High-order Framewise Smoothness-constrained Globally-optimal Tracking

Norimichi Ukita†[1] and Asami Okada†

†*Graduate School of Information Science, Nara Institute of Science and Technology*

Abstract

This paper proposes smoothness-constrained globally-optimal tracking of objects. Unlike previous globally-optimal tracking methods, the proposed method can evaluate the smoothness of object trajectories so that the smoothness at a frame can (1) be computed from object detections between the frame and any other frame (e.g., detections even between a long time-interval) and (2) be evaluated simultaneously with multiple neighboring frames. The former and latter properties are called high-order smoothness and high-order constraints, respectively. The high-order smoothness allows us to evaluate the smoothness robustly against noise in detections. The high-order constraints are useful because tracking optimization gets more stable as constraints increase in number. Moreover, smoothness can be evaluated between detections at each frame (i.e., framewise optimization) rather than between short tracks, each of which consists of detections at subsequent frames. Globally-optimal tracking is achieved on a graph where each node corresponds to a detected object region so that detected regions are temporally connected through frames in order to find optimal tracking paths. It is difficult for previous globally-optimal methods to use dynamic features (e.g., velocity), which are required to evaluate smoothness between frames. This is because different tracking paths passing through a node require this node to have inconsistent dynamic features. That is, since the different tracking paths in a graph correspond to spatially-different paths in a scene, each of the different tracking paths has its own dynamic feature at this intersection node. These inconsistent features at the intersection node are prohibited for global optimization on a graph. In the proposed method, such a node is divided such that each divided node has a unique dynamic feature. With public datasets, the proposed method can improve the rate of successful tracking.

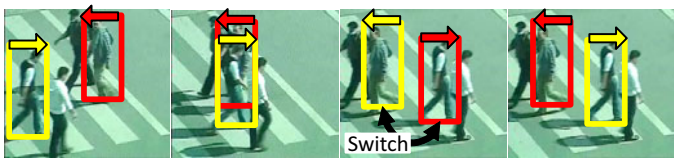
Keywords: Multi-object Tracking, Data association, Motion smoothness, High-order constraints

1. Introduction

Globally-optimal tracking [30, 7, 43, 24, 35, 6] tracks all objects by connecting their detected regions through frames. This tracking can acquire the mutually-consistent trajectories of the objects by minimum-cost path search on a graph and can work quickly using integer linear programming (ILP) [41, 42, 19, 4, 26, 29, 8].

For connecting the object regions consistently, a cost for this optimal path search is designed so that the attributes of connected object regions are consistent. For example, the consistencies of positions are evaluated in most conventional globally-optimal tracking methods; closer regions are connected between consecutive frames. As well as this position closeness, motion smoothness is also a critical cue for connecting the object regions consistently. In particular, motion smoothness is indispensable for tracking objects with similar appearance [15] such as people observed by a distant camera used for surveillance. With no smoothness, people passing each other, such as those shown in Fig. 1, are often miss-tracked (the third image from the left in Fig. 1).

Such miss-tracking occurs in tracking using ILP because ILP evaluates the consistency of object regions only between two consecutive frames. In other words, a graph in ILP has links only between a pair of nodes, each of which corresponds to an object region, in two consecutive frames. We call this limitation the **first-order link**. This limitation is required for efficient optimization by the dynamic programming or similar techniques.



$t-2$ $t-1$ t t

People passing each other | No motion smoothness | With motion smoothness

Figure 1: Effects of evaluating motion smoothness. The color of each rectangle indicates the tracking ID of each person. While a previous method may fail tracking people passing each other, they can be tracked successfully by evaluating motion smoothness.

The disadvantage of the first-order link limitation is that it disables evaluation of motion smoothness but allows us to employ only position smoothness between nodes. This is because a cost between two nodes is defined with only their features but each node has only the static features (e.g., position and color) of its object region while dynamic features (e.g., velocity) are required in each node for evaluating motion smoothness (abbreviated to smoothness in what follows).

The first-order link limitation also disables evaluating similarity (e.g., closeness and smoothness) between a node and other two or more nodes in a tracking path. For example, for

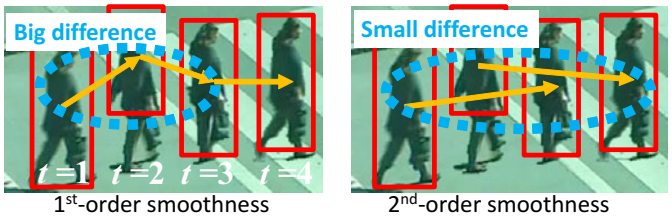


Figure 2: Advantage of high-order smoothness. The first-order smoothness (i.e., directed velocities computed by connecting detected regions in two consecutive frames, which are indicated by arrows in the left-hand figure) is noisy, as marked by "Big difference", due to noisy detected regions. On the other hand, the variance of the high-order smoothness (e.g., second-order smoothness in the right-hand figure) is smaller and useful for cues in tracking.

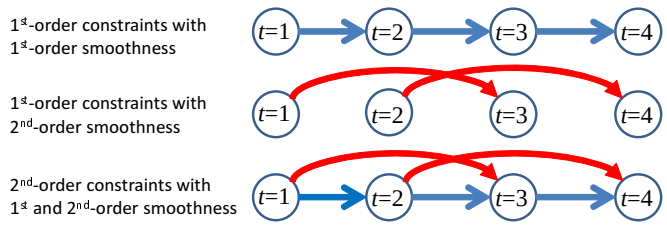


Figure 3: (Upper row) first-order smoothness and (Middle and lower rows) simple but inefficient/intractable high-order constraints. Each link (i.e., arrow) has a cost of dissimilarity between dynamic features in two nodes connected by the link. The second-order smoothness, which is indicated by red links, is evaluated between t-th and $(t+2)$-th frames (i.e., "t=1 and t=3" and "t=2 and t=4" in the figure). The second-order constraints are evaluated by connecting t-th frame with other two frames (e.g., $(t+1)$-th and $(t+2)$-th frames), as illustrated in the lower example.

evaluating a tracking path from $t = 1$ to $t = 4$ in Fig. 2, if similarity is evaluated between $t = 1$ and $t = 2$ by linking their corresponding nodes, a node at $t = 1$ cannot have any link that connects to nodes in other frames (i.e., $t = 3$ and $t = 4$). However, similarity evaluation among multiple frames may improve robustness in globally-optimal tracking in analogy with other computer vision problems using higher-order markov random field (e.g., high-order graph cut [20, 18]).

The contribution of this paper is the extension of a graph for smoothness-constrained globally-optimal tracking. To employ constrains on smoothness in a more flexible way rather than limited first-order smoothness [32, 38, 40, 10, 11], the proposed method is possessed of the following properties while maintaining the first-order link limitation in a graph:

- **High-order smoothness** is evaluated by a dynamic feature computed from a pair of nodes in any frames. In this paper, a dynamic feature computed from nodes in t-th and $(t+k)$-th frames is called a k-th-order dynamic feature of t-th frame; $k \in \mathbb{N}$ is an arbitrary positive integer. Smoothness evaluated by k-th-order dynamic features is called k-th-order smoothness. Examples of first- and second-order directed velocities (i.e. dynamic features) are indicated by arrows in Fig. 2. Under this definition, position closeness employed in most globally-optimal tracking methods is defined as 0-th order smoothness.

  High-order smoothness is useful rather than first-order smoothness, while first-order smoothness is noisy. As can be seen in the example shown in Fig. 2, noisy directed velocities in first-order smoothness can be stabilized by second-order smoothness. Tracking paths are optimized by employing constraints on smoothness so that the dynamic features of detections are smooth along each tracking path.

