Deep Continuous Conditional Random Fields with Asymmetric Inter-object Constraints for Online Multi-object Tracking

Hui Zhou, Wanli Ouyang, Jian Cheng, Xiaogang Wang, and Hongsheng Li

***Abstract*—Online Multi-Object Tracking (MOT) is a challenging problem and has many important applications including intelligence surveillance, robot navigation and autonomous driving. In existing MOT methods, individual object's movements and inter-object relations are mostly modeled separately and relations between them are still manually tuned. In addition, inter-object relations are mostly modeled in a symmetric way, which we argue is not an optimal setting. To tackle those difficulties, in this paper, we propose a Deep Continuous Conditional Random Field (DCCRF) for solving the online MOT problem in a track-by-detection framework. The DCCRF consists of unary and pairwise terms. The unary terms estimate tracked objects' displacements across time based on visual appearance information. They are modeled as deep Convolution Neural Networks, which are able to learn discriminative visual features for tracklet association. The asymmetric pairwise terms model inter-object relations in an asymmetric way, which encourages high-confidence tracklets to help correct errors of low-confidence tracklets and not to be affected by low-confidence ones much. The DCCRF is trained in an end-to-end manner for better adapting the influences of visual information as well as inter-object relations. Extensive experimental comparisons with state-of-the-arts as well as detailed component analysis of our proposed DCCRF on two public benchmarks demonstrate the effectiveness of our proposed MOT framework.**

***Index Terms*—Multi-object tracking, Deep neural networks, Continuous Conditional Random Fields, Asymmetric pairwise terms.**

I. Introduction

ROBUST tracking of multiple objects [1] is a challenging problem in computer vision and acts as an important component of many real-world applications. It aims to reliably recover trajectories and maintain identities of objects of interest in an image sequence. State-of-the-art Multi-Object Tracking (MOT) methods [2], [3] mostly utilize the tracking-by-detection strategy because of its robustness against tracking drift. Such a strategy generates per-frame object detection results from the image sequence and associates the detections into object trajectories. It is able to handle newly appearing objects and is robust to tracking drift. The tracking-by-detection methods can be categorized into offline and online methods. The offline methods [4] use both detection results from past and future with some global optimization techniques for linking detections to generate object trajectories. The online methods, on the other hand, use only detection results up to the current time to incrementally generate object trajectories. Our proposed method focuses on online MOT, which is more suitable for real-time applications including autonomous driving and intelligent surveillance.

In MOT methods, the tracked objects usually show consistent or slowly varying appearance across time. Visual features of the objects are therefore important cues for associating detection boxes into tracklets. In recent years, deep learning techniques have shown great potential in learning discriminative visual features for single-object and multi-object tracking. However, visual cues alone cannot guarantee robust tracking results. When tracked objects with similar appearances occlude or are close to each other, their trajectories might be wrongly associated to other objects. In addition, there also exist misdetections or inaccurate detections by imperfect object detectors. Such difficulties escalate when the camera is hold by hand or fixed on a car. Each object moves according to its own movement pattern as well as the global camera motion. Solving such problems was explored by modeling interactions between tracked objects in the optimization model. For online MOT methods, there were investigations on modeling inter-object interactions with social force models [5], [6], [7], relative spatial and speed differences [8], [9], [10], and relative motion constraints [3], [11]. Most of the previous methods model pairwise inter-object interactions in symmetric mathematical forms, i.e., pairs of objects influence each other with the same magnitude.

However, such pairwise object interactions should be directional and modeled in an asymmetric form, while existing methods model such interactions in a symmetric way. For instance, large-size detection boxes are more likely to be noisy (if measured in actual pixels). Smaller boxes should influence larger boxes more than large ones to small ones because the smaller ones usually provide more accurate localization for objects. Similarly, high-confidence trajectories should influence low-confidence ones more and low-confidence ones should have minimal impact on the high-confidence ones. In this way, the more accurate detections or trajectories could help correct errors of the inaccurate ones and would not be affected by the inaccurate ones much. Moreover, in existing methods,

Hui Zhou and Jian Cheng are with the School of Information and Communication Engineering at University of Electronic Science and Technology of China, Chengdu, China. Hongsheng Li and Xiaogang Wang are with the Department of Electronic Engineering at The Chinese University of Hong Kong, Hong Kong, China. Wanli Ouyang is with University of Sydney, Sydney, Australia. This work is done when Hui Zhou is a Research Assistant in the Department of Electronic Engineering at The Chinese University of Hong Kong. Hongsheng Li is the corresponding author (e-mail: hsli@ee.cuhk.edu.hk).

individual object's movements and inter-object interactions are usually modeled separately. The relations between the two terms are mostly manually tuned and not effectively studied in a unified framework.

To tackle the difficulties, we propose a Deep Continuous Conditional Random Field (DCCRF) with asymmetric inter-object constraints for solving the problem of online MOT. The DCCRF inputs a pair of consecutive images at time $t-1$ and time t, and tracked object's past trajectories up to time $t-1$. It estimates locations of the tracked objects at time t. The DCCRF optimizes an objective function with two terms, the unary terms, which estimate individual object's movement patterns, and the asymmetric pairwise terms, which model interactions between tracked objects. The unary terms are modeled by a deep Convolutional Neural Network (CNN), which is trained to estimate each individual object's displacement between time $t-1$ and time t with each object's visual appearance. The asymmetric pairwise terms aim to tackle the problem caused by object occlusions, object mis-detections and global camera motion. For two neighboring tracked trajectories, the pairwise influence is different along each direction to let the high-confidence trajectory assists the low-confidence one more. Our proposed DCCRF utilizes mean-field approximation for inference and is trained in an end-to-end manner to estimate the optimal displacement for each tracked object. Based on such estimated object locations, a final visual-similarity CNN is proposed for generating the final detection association results.

