporal window, the angle of each target within the others' coordinate system is computed and discretized angle is obtained (see Fig.1) in order to describe the location of one person in terms of the viewpoint of the other. Given each location bin, histogram ψ_p is built by counting number of occurrence of the bin number pair to encode the spatial relationship between two targets within a temporal window $(t - \triangle t,\ t + \triangle t)$.

• ψ_a models co-occurrence statistics of atomic actions of the two targets within a temporal window $(t - \triangle t,\ t + \triangle t)$. It is represented as a $|\mathcal{A}| \times (|\mathcal{A}| + 1)/2$ dimensional vector of $(a_i(t), a_j(t))$ histogram.

Note that the first two features ψ_l, ψ_p are dependent on the trajectories of the two targets. Thus, change in association will result in a higher or lower value of an interaction potential.

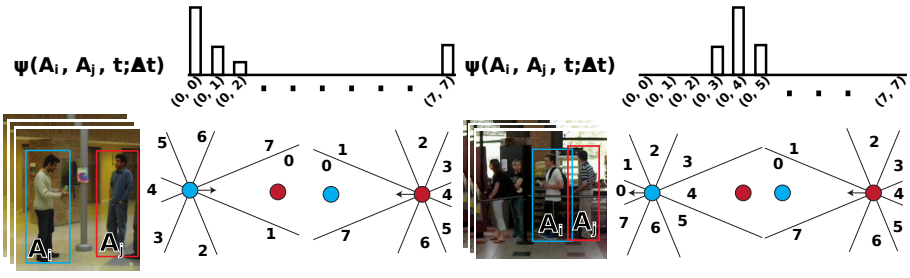

Fig. 1: Illustration of target centric coordinate and histogram ψ_p. **Left-bottom** and **Right-bottom** illustrate typical example of *facing-each-other* and *approaching* interaction. Given the location (circle) and pose (arrow) of target A_i and A_j, each one's location in terms of the other's view point is obtained as a discretized angle (numbers on the figure). The histograms ϕ_p of each example (**top**) are built by counting number of co-occuring discretized angle in a temporal window.

2 Tracklet Association Details

2.1 Hypothesis Generations

For any pair of tracklets τ_i, τ_j that are not co-present at the same time-stamp (thus can be linked), we generate K path hypotheses to associate the two tracklets into a unique track. Such hypotheses are obtained by finding K-shortest paths between the two tracklets in a detection graph (Fig.2). The graph is built by connecting the residual detections between the two tracklets.

To illustrate, consider the example shown in Fig.2. Beginning from the last frame (shown as $t-1$) of preceding tracklet τ_i, we find the residual detections at t that have sufficient amount of overlap with the bounding box of τ_i at $t-1$. We add these detections as a pair of nodes (shown as square nodes in Fig.2) and a cost edge (link the two nodes) into the graph. These nodes are linked to the previous frame's tracklet by a directed edge. Subsequently, we add detections in time stamp $t+1$, by calculating the overlap between the added detection in time t and all residual detections in time $t+1$. We add detection nodes in all time stamps between τ_i and τ_j iteratively and finish the graph building process by considering the connectivity between τ_j and detections at $t+2$. The detections in $t+2$ that do not overlap sufficiently with the bounding box of τ_j at the first frame are discarded.

As noted in the graph, there are exponential (and redundant) number of possible paths that link the two tracklets, which require extensive amount of computation. Especially, if we consider to take the interaction potential into account for tracklet association, it is required to compute an interaction feature for each possible path of target. This can result in infeasible amount of computation in target association. To avoid this issue, we use K-shortest path search method [1] that generate a concise set of path hypothesis to link the two tracklets (Fig.2). In practice, we consider the detection confidence to obtain the cost for simplicity. One can add more cost features such as color similarity, motion smoothness, if desired. To avoid having no proposal when there are missing

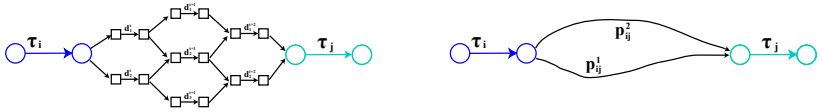

Fig. 2: Illustration of path hypothesis generation given detection residuals. **Left:** the graph is composed of detections in the temporal gap between τ_i and τ_j. Each detection is represent as a pair of square nodes that are linked by a detection response edge. The cost d associated with the edge encodes the detection confidence value. The detections in time $t+1$ that has enough overlap with the detections in time t are added to the graph. **Right:** given the detection residual graph above, we can obtain a concise set of path proposals using K-shortest path search method. Note that there can be exponential number of possible path in the first graph.

detections, we add one default hypothesis that link two tracklets in a shortest distance.