- **High-order constrains** are evaluated such that any number of constraints on smoothness in a node are evaluated along a path simultaneously. Multiple constraints in a node are evaluated by linking this node to nodes in multiple frames; see the third row in Fig. 3, for example. With high-order constraints, minimum-cost path search is constrained by smoothness from a frame of interest, time t, to multiple frames. That is, a set of k_1-th, $\cdots$, k_l-th-order smoothness is evaluated simultaneously to optimize each tracking path. $k_1, \cdots, k_l$ are l arbitrary different positive integers, where $k_p \neq k_q$ for any pair of p and q. In this paper, such constraints with l dynamic features are called l-th-order constraints.

  The difference between high-order smoothness and constraints is illustrated in graphs in Fig. 3. First-order and second-order smoothness are indicated by blue and red links, respectively. Since each node in the upper two graphs has only one constraint (i.e., one link), they are constrained only by first-order constraints. On the other hand, each node in the bottom graph with second-order constraints has two links.

- **Framewise smoothness** is evaluated with no tracklets (i.e., short tracks) of objects. That is, each node corresponds to an object region detected at each frame, while each tracklet corresponds to a node in tracklet-based methods. Since tracklets are obtained by conventional tracking methods, miss-tracking is inevitable in the tracklets even if they are obtained within only frames where no occlusion between detected objects is occurred. Framewise optimization allows us to avoid negative effects due to such a preprocess (i.e., inevitable miss-tracking in tracklets).

2. Related Work

While recent globally-optimal tracking methods using ILP [42, 19, 4, 26, 29] enable efficient optimization compared with earlier work [17], their computational complexity, i.e., $O(T^3 \log^2 T)$, strongly depends on the number of frames (denoted by T). Thus it is difficult to use such methods for a long sequence. Pirsiavash et al. [31] successfully reduced this computational complexity to $O(KT)$ where K denotes the number of objects. This method allows us to achieve globally-optimal tracking in long sequences.

However, in most globally-optimal tracking methods, nodes are connected only between two consecutive frames. Due to this first-order link limitation, only static cues (e.g., object position and scale) extracted from each frame are used to evaluate the connectivity between nodes.

A few globally-optimal tracking methods [32, 38, 40, 10, 11] introduce constraints on smoothness under the first-order link limitation. In [32, 38, 40, 10], smoothly-connected regions (i.e., a tracklet obtained by pre-processing) constitute a node in a graph. While dynamic features can be computed in each node and motion similarity can be evaluated between two consecutive nodes, miss-tracking in tracklets is inevitable. With no tracklets, Butt and Collins [11] evaluate the first-order smoothness between two consecutive frames. This is achieved by (1) merging any pair of regions detected in two consecutive frames into a node and (2) connecting such nodes by hyperlinks representing constraints that must be enforced so that each region is used only once. Since a graph with these hyperlinks requires complex optimization, approximate and computationally-huge optimization is used. While the first-order smoothness in a graph is used also in [14] with constant-velocity terms in a cost function, this method also requires an approximate solution method based on the iterated conditional modes algorithm, which requires significant time for convergence. In addition, the first-order smoothness [11, 14] is not so reliable due to noisy detections, as shown in Fig. 2.

In summary, the proposed method can apply high-order smoothness and constraints to a graph while maintaining the first-order link limitation. In addition, constraints on smoothness can be evaluated framewise such that each node corresponds to an object region rather than a tracklet.

In terms of similarity evaluation between detections, not only spatial features such as motion smoothness and position closeness but also appearance features are crucial for visual tracking. Globally-optimal tracking can employ any kinds of features independently by adding the similarity score of each feature to a link between the nodes corresponding to the detections. On the other hand, fusion of the features at the level of visual information can improve similarity evaluation. While the effectiveness of the visual-level fusion has been demonstrated in eariler work (e.g., for multi-target tracking [13] and single-object tracking [12, 21]), our implementation of the proposed method evaluates all features independently so that this work focus on how to extend a graph for smoothness-constrained globally-optimal tracking; Sec. 4, for more details.

3. Graph with Constraints on Smoothness

An original graph used for globally-optimal tracking is illustrated in Fig. 4; see its caption for details. Each inter-frame link has several costs relating to similarity between a pair of nodes connected by the link. For example, a cost relating to position closeness (i.e., cost of 0-th order smoothness) is given to each inter-frame link in previous globally-optimal methods as well as in our proposed method.

A cost function evaluating a set of tracking paths $\boldsymbol{T}$ (denoted by $C(\boldsymbol{T})$) is as follows[2]:

$$C(\boldsymbol{T}) = \sum_i C_i^s f_i^s + \sum_i C_i^t f_i^t + \sum_i C_i f_i + \sum_{i,j} C_{i,j} f_{i,j}, \quad (1)$$

[2]For more details of this cost function, refer to [42, 31].

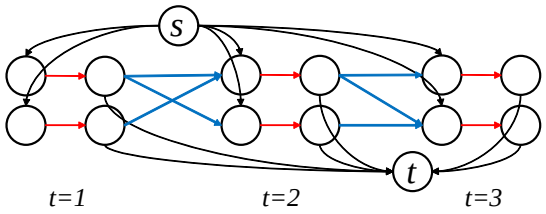


Figure 4: Example of a graph with first-order links. The graph consists of nodes, intra-frame links (indicated by red arrows), and inter-frame links (indicated by blue arrows). A pair of nodes connected via an intra-frame link corresponds to a detected object region. The intra-frame link has a cost of the detection of each object. An inter-frame link connects a pair of nodes between two consecutive frames. Each inter-frame link has several costs representing dissimilarity between two object regions. In this graph, paths with a lower cost are searched for and considered to be object paths (i.e., tracking results). For simplicity, two nodes connected by an intra-frame link are merged for visualization in the subsequent figures.

subject to the following equations:

$$f_i^s + \sum_j f_{j,i} = f_i = f_i^t + \sum_j f_{i,j}, \quad (2)$$

where C_i^s, C_i^t, C_i, and $C_{i,j}$ denote costs given to a link from s-node to i-th node, a link from i-th node to t-node, an intra-frame link of i-th node, and an inter-frame link between i-th and j-th nodes, respectively. f_i^s, f_i^t, f_i, and $f_{i,j}$ denote 0-1 indicator variables defined as follows:

- f_i^s is 1 if $\exists T_t \in \boldsymbol{T}$ where path T_t starts from i-th node. Otherwise, f_i^s is 0.
- f_i^t is 1 if $\exists T_t \in \boldsymbol{T}$ where path T_t ends at i-th node. Otherwise, f_i^t is 0.
- f_i is 1 if $\exists T_t \in \boldsymbol{T}$ where path T_t includes i-th node. Otherwise, f_i is 0.
- $f_{i,j}$ is 1 if $\exists T_t \in \boldsymbol{T}$ where path T_t includes a link between i-th and j-th nodes. Otherwise, $f_{i,j}$ is 0.

If and only if the constraint expressed by Eq. 2 is satisfied, any pair of paths in $\boldsymbol{T}$ has no overlap.

Globally-optimal tracking methods by minimum-cost path search on a graph, including our proposed method, produce tracking paths by connecting object detections. Since such globally-optimal tracking methods cannot track false-negative detections, these false-negatives should be reduced by a loose threshold in detection stage. While the loose threshold produces a number of false-positive detections, the globally-optimal tracking methods extract true-positive tracking paths by optimizing the cost function, Eq. (1), rather than assume all detections are true. Additional schemes for explicitly expressing long-term occlusions in optimization and for removing false-positive trajectories after optimization, as described in [42].