The contribution of our proposed online MOT framework is two-fold. (1) A novel DCCRF model is proposed for solving the online MOT problem. Each object's individual movement patterns as well as inter-object interactions are studied in a unified framework and trained in an end-to-end manner. In this way, the unary terms and pairwise terms of our DCCRF can better adapt each other to achieve more accurate tracking performance. (2) An asymmetric inter-object interaction term is proposed to model the directional influence between pairs of objects, which aims to correct errors of low-confidence trajectories while maintain the estimated displacements of the high-confidence ones. Extensive experiments on two public datasets show the effectiveness of our proposed MOT framework.

II. Related Work

There are a large number of methods on solving the multi-object tracking problem. We focus on reviewing online MOT methods that utilize interactive constraints, as well as single-object and multi-object tracking algorithms with deep neural networks.

Interaction models for MOT. Social force models were adopted in MOT methods [5], [6], [7] to model pairwise interactions (attraction and repulsion) between objects. These methods required objects' 3D positions for modeling inter-object interactions, which were obtained by visual odometry.

Grabner et al. [12] assumed that the relative positions between feature points and objects were more or less fixed over short-time intervals. Generalized Hough transform was therefore used to predict each target's location with the assist of supporter feature points. Duan et al. [10] proposed mutual relation models to describe the spatial relations between tracked objects and to handle occluded objects. Such constraints are learned by an online structured SVM. Zhang and Maaten [9] incorporated spatial constraints between objects into an MOT framework to track objects with similar appearances.

The CRF algorithm [13] was used frequently in segmentation tasks to model the relationship between different pixels in the spatial-domain. There were also many works that modeled the multi-object tracking problem with CRF models. Yang and Nevatia [14] proposed an online-learned CRF model for MOT, and assumed linear and smooth motion of the objects to associate past and future tracklets. Andriyenko et al. [15] modeled multi-object tracking as optimizing discrete and continuous CRF models. A continuous CRF was used for enforcing motion smoothness, and a discrete CRF with a temporal interaction pairwise term was optimized for data association. Milan et al. [16] designed new CRF potenials for modeling spatio-temporal constraints between pairs of trajectories to tackle detection and trajectory-level occlusions.

Deep learning based object tracking. Most existing deep learning based tracking methods focused on single object tracking, because deep neural networks were able to learn powerful visual features for distinguishing the tracked objects from the background and other similar objects. Early single-object tracking methods [17], [18] with deep learning focused on learning discriminative appearance features for online training. However, due to the large learning capacitity of deep neural networks, it is easy to overfit the data. [19], [20] pretrained deep convolutional neural networks on large-scale image dataset to learn discriminative visual features, and updated the classifier online with new training samples. More recently, methods that did not require model updating were proposed. Tao et al. [21] utilized Siamese CNNs to determine visual similarities between image pacthes for tracking. Bertinetto et al. [22] changed the network into a fully convolutional setting and achieved real-time running speed.

Recently, deep models have been applied to multi-object tracking. Milan et al. [23] proposed an online MOT framework with two RNNs. One RNN was used for state (object locations, motions, etc.) prediction and update, and the other for associating objects across time. However, this method did not utilize any visual feature and relied solely on spatial locations of the detection results. [24], [25] replaced the hand-crafted features (e.g., color histograms) with the learned features between image patches by a Siamese CNN, which increases the discriminative ability. However, those methods focused on modeling individual object's movement patterns with deep learning. Inter-object relations were not integrated into deep neural networks.

III. Method

The overall framework of our proposed MOT method is illustrated in Fig. 1. We propose a Deep Continuous Conditional Random Field (DCCRF) model for solving the online MOT problem. At each time t, the framework takes past tracklets up to time $t-1$ and detection boxes at time t as