2.2 Match Features

As discussed in the paper, each path p_{ij}^k is associated to a cost value c_{ij}^k that measures the likelihood that the two tracklets τ_i, τ_j belong to the same target. We model this cost value as a linear weighted some of multiple match features: color difference, height difference, motion difference and accumulated detection confidences of the path.

$$c_{ij}^k = w_m^T d_k(\tau_i, \tau_j) \tag{1}$$

where w_m is a model weight and $d_k(\tau_i, \tau_j)$ represent the vector representation of all the features. Each of the features is obtained by following: i) color difference is obtained by the Bhattacharyya distance between color histograms of τ_i and τ_j, ii) height difference is encoded by computing the difference between average height of τ_i and τ_j, iii) motion difference is computed by absolute difference in the velocity of τ_i and τ_j, and iv) accumulated detector confidence is calculated by summing up the detection confidence in the path p_{ij}^k.

Given the match features, we obtain the cost of each path proposal by Eq.1. In the case of target initiation and termination, we use the cost value c_{en}, c_{ex} to model the cost of initiating and terminating a target.

3 Branch-and-Bound Method for Tracklet Association with Interaction Potential

The target association problem with the interaction potential can be written as:

$$\begin{aligned} \hat{f} = \underset{f}{\operatorname{argmin}}\ & c^T f - \Psi(I, A, T(f)) \\ s.t.\ & f_{en,i}, f_{i,ex}, f_{ij}^k \in \{0,1\} \\ & f_{en,i} + \sum_j \sum_k f_{ji}^k = f_{i,ex} + \sum_j \sum_k f_{ij}^k = 1 \end{aligned} \tag{2}$$

where the constraints are summarized as: 1) binary flow constraints (the flow variable should be 0 or 1 integer value specifying that a path is valid or not) and

2) inflow-outflow constraints (the amount of flow coming into a tracklet should be the same as the amount of flow going out of it and the amount is either 0 or 1). The c vector is a cost vector that measures the likelihood of linking two tracklets c_{ij}^k or the cost to initiate/terminate a target c_{en}, c_{ex} and the second term encodes interaction potential which is dependent on the trajectories derived from tracklet association.

3.1 The Non-Convex Quadratic Objective Function

Though the match likelihood is represented as a linear function, the interaction potential involves quadratic relationship between flow variables. As discussed in the paper, the interaction potential $\Psi(I, A, T(f))$ is composed of a sum of interaction potentials each of which is associated to a single interaction variable.

$$\Psi(I, A, T) = \sum_{i,j} \Psi(A_i, A_j, I_{ij}, T) \tag{3}$$

$$\Psi(A_i, A_j, I_{ij}, T) = \sum_{t \in \mathcal{T}_V} \sum_{a \in \mathcal{I}} w_{ai}^a \cdot \psi(A_i, A_j, T, t; \triangle t)\ \mathbb{I}(a, I_{ij}) \tag{4}$$

Since the feature function ψ is dependent on, at most two, flow variables, the overall objective function can be represented as a quadratic function.

Before moving into detailed description, we define the *head* and *tail* path of a tracklet τ_i as the path through which the flow comes into τ_i and the path through which the flow goes out from τ_i, respectively. The *head* path of τ_i can be among the entering path $f_{en,i}$ and the path connecting from any other tracklet τ_l, f_{li}^k. Similarly, the *tail* path of τ_i can be among the exiting path $f_{ex,i}$ and the path connecting to any other tracklet τ_m, f_{im}^k. A tracklet τ_i is called *intact* in a certain temporal support $t \in (t_1,\ t_2)$, if the trajectory of the target is fully covered by the tracklet within the temporal support (i.e, the tracklet is not fragmentized within the time gap). Otherwise, it is called *fragmentized* in a certain temporal support $t \in (t_1,\ t_2)$.

In order to calculate the interaction between two targets i and j at certain time stamp t, we need to specify the trajectory of A_i and A_j in all time stamps $t \in (t - \triangle t,\ t + \triangle t)$ (the temporal support of an interaction, Sec.1), which can involve selecting at most two flow variables in our flow network.[1] If the both tracklets are *intact* within the temporal support of I_{ij}^t, the interaction potential does not get affected by tracklet association (we need to specify no flow variable to compute the interaction feature and thus it can be ignored). If only one of the tracklets is *fragmentized* and the other is *intact*, we need to specify only one *head* or *tail* path of the fragmentized tracklet. On the other hand, if the both

[1] To be complete, it can involve upto four selections of path proposal to fully specify the trajectories of A_i and A_j: *head* of A_i, *tail* of A_i, *head* of A_j and *tail* of A_j if the two tracklets are both fragmentized in both direction within the temporal support of an interaction. However, we ignore such cases since i) it rarely happens, ii) it make the algorithm to be over-complicated and iii) if the tracklets are too short there are not reliable information we can exploit.