3.1. Smoothness Cost

In such a graph, k-th-order smoothness is evaluated with a k-th-order dynamic feature computed between nodes in t-th and

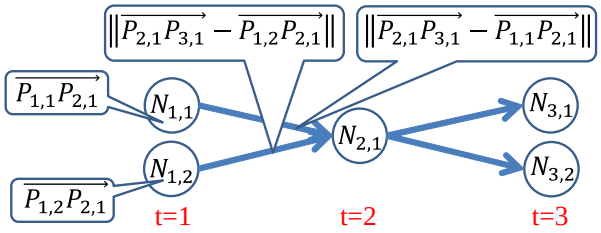


Figure 5: Smoothness cost. Each node corresponds to a detected object region. Balloons with a node and a link show the directed velocity of this node and the cost given to this link, respectively. The costs shown in this figure are those of tracking paths toward $N_{3,1}$.

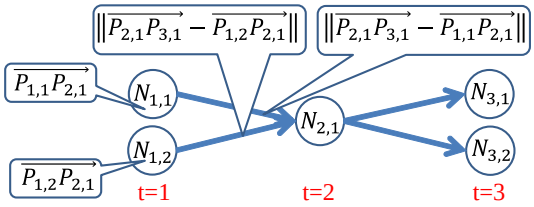


Figure 6: Node division for maintaining a unique dynamic feature in each sub-node. In this example, $N_{2,1}$ in Fig. 5 is divided into $N^1_{1,2}$ and $N^2_{1,2}$ that respectively have $\overrightarrow{\boldsymbol{P}_{2,1}\boldsymbol{P}_{3,1}}$ and $\overrightarrow{\boldsymbol{P}_{2,1}\boldsymbol{P}_{3,2}}$ as a dynamic feature. Sub-nodes divided from a node are enclosed by an orange rectangle.

$(t+k)$-th ($k \in \mathbb{N}$) frames. In this paper, this dynamic feature is expressed by a directed velocity in each node. Let $N_{t,i}$, $\boldsymbol{P}_{t,i}$, and $\boldsymbol{v}^{(k)}_{t,i}$ denote i-th node in t-th frame, the position of an object that corresponds to $N_{t,i}$, and its k-th-order directed velocity, respectively. Note that object IDs, i, are given for each frame independently prior to tracking; $N_{t,i}$ and $N_{t+1,i}$ may correspond to different objects. With these notations, $\boldsymbol{v}^{(k)}_{t,i}$ can be approximately computed as follows:

$$\boldsymbol{v}^{(k)}_{t,i} = \frac{\overrightarrow{\boldsymbol{P}_{t,i}\boldsymbol{P}_{t+k,m}}}{k} = \frac{\boldsymbol{P}_{t+k,m} - \boldsymbol{P}_{t,i}}{k}, \tag{3}$$

where $m \in \mathbb{N}$ denotes the ID of $(t+k)$-th frame's node connecting to $N_{t,i}$.

In this paper, the cost of smoothness between nodes in t-th and $(t+l)$-th frames is given by subtraction between $\boldsymbol{v}^{(k)}_{t,i}$ and $\boldsymbol{v}^{(k)}_{t+l,j}$ defined in Eq. 3:

$$c(k,t,i,t+l,j) \quad = \quad \|\boldsymbol{v}^{(k)}_{t,i} - \boldsymbol{v}^{(k)}_{t+l,j}\| \tag{4}$$

Figure 5 shows the simple examples of directed velocities and costs with $k = 1$ and $l = 1$.

3.2. Node Division for Constraints on Smoothness

While the cost of smoothness between nodes in any pair of frames is defined by Eq. 4, $\boldsymbol{v}^{(k)}_{t,i}$ is changed depending on $\boldsymbol{P}_{t+k,m}$ (i.e., depending on which node in $(t+k)$-th frame is connected to $N_{t,i}$). For example, at 1st frame of Fig. 5, the first-order dynamic features of $N_{1,1}$ and $N_{1,2}$ (i.e., $\overrightarrow{P_{1,1}P_{2,1}}$ and $\overrightarrow{P_{1,2}P_{2,1}}$) are fixed. At $t = 2$, on the other hand, $N_{2,1}$ has $\overrightarrow{P_{2,1}P_{3,1}}$ or $\overrightarrow{P_{2,1}P_{3,2}}$ as its dynamic feature. This variability prevents us from attributing dynamic features to a graph for minimum-cost path search.

In the proposed method, this variability of the dynamic features is removed by dividing nodes having variable features into sub-nodes each of which has only unique features, as illustrated in Fig. 6. While the following sections, Secs. 3.3.1, 3.3.2, and 3.3.3, describe graph building with different orders of smoothness and constrains, we define the following principle in node division that is independent of order:

Node-division principle: A node, including a sub-node, must have unique dynamic features. This uniqueness means a one-to-one correspondence between a pair of nodes in t-th and $(t+k)$-th frames ($k \in \mathbb{N}$) for k-th-order smoothness; if $N_{t,i}$ connects to $N_{t+k,j}$, there is no path from $N_{t,i}$ to any nodes in $(t+k)$-th frame but the path to $N_{t+k,j}$. Thus, $N_{t,i}$ has a unique dynamic feature computed with $N_{t+k,j}$. If this requirement is violated in a node, it is divided into sub-nodes.

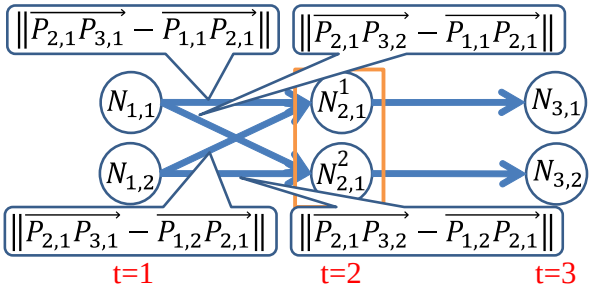


Figure 7: Costs of first-order smoothness, which are obtained from three consecutive frames (i.e., $t = 1$, $t = 2$ and $t = 3$), are computed by Eq. (4) and given to links between $t = 1$ and $t = 2$.

3.3. Graph Building with Constraints on Smoothness

The easiest problem with only first-order constraints by first-order smoothness is introduced in Sec. 3.3.1. Then problems with high-order smoothness and high-order constraints are described in Secs. 3.3.2 and 3.3.3, respectively.

3.3.1. Graph with First-order Constrains by First-order Smoothness

A general graph (e.g., Fig. 4) in which object regions and nodes have a one-to-one correspondence is extended as follows:

Step1: A directed velocity computed from object positions in t-th and $(t+1)$-th frames is regarded as the first-order dynamic feature. This directed velocity is given to a node in t-th frame, as illustrated by $\overrightarrow{\boldsymbol{P}_{1,1}\boldsymbol{P}_{2,1}}$ and $\overrightarrow{\boldsymbol{P}_{1,2}\boldsymbol{P}_{2,1}}$ in Fig. 5.