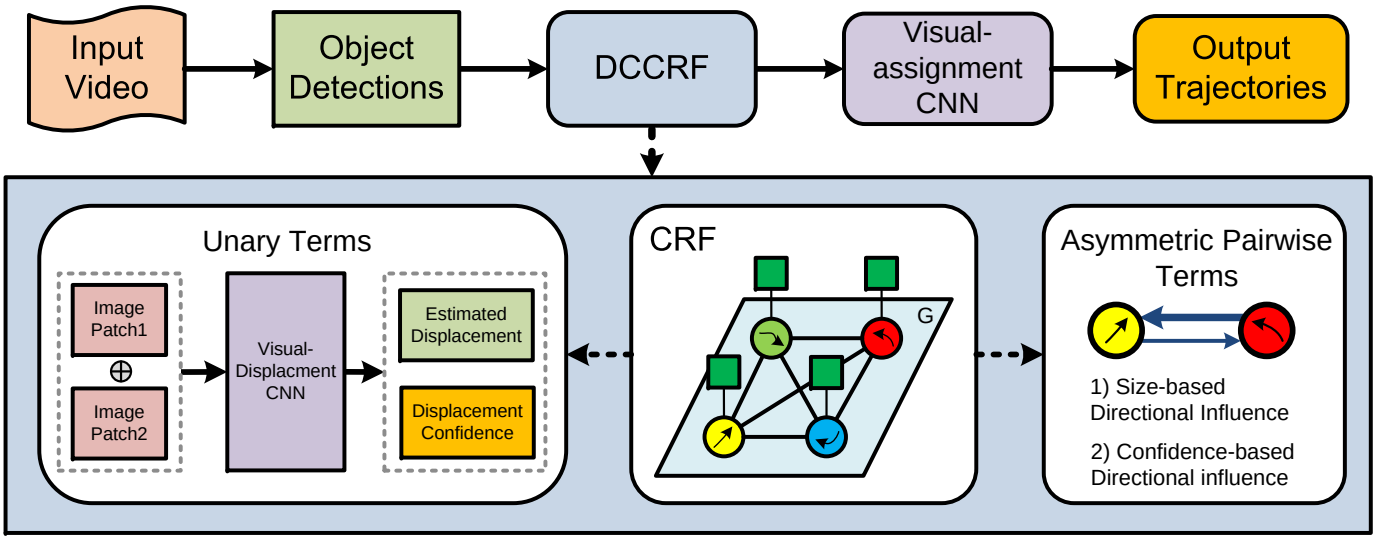


Fig. 1: Illustration of the overall multi-object tracking framework. The proposed Deep Continuous Conditional Random Field consists of unary terms and asymmetric pairwise terms (Section III-A). The unary terms are modeled by a visual-displacement CNN, which take pairs of object image patches as inputs and output the estimated object displacements between time $t-1$ and time t (Section III-A1). The asymmetric pairwise terms encourage to use high-confidence tracklets for correcting errors of low-confidence tracklets (Section III-A2). Size-based and confidence-based directional weighting functions are investigated.

inputs, and generates new tracklets up to time t. At each time t, new tracklets are also initialized and current tracklets are terminated if tracked objects disappear from the scene.

The core components of the proposed DCCRF consist of unary terms and asymmetric pairwise terms. The unary terms of our DCCRF are modeled by a deep CNN that estimates the individual tracked object's displacements between consecutive times $t-1$ and t. The asymmetric pariwise terms aim to model inter-object interactions, which consider differences of speeds, visual-confidence, and object sizes between neighboring objects. Unlike interaction terms in existing MOT methods, which treat inter-object interactions in a symmetric way, asymmetric relationship terms are proposed in our DCCRF. For pairs of tracklets in our DCCRF model, the proposed asymmetric pairwise term models the two directions differently, so that high-confidence trajectories with small-size detection boxes can help correct errors of low-confidence trajectories with large-size detection boxes. Based on the estimated object displacements by DCCRF, we adopt a visual-similarity CNN and Hungarian algorithm to obtain the final tracklet-detection associations.

A. Deep Continuous Conditional Random Field (DCCRF)

The proposed DCCRF takes object trajectories up to time $t-1$ and video frame at time t as inputs, and outputs each tracked object's displacement between time $t-1$ and time t. Let $\mathbf{r}$ represents a random field defined over a set of variables $\{r_1, r_2, \cdots, r_n\}$, where each of the n variables represents the visual and motion information of an object tracklet. Let $\mathbf{d}$ represents another random field defined over variables $\{d_1, d_2, \cdots, d_n\}$, where each variable represents the displacement of an object between time $t-1$ and time t. The domain of each variable is the two-dimensional space $\mathbb{R}^2$, denoting the x- and y-dimensional displacements of tracked objects. Let I represents the new video frame at time t.

The goal of our conditional random field $(\mathbf{r}, \mathbf{d})$ is to maximize the following conditional distribution,

$$P(\mathbf{d}|\mathbf{r}, I) = \frac{1}{Z(\mathbf{t})} \exp\left(-E(\mathbf{d}, \mathbf{r}, I)\right), \tag{1}$$

where $E(\mathbf{d}, \mathbf{r}, I)$ represents the Gibbs energy and $Z(\mathbf{t}) = \int_{\mathbf{r}} \exp(-E(\mathbf{d}, \mathbf{r}))d\mathbf{r}$ is the partition function. Maximizing the conditional distribution w.r.t. $\mathbf{d}$ is equivalent to minimizing the Gibbs energy function,

$$E(\mathbf{d}, \mathbf{r}, I) = \sum_{i=1}^{n} \phi(d_i, r_i, I) + \sum_{i,j} \psi(d_i, d_j, r_i, r_j, I), \tag{2}$$

where $\phi(d_i, r_i, I)$ and $\psi(d_i, d_j, r_i, r_j)$ are the unary terms and pairwise terms.

After the displacements $\mathbf{d}$ of tracked objects between time $t-1$ and time t are obtained, individual object's estimated locations at time t can be easily calculated for associating tracklets and detection boxes to generate tracklets up to time t. Such displacements are then iteratively calculated for the following time frames. Without loss of generality, we only discuss the approach for optimizing object displacements between time $t-1$ and time t in this section.

1) **Unary terms:** For the ith object tracklet, the unary term $\phi(d_i, r_i, I)$ of our DCCRF model is defined as

$$\phi(d_i, r_i, I) = w_{i,1} \left(d_i - f_d(r_i, I)\right)^2. \tag{3}$$

This term penalizes the quadratic deviations between the final output displacement d_i and the estimated displacement by a visual displacement estimation function f_d. $w_{i,1}$ is an online adaptive parameter for the ith object that controls to trust more the estimated displacement based on the ith object's visual

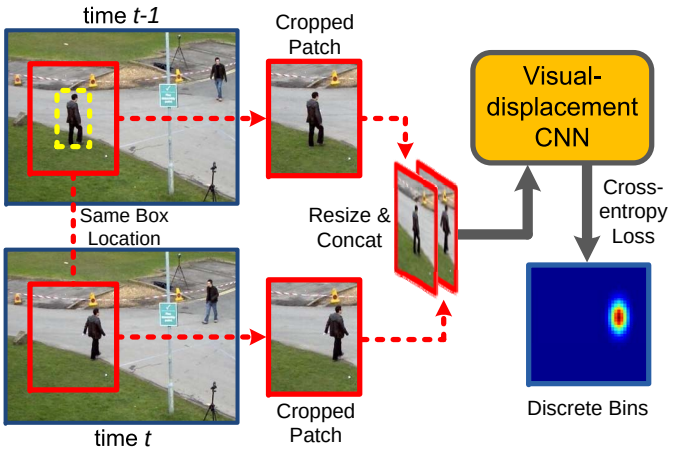


Fig. 2: Illustration of the visual-displacement CNN for modeling the unary terms. Two image patches are cropped from the same box location centered at the object location at time $t-1$ as inputs. The visual-displacement CNN estimates the confidences of discrete object displacements and is trained with cross-entropy loss.

cues (the unary terms) or based on inter-object relations (the pairwise terms). Intuitively, when the visual displacement estimator f_d has higher confidence on its estimated displacement, $w_{i,1}$ should be larger to bias the final output d_i towards the visually inferred displacements. On the other hand, when f_d has lower confidence on its estimation, due to object occlusion or appearing of similar objects, $w_{i,1}$ should be smaller to let the final displacement d_i be mainly inferred by inter-object constraints.