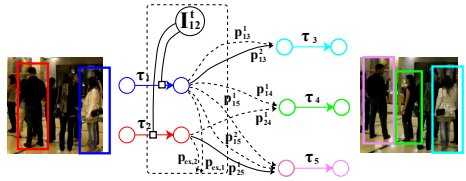

Fig. 3: Consider the case shown in the figure. In order to compute the interaction potential associated with I_{ij}^t, we need to specify the *tail* paths of both tracklet τ_i and τ_j since they are fragmentized in the temporal support of I_{ij}^t (shown as a dotted box).

τ_i and τ_j are fragmentized in the temporal support, we need to specify two flow variables to obtain the associated interaction feature (*head* or *tail* of τ_i and *head* or *tail* of τ_j) (see Fig.3 for more details).

Since the objective function can be specified as a sum of quadratic and linear functions of flow variable f, the problem can be re-written as follows:

$$\begin{aligned} \hat{f} &= \underset{f}{\operatorname{argmin}}\ c^T f - \Psi(I, A, T(f)) \\ &= \underset{f}{\operatorname{argmin}}\ c^T f + c_I^T f + f^T H_I f \\ &\quad s.t.\ f \in \mathbb{S} \end{aligned} \tag{5}$$

$\mathbb{S}$ represent the feasible set for f that satisfies the constraints discussed in previous section, the linear part of interaction potential c_I can be obtained by accumulating the interaction potentials that involve only one selection of path (one of the two tracklets τ_i, τ_j is *intact* within the temporal support), and H_I can be obtained by accumulating all interaction potentials that involve two selections of flow variables (both of τ_i, τ_j are fragmentized in the temporal support of the given interaction variable as in the example of Fig.3). Note that H_I is not positive semi-definite (thus non-convex) and standard quadratic programming techniques are not applicable.

3.2 Branch-and-Bound

Since the objective function is non-convex, we employ a novel Branch-and-Bound algorithm to solve the complicated tracklet association problem. The Branch-and-Bound (BB) algorithm we describe here find the global minimum of the objective function over the space $\mathbb{S}$. Starting from the initial subproblem $\mathcal{Q} = \mathbb{S}$, we split the space into two subspaces $\mathcal{Q}_0, \mathcal{Q}_1$ by setting 0 and 1 to a certain flow variable f_i (ignoring/selecting a path). Given each subproblem (where some of flow variables are already set either 0 or 1), we find the lower bound and upper bound (of optimal solution) in the subproblem, $L(\mathcal{Q})$ and $U(\mathcal{Q})$. If the difference between L and U is smaller than a specified precision ϵ and $U(\mathbb{S})$ is smaller than the lower bound of any other subspace, we stop the iteration and yield the global solution. Otherwise, the algorithm iterate the steps of 1) selecting a subproblem, 2) splitting the subproblem, and 3) finding the lower and upper bound in the subproblem. This is summarized in Algorithm.1.

Algorithm 1 Branch and Bound (BB) Tracklet Association

$\mathcal{Q} = \mathbb{S}$
$L_0 = L(\mathcal{Q})$
$U_0 = U(\mathcal{Q})$
$\mathcal{L} = \{\mathcal{Q}\}$
while $U_k - L_k > \epsilon$, k++ < maxIter **do**
  Select a subproblem $\mathcal{Q} \in \mathcal{L}_k$ for which $L(\mathcal{Q}) = L_k$.
  Split $\mathcal{Q}$ into $\mathcal{Q}_0$ and $\mathcal{Q}_1$
  Form $\mathcal{L}_{k+1}$ from $\mathcal{L}_k$ by removing $\mathcal{Q}$ and adding $\mathcal{Q}_0$ and $\mathcal{Q}_1$
  $L_{k+1} = \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} L(\mathcal{Q})$
  $U_{k+1} = \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} U(\mathcal{Q})$
end while

In following sections, we discuss about how we compute the lower and upper bound of a subproblem $\mathcal{Q}$ (Sec.3.3) and which variable is to be split to provide subproblems $\mathcal{Q}_0$ and $\mathcal{Q}_1$ (Sec.3.4).