Step2: If $N_{t,i}$ connects to D nodes in $(t+1)$-th frame, $N_{t,i}$ is divided into D sub-nodes, $N^d_{t,i}$, where $d \in \{1, \cdots, D\}$ is a division ID. The features given to each sub-node $N^d_{t,i}$ are the same as those of $N_{t,i}$ with the exception of a directed velocity. $N^d_{t,i}$ connects to all $(t-1)$-th frame's nodes connecting to its original node, $N_{t,i}$. In $(t+1)$-th frame, on the other hand, $N^d_{t,i}$ only connects to a node that is used to

compute a directed velocity given to $N^d_{t,i}$. In the example shown in Fig. 5, $N_{2,1}$ must be divided because it is connected to $N_{3,1}$ and $N_{3,2}$. Directed velocities toward $P_{3,1}$ and $P_{3,2}$ (i.e., $\overrightarrow{P_{2,1}P_{3,1}}$ and $\overrightarrow{P_{2,1}P_{3,2}}$) are given to sub-nodes $N^1_{2,1}$ and $N^2_{2,1}$, respectively, as shown in Fig. 6. Then both of $N^1_{2,1}$ and $N^2_{2,1}$ connect to $N_{1,1}$ and $N_{1,2}$, while $N^1_{2,1}$ and $N^2_{2,1}$ respectively connect to $N_{3,1}$ and $N_{3,2}$.

Step3: After directed velocities are given to nodes and sub-nodes in all frames, the costs defined by Eq. (4) are computed from the directed velocities in all possible pairs of t-th and $(t+1)$-th frames. They are then given to their corresponding links between t-th and $(t+1)$-th frames. In the example shown in Fig. 7, each of the four links between $t = 1$ and $t = 2$ has its own cost.

The costs of the directed velocities between t-th and $(t+1)$-th frames are computed from the object positions in t-th, $(t+1)$-th, and $(t+2)$-th frames. This means that a cost given to a link between t-th and $(t+1)$-th frames must correspond to only one of the possible paths from t-th to $(t+2)$-th frames. In Fig. 7, there are four possible paths from $t = 1$ to $t = 3$, i.e., $N_{1,1} \rightarrow N^1_{2,1} \rightarrow N_{3,1}$, $N_{1,1} \rightarrow N^2_{2,1} \rightarrow N_{3,2}$, $N_{1,2} \rightarrow N^1_{2,1} \rightarrow N_{3,1}$, and $N_{1,2} \rightarrow N^2_{2,1} \rightarrow N_{3,2}$. As can be seen, each of the four costs given between $t = 1$ and $t = 2$ corresponds to one of these four paths.

3.3.2. Graph with High-order Smoothness

The pseudo code of the whole graph building is shown in Fig. 11. The procedures SMOOTHNESS and CONSTRAINTS called in the graph building are described in Secs. 3.3.2 and 3.3.3, respectively.

k-th-order smoothness is evaluated with k-th-order dynamic features computed from object positions in t-th and $(t+k)$-th frames. To give unique k-th-order dynamic features to a graph based on the node-division principle, if $N_{t,i}$ connects to D nodes in $(t+k)$-th frame via frames between them, $N_{t,i}$ must be divided into D sub-nodes. This division makes each of the D sub-nodes having only one of D dynamic features computed from $N_{t,i}$ and D nodes in $(t+k)$-th frame.

The process described above is almost the same as node division for the first-order smoothness described in Sec. 3.3.1. The difference between these two division schemes is that, k-th-order smoothness connects $N^d_{t,i}$ to $N_{t+k,j}$ via nodes between $(t+1)$-th and $(t+k-1)$-th frames. These intermediate nodes must also be divided into D sub-nodes and connected to each other such that $N^d_{t,i}$, which has $\frac{\boldsymbol{P}_{t+k,j}-\boldsymbol{P}_{t,i}}{k}$ as its dynamic feature, connects to only $N_{t+k,j}$. All sub-nodes are connected such that $N^d_{t,i}$ connects only to d-th sub-nodes of each intermediate node toward d-th node in $(t+k)$-th frame. Figure 8 shows the simple example of node division for the k-th-order smoothness; $k = 2$. As described in the caption of this figure, division for k-th-order dynamic features are executed after division for $(k-1)$-th dynamic features.

After node division and dynamic feature computation, the costs of smoothness are given to a graph. As well as the cost of the first-order smoothness between nodes in t-th and $(t+1)$-th

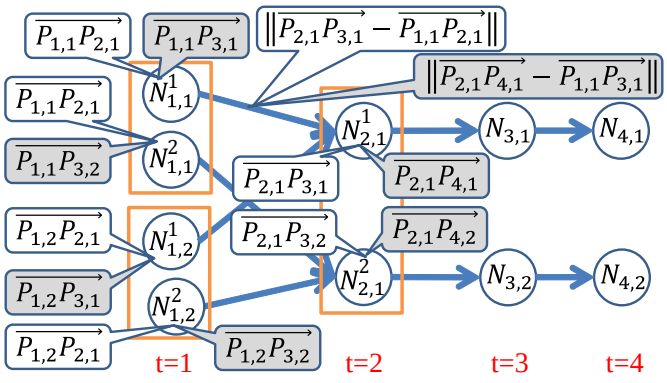


Figure 8: Node division for high-order smoothness and constraints. Division for the second-order smoothness is shown in this figure. This division is executed following the one for the first-order smoothness, which is shown in Fig. 7. Second-order dynamic features and costs using them are indicated by gray balloons. One-to-one correspondence between nodes at $t = 1$ (i.e., $N^1_{1,1}$ and $N^2_{1,1}$) and those at $t = 3$ must be maintained for the node-division principle. To this end, while nodes at $t = 2$ must be also divided, they are already divided properly for the first-order smoothness prior to division for the second-order smoothness.

frames, the ones of high-order smoothness between them are embedded into a link between t-th and $(t+1)$-th frames. In the example in Fig. 8, the costs of the first- and second-order smoothness (i.e., $\|\overrightarrow{\boldsymbol{P}_{2,1}\boldsymbol{P}_{3,1}} - \overrightarrow{\boldsymbol{P}_{1,1}\boldsymbol{P}_{2,1}}\|$ and $\|\overrightarrow{\boldsymbol{P}_{2,1}\boldsymbol{P}_{4,1}} - \overrightarrow{\boldsymbol{P}_{1,1}\boldsymbol{P}_{3,1}}\|$) are embedded into the same link connected to $N^1_{1,1}$. Recall that, in a graph with k-th-order smoothness, $N^d_{t,i}$ having a path to $N_{t+k,j}$ connects only to $N_{t+k,j}$ in $(t+k-1)$-th frame. Under this condition, the costs of k-th-order smoothness can be given to any link along the path from $N^d_{t,i}$ toward $N_{t+k,j}$ because this path is a unique path from $N^d_{t,i}$ toward $N_{t+k,j}$. In the proposed method, these costs are given to a link between t-th and $(t+1)$-th frames as described above.

The pseudo code of this node division for high-order smoothness is shown in the procedure SMOOTHNESS of Fig. 12. The example for interpreting lines 12 and 18 in this procedure is shown in Fig. 10.