In our framework, the visual displacement estimation function f_d is modeled as a deep Convolution Neural Network (CNN) that utilizes only the tracked objects' visual information for estimating its location displacement between time $t-1$ and time t. For each tracked object r_i, our visual-displacement CNN takes a pair of images patched from frames $t-1$ and t as inputs, and outputs the object's inferred displacement. A network structure similar to ResNet-101 [26] (except for the topmost layer) is adopted for our visual-displacement CNN. The network inputs and outputs are illustrate in Fig. 2. For the inputs, given currently tracked object r_i's bounding box location b_i at time $t-1$, a larger bounding box $\bar{b}_i$ centered at b_i is first created. Two image patches are cropped at the same spatial location $\bar{b}_i$ but from different frames at time $t-1$ and time t. They are then concatenated along the channel dimension to serve as the inputs for our visual-displacement CNN. The reasons for using a larger bounding box $\bar{b}_i$ instead of the original box b_i are to tolerate large possible displacement between the two consecutive frames and also to incorporate more visual contextual information of the object for more accurate displacement estimation. After training with thousands of such pairs, the visual-displacement CNN is able to capture important visual cues from image-patch pairs to infer object displacements between time $t-1$ and time t.

For the CNN outputs, instead of directly estimating objects' two dimensional x- and y-dimensional displacements, we discretize possible 2D continuous displacements into a 2D discrete grid $\{p_i^1, p_i^2, \cdots, p_i^m\}$ (bottom-right part in Fig. 2), where $p_i^k \in \mathbb{R}^2$ represents the displacement corresponding to the kth bin of the ith object. The visual-displacement CNN is trained to output confidence scores c_i^k for the displacement bins p_i^k with a softmax function. The cross-entropy loss is therefore used to train the CNN, and the final estimated displacement for the tracked object r_i is calculated as the weighted average of all possible displacements $\sum_{k=1}^m c_i^k p_i^k$, where $\sum_{k=1}^m c_i^k = 1$. In practice, we discretize displacements into $m = 20 \times 20$ bins, which is a good trade-off between discretization accuracy and robustness. Note that there are existing tracking methods [22], [27] that also utilize pairs of image patches as inputs to directly estimate object displacements. However, in our method, we propose to use cross-entropy loss for estimating displacements and find that its result achieves more accurate and robust displacement estimations in our experiments. More importantly, it provides displacement confidence scores $\{c_i^1, \cdots, c_i^m\}$ for calculating the adaptive parameter $w_{i,1}$ in Eq. (3) to weight the unary and pairwise terms.

The confidence weight $w_{i,1}$ is obtained by the following equation,

$$w_{i,1} = \sigma\left(a_1 \max(\mathbf{c}_i) + b_1\right), \tag{4}$$

where σ is the sigmoid function constraining the range of $w_{i,1}$ being between 0 and 1, $\max(\mathbf{c}_i)$ obtains the maximal confidence of $\mathbf{c}_i = \{c_i^1, c_i^2, \cdots, c_i^m\}$, and a_1 and b_1 are learnable scalar parameters. In our experiments, the learned parameter a_1 is generally positive after training, which denotes that, if the visual-displacement CNN is more confident about its displacement estimations, the value of $w_{i,1}$ is larger and the final output displacement d_i can be more biased towards the visually inferred displacement $f_d(r_i, I)$. Otherwise, the final displacement d_i can be biased to be inferred by inter-object constraints.

If the energy function E in Eq. (2) consists of only the unary terms $\phi(d_i, r_i, I)$, the final output displacement d_i can be solely dependent on each tracked object's visual information without considering inter-object constraints.

2) **Asymmetric pairwise terms:** The pairwise terms in Eq. (2) are utilized to model asymmetric inter-object relations between object tracklets for regularizing the final displacement results $\mathbf{d}$. To handle global camera motion, we assume that from time $t-1$ to time t, the speed differences between two tracked objects should be maintained, i.e.,

$$\psi(d_i, d_j, r_i, r_j, I) = (1 - w_{i,1}) \sum_k w_{ij,2}^{(k)} \left(\Delta d_{ij} - \Delta s_{ij}\right)^2, \tag{5}$$

where $\Delta d_{ij} = d_i - d_j$ is the displacement (which can be viewed as speed) difference between objects i and j at time t, $\Delta s_{ij} = s_i - s_j$ is the speed difference at the previous time $t-1$, and $w_{ij,2}^{(k)}$ are a series of weighting functions (two in our experiments) that control the directional influences between the pair of objects,