3.3 Lower Bound

In this section, we discuss about the lower bound function that we optimize over in each iteration of our BB algorithm. To make it efficient to solve, we find a linear lower bound function:

$$L(f) = (c + c_I + l)^T f \leq (c + c_I)^T f + f^T H_I f, f \in \mathcal{Q} \tag{6}$$

Since the whole interaction potential is represented as a sum of interaction potentials associated with a single interaction variable, it suffices to show that the $l^T f$ is less than or equal to $f^T H f$ within one interaction potential (associated to a single interaction variable I^k_{ij}). Thus, we decompose the whole Hessian H into summation of H_i and show that there exists l_i which is a linear vector that yields a lower bound of $f^T H_i f$, where i denotes an index that enumerates all interaction variables I^k_{ij}. It is trivial to show that $l^T f \leq f^T H f$, if $l_i^T f \leq f^T H_i f, \forall i$ where $l = \sum_i l_i$ and $H = \sum_i H_i$. The matrix H_i can be obtained by computing the corresponding interaction potential $\Psi(A_i, A_j, I^t_{ij}, T(f))$ given each possible configuration of path flows, e.g. selecting the two solid paths shown in the Fig.3.

$$H_i(a, b) = -\frac{1}{2}\Psi(A_i, A_j, I^t_{ij}, T(f)) \; where \; f_a = f_b = 1 \tag{7}$$

To obtain the lower bound of $f^T H_i f$, we note on the two characteristics of our problem: i) the variables are binary and ii) there must be one and only one inflow and outflow for each tracklet τ_i. These two facts can be easily derived from the basic constraints of the problem ($\mathbb{S}$). Given these, we notice that always two elements in H_i are selected with symmetry (shown as red box in Fig.4) and the values are added to produce $f^T H_i f = H_i(a, b) + H_i(b, a)$ where a and b are the indices of the selected variables in f. Thus, it is easy to show that,

$$\min_k H_i(a, k) + \min_k H_i(b, k) \leq H_i(a, b) + H_i(b, a) \tag{8}$$

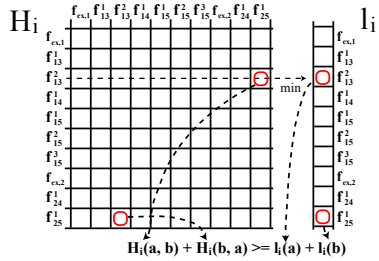

Fig. 4: Illustration of lower bound L computation for the interaction variable corresponding to Fig.3. Each element of the Hessian H_i is obtained by computing the corresponding interaction potential $\Psi(A_i, A_j, I_{ij}^t, T)$ given the flow configuration. A linear lower bound $l^T f$ is derived from $f^T H f$ by taking the minimum of each row in the hessian H matrix. Note that only one configuration can be selected in the matrix H with symmetry since no two flow coming out from one tracklet τ_i or τ_j can be set simultaneously. The example shows the case when solid edges in Fig.3 are selected.

From this, we obtain the lower bound vector l_i for H_i as

$$l_i(a) = \min_k H_i(a, k) \tag{9}$$

see Fig.4 for illustration. The overall lower bound function is obtained by summing up all lower bounds associated to each interaction variable. $l = \sum_i l_i$.

Given the lower bound vector l, the lower bound of $\mathcal{Q}$ is obtained by applying binary integer programming on the lower bound with the given constraints of $\mathcal{Q}$, $\bar{f} = \text{argmin}_f (c + c_I + l)^T f,\ s.t.\ f \in \mathcal{Q}$. The upper bound is set to be infinite if there is no feasible solution, or set to be the value of original objective function if the solution $\bar{f}$ we obtained is feasible.

3.4 Split Variable Selection

Though the presented lower bound can generate quite tight lower bound in our problem, not all the variables in f have the same "tightness". Setting some variable one or zero will have more uncertainties in the difference between the lower bound and actual objective function, and some will generate smaller differences. To efficiently split the space and find the solution, we choose the variable to be selected based on the selecting 'most ambiguous' variable first strategy.

In order to measure the ambiguity, we derive upper bound vector u_i from H_i by

$$u_i(a) = \max_k H_i(a, k) \tag{10}$$

Notice that we take the maximum of a given row in contrast to the minimum in lower bound case (Eq.9). Similar to the lower bound vector case, we can obtain full upper bound vector u by accumulating over different interaction variables. It is trivial to show that :

$$l^T f \leq f^T H f \leq u^T f \tag{11}$$

Notice that if the value of $l(a)$ is the same as $u(a)$, the value added up in the final objective function by selecting a flow variable a does not make any

difference among the above three functions (less ambiguous). However, if the difference between $l(a)$ and $u(a)$ is large, it means that the variable is more ambiguous. Therefore, we choose the variable to be split by finding the variable that has largest difference, $\text{argmax}_a\, u(a) - l(a)$.

References

1. Yen, J.Y.: Finding the k shortest loopless paths in a network. Management Science (1971)