In an extended graph built by the proposed method, the number of nodes is increased only by this procedure SMOOTHNESS. Given the number of frames (denoted by T) and the mean number of nodes in each frame (denoted by $\bar{I}$), the number of nodes in an extended graph is increased from $O(TI)$, which is the number of nodes in its original graph, to $O(TI^{k+1})$. Note that $O(TI^{k+1})$ is significantly smaller than $O(TI^{T-1})$ that is the number of nodes in a graph in which all possible paths between the first and last frames are uniquely constructed by dividing nodes Figure 9 shows why the number of nodes is $O(TI^{k+1})$ rather than being increased exponentially, i.e., $O(TI^T)$. In the example shown in Fig. 9, two nodes at $t = 2$ increase twofold for the second-order smoothness because "two" nodes at $t = 1$ require unique paths toward the nodes at $t = 2$. Two nodes at $t = 4$ also increase twofold rather than fourfold even though the number of sub-nodes at $t = 2$ is four. This is because, for example, $N^1_{2,1}$ and $N^2_{2,1}$ have the same position, $\boldsymbol{P}_{2,1}$, and so these

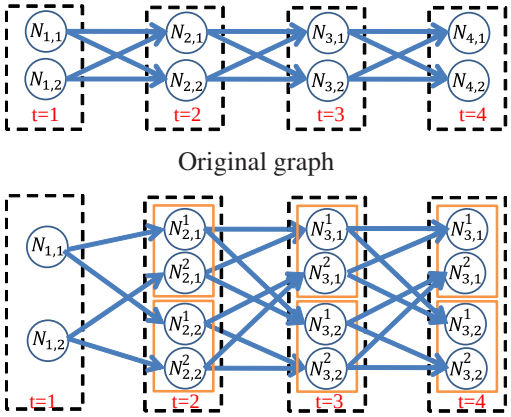


Figure 9: The number of nodes divided for constraints on smoothness. This example is the case of $k = 2$. The upper and lower graphs show an original graph and its extended graph, respectively.

two sub-nodes require only one sub-node for each of two nodes at $t = 3$, i.e., $N^1_{3,1}$ for $N_{3,1}$ and $N^1_{3,2}$ for $N_{3,2}$.

3.3.3. Graph with High-order Constraints

High-order constraints can be implemented by high-order links that connect a node in t-th frame to nodes in multiple frames. Each high-order link has its cost representing the smoothness between the dynamic features of two nodes connected by this link. These high-order links are prohibited by the limitation of the first-order link.

Rather than using these high-order links, we employ a simple alternative in a graph with high-order smoothness that is built by the node division described in Sec. 3.3.2. Recall that the cost of k-th-order smoothness can be given to a link between nodes in t-th and $(t + 1)$-th frames. In the example of the second-order smoothness shown in Fig. 8, its cost is embedded with the cost of the first-order smoothness into the same link. This is the second-order constraints. In a graph with higher-order smoothness (denoted by k-th-order smoothness) also, any k'-th-order smoothness (where $1 \leq k' \leq k$) can be given to a link between t-th and $(t + 1)$-th frames in order to give higher-order constraints to a graph. If all of constraints on $\{1, 2, \cdots, k\}$-th-order smoothness are given to this link, k-th-order constraints are represented.

Note that, if a graph is built for k-th-order smoothness, the constraints of a node in t-th frame cannot have a cost computed with any nodes in $(t + k + 1)$-th frame or more distant frames.

The pseudo code for high-order constrains is shown in the procedure CONSTRAINTS of Fig. 12.

3.4. Multi-object Tracking using Graph with Constraints on Smoothness using Sub-nodes

Unlike the original graph, the extended graph proposed in this paper has sub-nodes corresponding to a single object region. Since these sub-nodes are regarded as different objects in existing globally-optimal tracking methods, these methods may

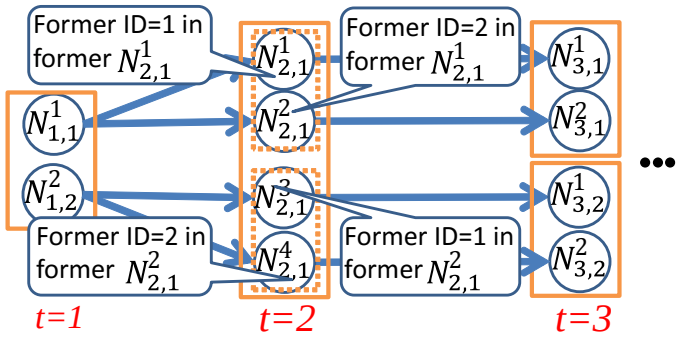


Figure 10: This graph is generated in the loop with $k' = 2$ and $t = 2$ of the procedure SMOOTHNESS in Fig. 12. In this loop, $N^1_{2,1}$ and $N^2_{2,1}$ are generated by dividing the former $N^1_{2,1}$ (denoted by $N'^1_{2,1}$), as well as $N^3_{2,1}$ and $N^4_{2,1}$ by the former $N^2_{2,1}$. Dotted rectangles indicate these former sub-nodes. In line 12 with $i = 1$ and $d = 1$, for example, $L^1_{1,1}$ includes $N^1_{2,1}$ and $N^2_{2,1}$ because their pre-divided former node, $N'^1_{2,1}$, was connected to $N^1_{1,1}$. In line 18 with $i = 1$ and $d = 3$, for example, $L^3_{2,1}$, whose former ID is $d' = 1$, includes $(d' = 1)$-th node of $\{N^1_{3,2}, N^2_{3,2}\}$ (i.e., $N^1_{3,2}$) because $\{N^1_{3,2}, N^2_{3,2}\}$ was connected to $N'^1_{2,1}$ in the former loop.

Input: $\forall t \forall i_t N_{t,i_t}$ ▷ $t \in \{1, \cdots, T\}$ and $i_t \in \{1, \cdots, I_t\}$.

1: **procedure** BUILDGRAPH
2: $\forall t \forall i_t \boldsymbol{L}_{t,i_t}$ = **BuildOriginalGraph**$(\forall t \forall i_t N_{t,i_t})$ ▷ $\boldsymbol{L}_{t,i_t}$ is a set of $(t+1)$-th frame's node IDs to which N_{t,i_t} connects by links.
3: **Smoothness**$(\forall t \forall i_t N_{t,i_t}, \forall t \forall i_t \boldsymbol{L}_{t,i_t})$
4: **Constraints**$(\forall t \forall i_t \forall d N^d_{t,i_t}, \forall t \forall i_t \forall d \boldsymbol{L}^d_{t,i_t})$
5: **end procedure**

Output: $\forall t \forall i_t \forall d N^d_{t,i_t}$, and $\forall t \forall i_t \forall d \boldsymbol{L}^d_{t,i_t}$.

Figure 11: Graph building for l-th-order constraints with k-th-order smoothness. For convenience, each original node, $N_{t,i}$, is replaced by $N^1_{t,i}$ in Figs. 11 and 12. T and I_t denote the number of frames and nodes in t-th frame, respectively. This algorithm is designed to build a fully-connected graph under the first-link limitation, where all nodes in two consecutive frames are connected to each other.

find inconsistently multiple object paths from the sub-nodes of a single object. To avoid such inconsistent tracking paths, our proposed method must detect only one path from the sub-nodes of a single object on the extended graph.

This path tracking is achieved by iterative min-cost path search [31], in which the tracking paths of multiple objects are detected by a greedy, iterative shortest-path algorithm. For globally-optimal tracking of objects by iterative path search, a newly detected path edits paths detected in previous search steps if these paths overlap the new path. This graph editing allows us to revise previously-mistracked object paths; see [31] for details.