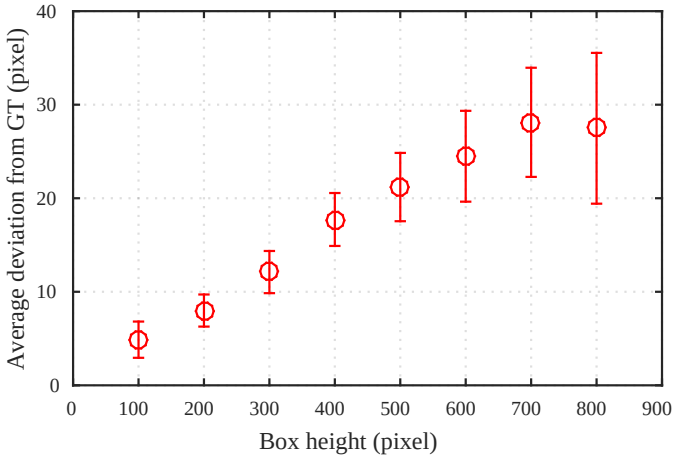
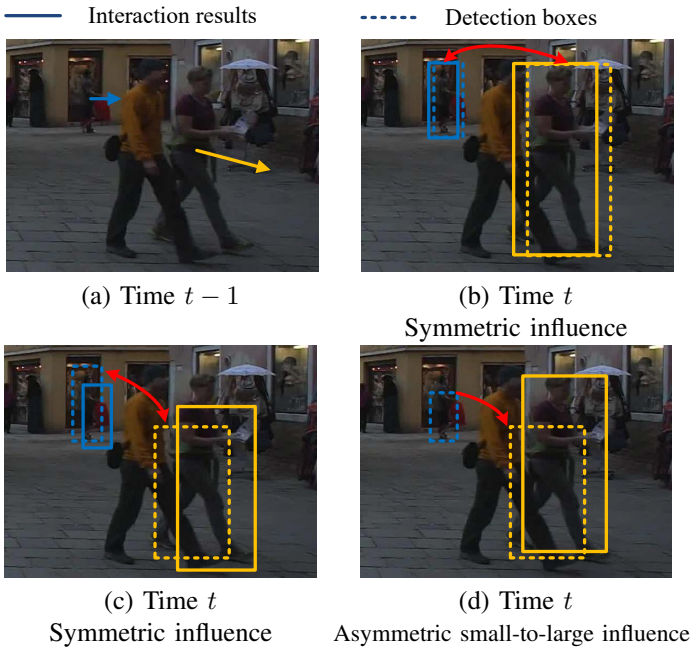


Fig. 3: The average deviation of detection boxes from their ground-truth locations is approximately proportional to the detection box size. The statistics are calculated from the 2DMOT16 training set [28] where the detection boxes are provided by the dataset.



(a) Time $t-1$ (b) Time t Symmetric influence

(c) Time t Symmetric influence (d) Time t Asymmetric small-to-large influence

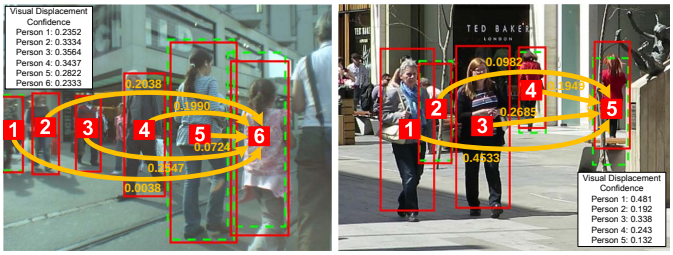
Fig. 4: Illustration of symmetric and asymmetric inter-object constraints. (a) Two tracked objects at time $t-1$ with their estimated speeds (denoted by arrows). (b) Tracked objects at time t. Symmetric inter-object constraints work well when there are little detection noise for all detection boxes. (c) When there is localization noise for the large-size detection box, symmetric inter-object constraints are likely to deteriorate the tracking of the small-size object. (d) Asymmetric small-to-large-size inter-object constraints are more robust than symmetric inter-object constraints, even when there is localization noise for the small-size detection box.

For better modeling inter-object relations, two important observations are made to define the asymmetric weighting functions $w_{ij,2}^{(k)}$. 1) For detection boxes, in terms of localization accuracy, larger object detection boxes are more likely to be noisy, while smaller ones tend to be more stable (as shown in Fig. 3). This is because the displacements of both large and small detection boxes are all recorded in pixels in our tracking frameworks. Noisy large detection boxes would significantly influence the displacement estimation for other boxes. This problem is illustrated in Fig. 4. The two targets in Fig. 4(a) have accurate locations and speeds which can be used to build inter-object constraints at time $t-1$. When the detector outputs roughly accurate bounding boxes for both targets at time t, symmetric inter-object constraints could well refine the objects' locations (see Fig. 4(b)). However, since the larger-size detection boxes are more likely to be noisy, using the symmetric inter-object constraints would significantly affect tracking results of the small-size objects (see Fig. 4(c)). In contrast, small-size objects have smaller localization errors and could better infer larger-size objects' locations. Asymmetric small-to-large-size inter-object constraints are robust, even when the smaller-size detection box is noisy(see Fig. 4(d)). Therefore, between a pair of tracked objects, the one with smaller detection box should have more influence to infer the displacement of the ones with larger detection box, and the object with a larger box should have less chance to deteriorate the displacement estimation of the smaller one. 2) If our above mentioned visual-displacement CNN has high confidence for an object's displacement, this object's visually inferred displacement should be used more to infer other objects' displacements. On the other hand, the objects with low confidences on their visually inferred displacements should not affect other objects with high-confidence displacements. Based on the two observations, we model the weighting function $w_{ij,2}^{(k)}$ by a product of a size-based weighting function and a confidence-based weighting function between a pair of tracked objects as

$$\begin{aligned} w_{ij,2}^{(k)} =& \sigma(a_{21}^{(k)} \log(area_i/area_j) + b_{21}^{(k)})) \times \\ & \sigma(a_{22}^{(k)}(\max(\mathbf{c}_i) - \max(\mathbf{c}_j)) + b_{22}^{(k)}) \end{aligned} \quad (6)$$

where σ denotes the sigmoid function, $area_i$ denotes the size of the ith tracked object at time $t-1$, $\max(\mathbf{c}_i)$ obtains the maximal displacement confidence from $\{c_i^1, c_i^2, \cdots, c_i^m\}$ by our proposed visual-displacement CNN, and $a_{21}^{(k)}$, $b_{21}^{(k)}$, $a_{22}^{(k)}$, $b_{22}^{(k)}$ are learnable scalar parameters. In our DCCRF, these parameters can be learned by back-propagation algorithm with mean-field approximation. If we use the mean-field approximation for DCCRF inference, the influence from object r_i to r_j and that from r_j and r_i are different (see next subsection for details). After training, we see that $a_{21}^{(k)} > 0$ and $a_{22}^{(k)} < 0$, which means that smaller $area_i/area_j$ and larger $\max(\mathbf{c}_i) - \max(\mathbf{c}_j)$ lead to greater weights. It validates our above mentioned observations that objects with smaller sizes and greater visual-displacement confidences should have greater influences to other objects, but not the other around.

In Fig. 5, we show example values of one learned weighting function $w_{ij,2}^{(k)}$. In Fig. 5(a), compared with object 6, objects 2-4 are of smaller sizes and also higher visual confidences. With the directional weighting functions, they have greater influence



(a) (b)

Fig. 5: Example values of asymmetric weighting function $w_{ij}^{(k)}$ between tracked objects of different sizes and confidences. Green dashed rectangles denote estimated object locations by the unary terms (visual-displacement CNN) only. Red rectangles denote estimated object locations by both unary and pairwise terms. Orange arrows and numbers denote the weighting function values from one object to the other. (a) Small-size objects (objects 2-4) help correct errors of large-size object (object 6) with higher diretional weighting function values. (b) Objects with higher visual-displacement confidences (objects 1, 3, 4) help correct errors of the object (object 5) with lower visual-displacement confidences.

to correct errors of tracking object 6 (red vs. green rectangles of object 6) and are not affected much by the erroneous estimation of object 6. Similar directional weighting function values can be found in Fig. 5(b), where objects 1, 3, 4 with high visual-displacement confidences are able to correct tracking errors of object 5 with low visual-displacement confidence.

3) **Inference:** For our unary terms, we utilize forward propagation of the visual-displacement CNN for calculating objects' estimated displacements and displacement confidences $\{c_i^1, c_i^2, \cdots, c_i^m\}$. After the unary term inference, the overall maximum posterior marginal inference is achieved by mean-field approximation. This approximation yields an iterative message passing for approximate inference. Our unary terms and pairwise terms are both of quadratic form. The energy function is convex and the optimal displacement is obtained as the mean value of the energy function,

$$d_i \longleftarrow \frac{w_{i,1} f_d(r_i, I) + (1 - w_{i,1}) \sum_{i \neq j} \sum_k w_{ij,2}^{(k)} (d_j - \Delta s_{ij})}{w_{i,1} + (1 - w_{i,1}) \sum_{i \neq j} \sum_k w_{ij,2}^{(k)}}. \tag{7}$$

In each iteration, the node i receives messages from all other objects to update its displacement estimation. The mean-field approximation is usually converged in 5-10 iterations. The above displacement update equation clearly shows the differences between the messages transmitted from i to j and that from object j to i because of the asymmetric weighting functions $w_{ij,2}^{(k)}$. For a pair of objects, $w_{ij,2}^{(k)}$ and $w_{ji,2}^{(k)}$ are generally different. Even if $w_{i,1} = w_{j,1}$, when $w_{ij,2}^{(k)} > w_{ji,2}^{(k)}$, object j has greater influence to i than that from j to i.

A detailed derivation of Eq. (7) is given as follows. The mean-field method is to approximate the distribution $P(\mathbf{d}|\mathbf{r}, I)$ with a distribution $Q(\mathbf{d}|\mathbf{r}, I)$, which can be expressed as a product of independent marginals $Q(\mathbf{d}|\mathbf{r}, I) = \prod_1^N Q_i(d_i|\mathbf{r}, I)$. The optimal approximation of Q is obtained by minimizing Kullback-Leibler (KL) divergence between P and Q. The solution for Q has the following form,

$$\log(Q_i(d_i|\mathbf{r}, I)) = E_{i \neq j}[\log(P(\mathbf{d}|\mathbf{r}, I))] + \text{const}, \tag{8}$$

where $E_{i \neq j}$ denotes expectation under Q distributions over all variables d_j for $j \neq i$. The inference is formulated as

$$\begin{aligned} &\log(Q_i(d_i|\mathbf{r}, I)) = \phi(d_i, r_i, I) + \sum_{i,j} \psi(d_i, d_j, r_i, r_j, I) \\ &= w_{i,1}(d_i - f_d(r_i, I))^2 \\ &\quad + (1 - w_{i,1}) \sum_{i \neq j} \sum_k w_{ij,2}^{(k)} (\Delta d_{ij} - \Delta s_{ij})^2 \\ &= (w_{i,1} + (1 - w_{i,1}) \sum_{i \neq j} \sum_k w_{ij,2}^{(k)}) d_i^2 \\ &\quad - 2(w_{i,1} f_d(r_i, I) + (1 - w_{i,1}) \sum_{i \neq j} \sum_k w_{ij,2}^{(k)} (d_j + \Delta s_{ij})) d_i \\ &\quad + \text{const}. \end{aligned} \tag{9}$$

Each $\log(Q_i(d_i|\mathbf{r}, I))$ is a quadratic form with respect to d_i and its means therefore are

$$\mu_i = \frac{w_{i,1} f_d(r_i, I) + (1 - w_{i,1}) \sum_{i \neq j} \sum_k w_{ij,2}^{(k)} (d_j - \Delta s_{ij})}{w_{i,1} + (1 - w_{i,1}) \sum_{i \neq j} \sum_k w_{ij,2}^{(k)}}. \tag{10}$$