Before this graph editing, the extended graph is edited so that the sub-nodes of a single object are not included in multiple tracking paths as follows:

1. Given a path (consisting of a set of sub-nodes, $N^{(S)}$) detected in r-th iteration, let $N^{(O)}$ be a set of original nodes

1: **procedure** SMOOTHNESS($\forall t \forall i_t N^1_{t,i_t}, \forall t \forall i_t \boldsymbol{L}^1_{t,i_t}$)
2: **for** $k' = 1$ to k **do**
3: **for** $t = 1$ to $T - k'$ **do**
4: **for** $i = 1$ to I_t **do**
5: **for** $d = 1$ to $D_{t,i}$ **do**
6: $N^d_{t,i}$ is divided to $I_{t+k'}$ sub-nodes.
7: **end for**
8: **end for**
9: **if** $t \neq 1$ **then**
10: **for** $i = 1$ to I_{t-1} **do**
11: **for** $d = 1$ to $D_{t-1,i}$ **do**
12: $\boldsymbol{L}^d_{t-1,i}$ is renewed to include sub-nodes, generated in line 6, whose pre-divided former node (denoted by M_t) was connected to $N^d_{t-1,i}$.
13: **end for**
14: **end for**
15: **end if**
16: **for** $i = 1$ to I_t **do**
17: **for** $d = 1$ to $D_{t,i}$ **do**
18: $\boldsymbol{L}^d_{t,i}$ is renewed to include only d'-th node/sub-node among nodes/sub-nodes that were together connected to M_t; d' denotes the former ID of $N^d_{t,i}$ among sub-nodes generated by dividing M_t.
19: **end for**
20: **end for**
21: **end for**
22: **end for**
23: **end procedure**

1: **procedure** CONSTRAINTS($\forall t \forall i_t \forall d N^d_{t,i_t}, \forall t \forall i_t \forall d \boldsymbol{L}^d_{t,i_t}$)
2: **for** $k' = 1$ to k **do**
3: **for** $t = 1$ to $T - k'$ **do**
4: **for** $i = 1$ to I_t **do**
5: **for** $d = 1$ to $D_{t,i}$ **do**
6: **for** $l' = 1$ to l **do**
7: Cost (4), $c(k', t, i, t + l', j)$, is given to links from $N^d_{t,i}$ to $\boldsymbol{L}^d_{t,i}$. ▷ j is determined by $k', t, i, t + l'$ because of node-division principle.
8: **end for**
9: **end for**
10: **end for**
11: **end for**
12: **end for**
13: **end procedure**

Figure 12: Procedures used in the graph building shown in Fig. 11. For convenience, each original node, $N_{t,i}$, is replaced by $N^1_{t,i}$ in this figure. $D_{t,i}$ is the number of sub-nodes of $N^0_{t,i}$.

each of which produces one of sub-nodes in $N^{(S)}$. All sub-nodes produced from $N^{(O)}$, exclusive of $N^{(S)}$, are not used after r-th iteration to avoid tracking the same object multiple times. To this end, costs given to the links of these sub-nodes are set to be infinity.

2. In addition, all links of $N^{(S)}$ are edited so that these links have negative cost with the opposite direction, as proposed in [31].

For the aforementioned path search method to find globally-optimal tracking paths, an assumption is that, if a sub-node of $N_{t,i}$ is included in a tracking path detected in r-th iteration, (1) this sub-node must be included in one of optimal paths and (2) other sub-nodes of $N_{t,i}$ must not be included in any optimal paths. This is because other sub-nodes of $N_{t,i}$ are not used in neither min-cost path search nor previously-mistracked path revision after r-th iteration.

The computational cost of the iterative path search using approximate DP [31] is $O(KT)$ where K and T respectively denote the unknown optimal number of objects and all frames. Since K and T are not changed by the proposed node division, its theoretical computational cost is maintained as $O(KT)$.

4. Experiments

We tested the proposed method with the public datasets, PETS2009 [3], ETHMS [16] and CAVIAR [2] datasets.

PETS2009: A sequence S2.L1 with 795 frames was used. 768 × 576 pixels.

ETHMS: 999 and 354 frames (BAHNHOF and SUNNYDAY). 14 fps. 640 × 480 pixels. Only a left-eye videos were used, while the data has stereo sequences.

CAVIAR (2nd set): 20 videos (25587 frames in total). 25 fps. 384 × 288 pixels.

Detections in all frames and the ground-truth of tracking trajectories were obtained from the publicly-available datasets [39, 40]. All datasets are available online [25, 1]. In the datasets, the detections were obtained and represented as bounding boxes automatically by a detector, while the tracking ground-truth was given manually so that the detections were connected temporally.

A constant cost was given to all links connected to s and t nodes (i.e., black-colored links in Fig. 4), as was done in [31]. The costs given to intra- and inter-frame links were determined as follows:

Intra-frame: The score of each intra-frame link (i.e., C_i in a cost function defined by Eq. (1)) was determined to be constant.

Inter-frame: Inter-frame links between two consecutive frames were given only between two nodes whose corresponding regions spatially overlapped, as was done in [31]. According to the affinity model of [5], an inter-frame link between two nodes N_i and N_j has a cost (i.e., $C_{i,j}$ in a cost function defined by Eq. (1)) consisting of appearance, shape, and motion smoothness costs indicated by $C^A_{i,j}$, $C^S_{i,j}$, and $C^M_{i,j}$ respectively:

$$C_{i,j} = C^A_{i,j} C^S_{i,j} C^M_{i,j} \tag{5}$$

$$C^A_{i,j} = \exp\left(-\frac{a(N_i) \cdot a(N_j)}{||a(N_i)||||a(N_i)||}\right) \tag{6}$$

$$C^S_{i,j} = 1 - \exp\left(-\left(\frac{||h(N_i) - h(N_j)||}{h(N_i) + h(N_j)} + \frac{||w(N_i) - w(N_j)||}{w(N_i) + w(N_j)}\right)\right) \tag{7}$$

$$C^M_{i,j} = 1 - \exp\left(-||\boldsymbol{v}^{(k)}_i - \boldsymbol{v}^{(k)}_j||\right), \tag{8}$$

where $a(N_i)$, $h(N_i)$, and $w(N_i)$ denotes the appearance feature, height, and width of a human window corresponding to N_i. $a(N_i)$ is the concatenation of size-normalized HSV color images of N_i. $\boldsymbol{v}_i^{(k)}$ denotes the directed velocity of a window corresponding to N_i. Note that, in Eq. (3) expressing $\boldsymbol{v}_i^{(k)}$ [3], an unknown variable m remains, but m is determined uniquely in our extended graph because there is a unique path from N_i to any node if the path exists.

In accordance with the previous work, the tracking performance was evaluated with the following criteria [34, 9, 23]:

MT↑: Mostly Tracked trajectories indicate the number of trajectories tracked successfully for more than 80%.

ML↓: Mostly Lost trajectories indicate the number of trajectories tracked for less than 20%.

FP↓ and FN↓: The false-positive and false-negative detection rates, which are respectively computed by normalizing the number of false-positives and false-negatives by the total number of objects.

Frag↓: The number of times a trajectory is interrupted.

IDS↓: ID Switches indicate the number of times two trajectories switch their IDs. It is expected that this criterion directly evaluates whether or not the constraints on smoothness of the proposed method work well, as shown in Fig. 1.

MOTA↑: $1-\frac{\sum_t N_t^{(fp)} N_t^{(fn)} N_t^{(ids)}}{\sum_t N_t^{(gt)}}$, where $N_t^{(fp)}$, $N_t^{(fn)}$, $N_t^{(ids)}$, and $N_t^{(gt)}$ denote the number of false detections, false negatives, ID switches, and true detections at t-th frame, respectively.

MOTP↑: The misalignment between the annotated and the predicted bounding boxes: $\frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}$, where $d_{t,i}$ and c_t denote the overlap between detected and annotated bounding boxes of i-th object and the number of matched pairs of detected and annotated bounding boxes, respectively.

Sign ↑ represents that higher scores indicate better results, while sign ↓ denotes that lower scores indicate better results.