The inference task is to minimize $P(\mathbf{d}|\mathbf{r}, I)$. Since we approximate conditional distribution with product of independent marginals, an estimate of each d_i is obtained as the expected value μ_i of the corresponding quadratic function,

$$\widehat{d_i} = \arg\min_{d_i}(Q_i(d_i|\mathbf{r}, I)) = \mu_i. \tag{11}$$

B. The Overall MOT Algorithm

The overall algorithm with our proposed DCCRF is shown in Algorithm 1. At each time t, the DCCRF inputs are existing tracklets up time $t-1$, and consecutive frames at time $t-1$ and time t. It outputs each tracklet's displacement estimation. After obtaining displacement estimations $\widehat{d_i}$ for each tracklet r_i by DCCRF, its estimated location at time t can be simply calculated as the summation of its location b_{r_i} at time $t-1$ and its estimated displacement $\widehat{d_i}$, i.e.,

$$\widehat{b_{r_i}} = b_{r_i} + \widehat{d_i}. \tag{12}$$

Based on such estimated locations, we utilize a visual-similarity CNN (Section III-B1) as well as the Intersection-over-Union value as the criterion for tracklet-detection association to generate longer tracklets (Section III-B2). To make our online MOT system complete, we also specify our detailed strategies for tracklet initialization (Section III-B3), occlusion handling and tracklet termination (Section III-B4).

1) Visual-similarity CNN: The tracklet-detection associations need to be determined based on visual cues and spatial cues simultaneously. We propose a visual-similarity CNN for calculating visual similarities between image patches cropped at bounding box locations in the same frame. The visual-similarity CNN has similar network structure as our visual-displacement CNN in Section III-A1. However, the network

Algorithm 1: The overall MOT algorithm

Input: Images sequence up to time t, per-frame object detections $b_1, b_2, b_3, \cdots$
Output: Object tracklets up to time t.

for *time* = $1, \cdots, t$ **do**
  Estimate tracked object displacements d_i (Section III-A);
  Estimate tracklet location $\widehat{b_{r_i}}$ (Eq. (12));
  Calculate tracklet-detection similarities $(\widehat{b_{r_i}}, b_j)$ (Section III-B2);
  Hungarian algorithm to obtain tracklet-associated detection b_{r_i} (Section III-B2);
  for *each tracklet* r_i **do**
    if $IoU(\widehat{b_{r_i}}, b_j) \geq 0.5$ **then**
      Append b_j to tracklet r_i;
    else if $0.5 > IoU(\widehat{b_{r_i}}, b_j) \geq 0.3$ **then**
      Append $(b_j + \widehat{b_{r_i}})/2$ to tracklet r_i;
    else
      r_i has no detection association;
      if *no association* $> m$ *frames* **then**
        Tracklet termination (Section III-B4);
      else
        Append $\widehat{b_{r_i}}$ to tracklet r_i (Section III-B4);
      end
    end
  end
  for *detections not associated to tracklets* **do**
    if *high overall similarity for* k *frames* **then**
      Tracklet initialization (Section III-B3);
    end
  end
end

takes image patches in the same video frame as inputs and outputs the confidence whether the input pair represents the same object. It is therefore trained with a binary cross-entropy loss. In addition, the training samples are generated differently for the visual-similarity CNN. Instead of cropping two consecutive video frames at the same bounding box locations as the visual-displacement CNN, the visual-similarity CNN requires positive pairs to be cropped at different locations of the same object at anytime in the same video, while the negative pairs to be image patches belonging to different objects. For cropping image patches, we dont't enlarge the object's bounding box, which is also different to our visual-displacement CNN. During training, the ratio between positive and negative pairs are set to 1:3 and the network is trained similarly to that of visual-displacement CNN.

2) Tracklet-detection association: Given the estimated tracklet locations and detection boxes at time t, they are associated with detection boxes based on the visual and spatial similarities between them. The associated detection boxes can then be appended to their corresponding tracklets to form longer ones up to time t. Let $\widehat{b_{r_i}}$ and b_j denote the ith tracklet's estimated location and the jth detection box at time t. Their visual similarity calculated by the visual-similarity CNN in Section III-B1 is denoted as $V(\widehat{b_{r_i}}, b_j)$. The spatial similarity between the estimated tracklet locations and detection boxes are measured as the their box Intersection-over-Union values $IoU(\widehat{b_{r_i}}, b_j)$. If a detection box is tried to be associated with multiple tracklets, Hungrian algorithm is utilized to determine the optimal associations with the following overall similarity,

$$S(\widehat{b_{r_i}}, b_j) = V(\widehat{b_{r_i}}, b_j) + \lambda IoU(\widehat{b_{r_i}}, b_j), \tag{13}$$

where λ is the weight balancing the visual and spatial similarities and is set to 1 in our experiments. After the box association by Hungarian algorithm, if a tracklet is associated with a detection box that has an IoU value greater than 0.5 with it, the associated detection box are directly appended to the end of the tracklet. If the IoU value is between 0.3 and 0.5, the average of the associated detection box and estimated tracklet box are appended to the tracklet to compensate for the possible noisy detection box. If the IoU value is smaller than 0.3, tracklet might be considered as being terminated or temporally occluded (Section III-B4).

3) Tracklet initialization: If an object detection box at time $t-1$ is not associated to any tracklet in the above tracklet-detection association step, it is treated as a *candidate box* for initializing new tracklets. For each such candidate box at time $t-1$, its visually inferred displacement between time $t-1$ and t is first obtained by our visual-displacement CNN in Section III-A1. Its estimated box location can be easily calculated following Eq. (12). The visual similarities V and spatial similarities IoU between the estimated box at t and candidate boxes at t are calculated. To form new candidate tracklet, the candidate box at time $t-1$ is only associated with the candidate box at time t that has 1) greater-than-0.3 IoU and 2) greater-than-0.8 visual similarity with its estimated box location. If there are multiple candidate associations, Hungarian algorithm is utilized to associate the candidate box at t to its optimal candidate association at $t-1$ according to the overall similarities (Eq. (13)). If none of the candidate associations at time t satisfies the above two conditions with the candidate box at $t-1$, the candidate box is ignored and would not be used for tracking initialization. Such operations are iterated over time to generate longer candidate tracklets. If a candidate tracklet is over k frames ($k = 4$ for pedestrain tracking with 25-fps videos), it is initialized as a new tracklet.