The proposed method was tested with four different parameters, Ours1, Ours1', Ours2, and Ours3. Ours1 and Ours1' used the first-order smoothness constraints, and Ours2 and Our3 used the second- and third-order smoothness constraints, respectively. Here, in n-th-order smoothness constraints, all of $\{1, \cdots, n\}$-th-order smoothness were presented in a graph. The difference between Ours1 and Ours1' is that every other frame of an original sequence was used in Ours1'[4], while Ours1 used all frames.

Ours2 and Ours3 are expected to be better than Ours1. This is because first-order directed velocities were unstable due to the noisy positions of detected human regions. In terms of the amount of noise in dynamic features, Ours1' had less noise than Ours1 because the first-order directed velocities in every other frame (i.e., dynamic features of Ours1') are equal to the second-order directed velocities in the original sequence (i.e., dynamic features of Ours2). The difference between Ours1' and Ours2 is that Ours1' evaluates the static cues of human regions between every other frame, while Ours2 evaluates them between two consecutive frames. This difference caused an increase in dissimilarity between two nodes of a single person in Ours1' rather than in Ours2.

[3] For simplicity, a time index t is ommitted from Eq. (3).

[4] Since the number of frames used in the evaluation of Ours1' is reduced, its results in Tables 2 and 3 are shown just for reference to validate the effect of high-order smoothness.

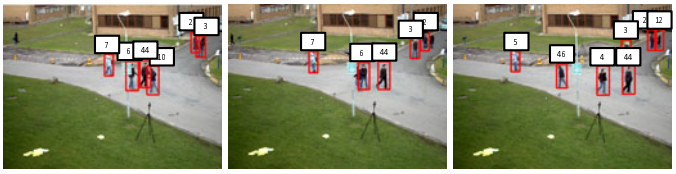


Figure 13: Tracking results of the proposed method in the PETS2009 dataset.

4.1. PETS2009 dataset

The first dataset, PETS2009, is a relatively easy one. No crowded people were observed and a camera observed people obliquely from above so that they were not occluded with each other in many frames. Figure 13 shows a part of tracking results.

The results of quantitative evaluation are shown in Table 1. The proposed method with the first-order smoothness (i.e., Ours1 and Ours1') are inferior to the one with higher-order smoothness constraints. This is because the first-order smoothness is noisy and often causes miss-tracking such as tracking fragments, while the noise is suppressed by evaluating the smoothness every other frame (i.e., Ours1'). The results of the proposed method with the second- and third-order smoothness constraints (i.e., Ours2 and Ours3) are almost same as each other. The results of high-order smoothness constraints are saturated depending on several factors such as the noise level and the moving velocity of a target object. For example, if the positions of detected human regions are more noisy, directed velocities computed between a short frame interval are also noisy and so higher-order smoothness is required. In detections given by the dataset [40], since the positions of detected human regions are not so noisy, directed velocities computed between t-th and $(t+2)$-th frames in Ours2 are sufficiently stable to suppress the negative impact of the noise of the detected positions.

As can be seen in Table 1, while most state-of-the-arts [27, 22, 39, 5] have their strong and weak points, the method proposed in [27] has the best scores. The proposed methods with the second- and third-order smoothness constraints (i.e., Ours2 and Ours3) are comparative to these state-of-the-arts. Since the quantitative performance in this dataset is almost saturated by these state-of-the-arts, comparison with other two datasets (i.e., ETHMS and CAVIAR datasets) may be more interesting.

Table 1: Comparison of tracking accuracy with the PETS2009 dataset. While the best scores in each column are indicated in bold, the underlined ones indicate scores that are within ± 5 % of the best scores. The scores of all other methods were given from their papers.

	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	Frag↓	IDS↓
Energy Min [28]	80.2	–	91.3	4.3	–	–	6	11
PIRMPT [22]	–	–	78.9	**0**	–	–	23	1
OLMOAP [39]	–	–	89.5	**0**	–	–	9	**0**
UHMOT [27]	**97.8**	**75.3**	**100**	**0**	–	–	8	8
OTC w/o learning [5]	78.2	69.4	**100**	**0**	20.3	1.3	**10**	16
OTC [5]	83.0	69.6	**100**	**0**	19.4	1.2	**4**	4
Ours1 (1st)	88.3	62.8	87.8	7.3	1.9	9.6	11	6
Ours1' (1st & skip)	91.4	63.3	92.7	2.4	1.5	7.1	9	4
Ours2 (2nd)	91.9	67.5	97.6	**0**	1.3	6.7	5	2
Ours3 (3rd)	92.3	67.5	97.6	**0**	1.3	5.9	**4**	2

Table 2: Comparison of tracking accuracy with the ETHMS. The best and near-best scores are indicated in bold and by underline, respectively. The scores of all other methods were given from their papers.

	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	Frag↓	IDS↓
PIRMPT [22]	–	–	58.4	8.0	–	–	23	**11**
Online CRF [40]	–	–	68.0	7.2	–	–	**19**	**11**
OTC w/o learning [5]	70.4	59.7	65.9	6.4	3.8	25.4	50	44
OTC [5]	**72.0**	**64.0**	**73.8**	**2.4**	**4.2**	23.6	38	18
Ours1 (1st)	60.4	53.0	60.5	8.9	7.9	31.4	71	27
Ours1' (1st & skip)	65.8	55.6	65.3	7.3	7.6	26.3	39	30
Ours2 (2nd)	69.8	59.2	71.0	6.5	6.5	23.6	24	14
Ours3 (3rd)	70.8	61.0	70.2	4.8	6.3	**22.8**	30	17

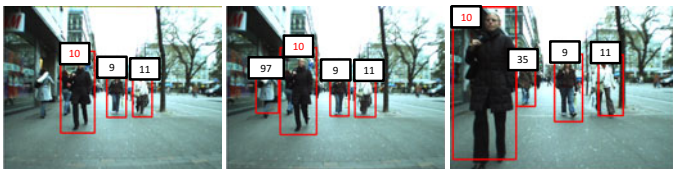


Figure 14: Tracking results of the proposed method in the ETHMS dataset (bahnhoff). A pedestrian of ID=10 could be tracked successfully, while she intersects with a pedestrian of ID=97.

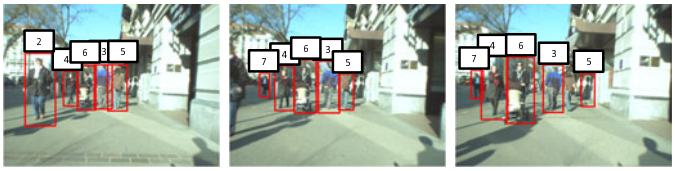


Figure 15: Tracking results of the proposed method in the ETHMS dataset (sunnyday).

4.2. *ETHMS dataset*

In the second dataset, ETHMS, most pedestrians are occluded with each other since the videos were captured from a stroller. Tracking results in bahnhof and sunnyday sequences are shown in Figs. 14 and 15, respectively.

The results of quantitative evaluation are shown in Table 2. The relative performance of the proposed methods (i.e., Ours1, Ours1', Ours2, and Ours3) are almost identical to the one shown in Table 1. While the proposed method with higher-order smoothness constraints (i.e., Ours2 and Ours3) can acquire higher MT, which is comparative with MT of the state-of-the-art [5], the results of other criteria are mired in mediocrity in the proposed method.

One of big differences between this state-of-the-art [5] and the proposed method is that this method [5] improves an appearance model with online learning. Since the proposed method just compares extracted appearance features between two frames as defined in Eq. 6, its matching score is unreliable rather than the online tracklets [5]. The performance of the online tracklets [5] without online appearance learning is also shown in Table 2. The best proposed method, Ours2, is superior to this version in all criteria, MT, ML, Frag, and IDS. These results suggest enhancing the appearance score function of the proposed method for further performance improvement.

In the ETHMS, the regions of many pedestrians are enlarged when they passed each other (e.g., ID=10 in Fig. 14). Such an enlarged region causes long-term complete occlusion, which makes even higher-smoothness ineffective. Figure 14 shows an example where miss-tracking in the base method [31] was corrected by the proposed method. Whereas ID=10 and ID=97 were switched incorrectly by [31], the proposed method tracked them successfully.

4.3. *CAVIAR dataset*

Table 3 shows the results of quantitative evaluation with sequences of the CAVIAR. As with the results of the PETS2009 and ETHMS datasets, (1) the proposed method with the second-

Table 3: Comparison of tracking accuracy with CAVIAR. The best and near-best scores are indicated in bold and by underline, respectively. The scores of all other methods were given from their papers.

	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	Frag↓	IDS↓
PIRMPT [22]	–	–	86.0	0.7	–	–	17	**4**
MOTLGM [36] –	83.5	82.0	90.7	2.7	–	–	**6**	5
OTC w/o learning [5]	81.4	85.2	85.3	1.1	1.2	17.5	28	25
OTC [5]	**86.5**	**87.2**	89.5	**0.0**	**1.0**	**11.4**	8	9
Ours1 (1st)	79.7	80.3	69.2	8.4	3.2	17.1	33	27
Ours1' (1st & skip)	80.7	82.0	72.0	7.0	2.7	16.6	21	19
Ours2 (2nd)	84.0	84.5	88.8	2.8	1.4	14.6	12	**4**
Ours3 (3rd)	84.9	87.1	**90.9**	2.8	1.7	13.4	8	5

Figure 16: Tracking results of the proposed method in the CAVIAR dataset (EnterExitCrossingPaths1cor).

and third-order smoothness constraints demonstrated the improved performance rather than the first-order ones, but (2) the proposed method even with high-order smoothness constraints is inferior to the state-of-the-art [5] in terms of many evaluation criteria. However, the proposed method with the third-order smoothness constraints outperformed (1) the state-of-the-art [5] without online visual learning except for ML and FP and (2) also the full version of state-of-the-art [5] in MT and IDS. These facts reveal that (i) visual cues play an important role for visual tracking and (ii) high-order smoothness and constraints can be also important cues.

Unlike all other methods, the method proposed in [36] employs group-struecture information [37, 33] as an additional cue for tracking. Since these methods [5, 36] including the proposed method got comparative good performance and each of them has its advantages and disadvantages, all prospective cues may be useful for improving the total performance.

Figure 16 shows an example in which the proposed method could successfully track occluded pedestrians (i.e., ID=30).

5. Concluding Remarks

This paper proposed a method for improved globally-optimal tracking of objects using smoothness in motion. Unlike previous methods, high-order constraints of high-order smoothness can be employed by extending a graph with the dynamic features of nodes, while maintaining the first-order link limitation for efficient optimization.

Compared with previous methods, the proposed method improved performance in public datasets. Several visualization results demonstrating its typical effects are also shown.

Important future work includes long-term occlusion handling. During such long-term occlusion, an inter-frame score for object appearance is not useful or is harmful. This problem should be solved by, for example, explicit occlusion handling that discriminate between nodes where target objects are observed and occluded; see [42], for exmaple. In addition to occlusion handling, sophisticated features for visual information should be also employed because its effectiveness has been demonstrated [5, 13]. For fair and conprehensive evaliuation. experiments with a recently-available multi-object tracking benchmark, MOTChallenge [23], is also an important future work.

[1] http://iris.usc.edu/people/yangbo/downloads.html. 7
[2] Caviar. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/. 7
[3] Pets 2009. http://www.cvg.reading.ac.uk/PETS2009/. 7
[4] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *ECCV*, 2010. 1, 2
[5] S. H. Bae and K. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, 2014. 7, 8, 9, 10
[6] T. Baumgartner, D. Mitzel, and B. Leibe. Tracking people and their objects. In *CVPR*, 2013. 1
[7] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, 2006. 1
[8] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1806–1819, 2011. 1
[9] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image and Video Processing*, 2008, 2008. 8
[10] A. A. Butt and R. T. Collins. Multiple target tracking using frame triplets. In *ACCV*, 2012. 2, 3
[11] A. A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *CVPR*, 2013. 2, 3
[12] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(4):941–953, 2013. 3
[13] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *ICCV*, 2015. 3, 10
[14] R. T. Collins. Multitarget data association with higher-order motion models. In *CVPR*, 2012. 3
[15] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *ICCV*, 2013. 1
[16] A. Ess, B. Leibe, and L. J. V. Gool. Depth and appearance for mobile scene analysis. In *ICCV*, 2007. 7
[17] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *19th IEEE Conference on Decision and Control*, 1980. 2
[18] H. Ishikawa. Higher-order gradient descent by fusion-move graph cut. In *ICCV*, 2009. 2
[19] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 1, 2
[20] P. Kohli, M. P. Kumar, and P. H. S. Torr. P & beyond: Move making algorithms for solving higher order functions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(9):1645–1656, 2009. 2
[21] M. Kristan, J. Pers, S. Kovacic, and A. Leonardis. A local-motion-based

probabilistic model for visual tracking. *Pattern Recognition*, 42(9):2160–2168, 2009. 3

[22] C. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011. 8, 9, 10

[23] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, Apr. 2015. arXiv: 1504.01942. 8, 10

[24] B. Leibe, K. Schindler, N. Cornelis, and L. J. V. Gool. Coupled object detection and tracking from static cameras and moving vehicles. *PAMI*, pages 1683–1698, 2008. 1

[25] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009. 7

[26] Y. Ma, Q. Yu, and I. Cohen. Target tracking with incomplete detection. *CVIU*, 113(4):580–587, 2009. 1, 2

[27] G. R. Martin Hofmann, Michael Haag. Unified hierarchical multi-object tracking using global data association. In *PETS*, 2013. 8, 9

[28] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *PAMI*, 36(1):58–72, 2014. 9

[29] S. Pellegrini, A. Ess, and L. J. V. Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010. 1, 2

[30] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006. 1

[31] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 2, 3, 6, 7, 9

[32] C. Stauffer. Estimating tracking sources and sinks. In *IEEE Workshop on Event Mining*, 2003. 2, 3

[33] N. Ukita, Y. Moriguchi, and N. Hagita. People re-identification across non-overlapping cameras using group features. *Computer Vision and Image Understanding*, 144:228–236, 2016. 10

[34] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR*, 2006. 8

[35] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *CVPR*, 2012. 1

[36] L. A. B. B. Xiaojing Chen, Zhen Qin. Multi-person tracking by online learned grouping model with non-linear motion context. *IEEE TRANS. on CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 2015. 10

[37] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *CVPR*, 2011. 10

[38] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In *CVPR*, 2011. 2, 3

[39] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR*, 2012. 7, 8, 9

[40] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR*, 2012. 2, 3, 7, 8, 10

[41] Q. Yu, G. G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *CVPR*, 2007. 1

[42] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 1, 2, 3, 4, 10

[43] L. Zhang, B. Wu, and R. Nevatia. Detection and tracking of multiple humans with extensive pose articulation. In *ICCV*, 2007. 1