4) Occlusion handling and tracklet termination: If a past tracklet is not associated to any detection box at time t, the tracked object is considered as being possibly occluded or temporally missed. For a possibly occluded object, we directly associate its past tracklet to its estimated location by our DCCRF at time t to create a virtual tracklet. The same operation is iterated for m frames, i.e., if the virtual tracklet is not associated to any detection box for more than m time steps, the virtual tracklet is terminated. For pedestrian tracking, we empirically set $m = 5$.

IV. Experiments

In this section, we present experimental results of the proposed online MOT algorithm. We first introduce evaluation datasets and implementation details for our proposed

framework in Sections IV-A and IV-B. In Section IV-C, we compare the proposed method with state-of-the-art approaches on the public MOT datasets. The individual components of our proposed method are evaluated in Section IV-D.

A. Datasets and Evaluation Metric

We conduct experiments on the 2DMOT15 [29] and 2DMOT16 [28] benchmarks, which are widely used to evaluate the performance of MOT methods. Both of them have two tracks: public detection boxes [2], [3], [24] and private detection boxes [30], [31]. For comparing with only the performance of tracking algorithms, we evaluate our method with the provided public detection boxes.

1) 2DMOT15: This dataset is one of the largest datasets with moving or static cameras, different viewpoints and different weather conditions. It contains a total of 22 sequences, half for training and half for testing, with a total of 11286 frames (or 996 seconds). The training sequences contain over 5500 frames, 500 annotated trajectories and 39905 annotated bounding boxes. The testing sequences contain over 5700 frames, 721 annotated trajectories and 61440 annotated bounding boxes. The public detection boxes in 2DMOT15 are generated with aggregated channel features (ACF).

2) 2DMOT16: This dataset is an extension to 2DMOT15. Compared to 2DMOT15, new sequences are added and the dataset contains almost 3 times more bounding boxes for training and testing. Most sequences are in high resolution, and the average pedestrian number in each video frame is 3 times higher than that of the 2DMOT15. In 2DMOT16, deformable part models (DPM) based methods are used to generate public detection boxes, which are more accurate than boxes in 2DMOT15.

3) Evaluation Metric: For the quantitative evaluation, we adopt the popular CLEAR MOT metrics [29], which include:

- **MOTA:** Multiple Object Tracking Accuracy. This metric is usually chosen as the main performance indicator for MOT methods. It combines three types of errors: false positives, false negatives, and identity switches.
- **MOTP:** Multiple Object Tracking Precision. The misalignment between the annotated and the predicted bounding boxes.
- **MT:** Mostly Tracked targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.
- **ML:** Mostly Lost targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.
- **FP:** The total number of false positives.
- **FN:** The total number of false negatives (missed targets).
- **ID Sw:** The total number of identity switches. Please note that we follow the stricter definition of identity switches as described in MOT challenge.
- **Frag:** The total number of times a trajectory is fragmented (i.e., interrupted during tracking).

B. Implementation details

1) Training schemes and setting: For visual-displacement and visual-similarity CNNs, we adopt ResNet-101 [26], [32] as the network structure and replace the topmost layer to output displacement confidence or same-object confidence. Both CNN are pretrained on the ImageNet dataset. For cropping image patches from $\bar{b}_i$, we enlarge each detection box b_i by a factor of 5 in width and 2 in height to obtain $\bar{b}_i$. Image patches for the two CNNs are cropped at the same locations from consecutive frames as described in Section III-A1, which are then resized to 224×224 as the CNN inputs.

We train our proposed DCCRF in three stages. In the first stage, the proposed visual-displacement CNN is trained with the cross-entropy loss and batch Stochastic Gradient Descent (SGD) with a batch size of 5. The initial learning rate is set to 10^{-6} and is decreased by a factor of 1/10 every 50,000 iterations. The training generally converges after 600,000 iterations. In the second stage, the learned visual-displacement CNN from stage-1 is fixed and other parameters in our DCCRF are trained with L_1 loss,

$$\zeta_{loss} = \sum \|\widehat{d}_i - d_i^{gt}\|_1, \tag{14}$$

where $\widehat{d}_i$ and d_i^{gt} are estimated displacements and the ground-truth displacements for the ith tracked object. In the final stage, the DCCRF is trained in an end-to-end manner with the above L_1 loss and the cross-entropy loss for visual-displacement CNN in unary terms. We find that 5 iterations of the mean-field approximation generate satisfactory results. The DCCRF is trained with an initial learning rate of 10^{-4}, which is decreased by a factor of 1/3 every 5,000 iterations. The training typically converges after 3 epochs.

Our code is implemented with MATLAB and Caffe. The overall tracking speed of the proposed method on MOT16 test sequences is 0.1 fps using the 2.4GHz CPU and a Maxwell TITAN X GPU without some acceleration library packages.

2) Data augmentation: To introduce more variation into the training data and thus reduce possible overfitting, we augment the training data. For pre-training the visual-displacement CNN, the input images are image patches centered at detection boxes. We augment the training samples by random flipping as well as randomly shifting the cropping positions by no more than $\pm 1/5$ of detection box width or height for x and y dimensions respectively. For end-to-end training the DCCRF, except for random flipping of whole video frames, the time intervals between the two input video frames are randomly sampled from the interval of $[1, 3]$ to generate more frame pairs with larger possible displacements between them.

C. Quantitative results on 2DMOT15 and 2DMOT16

On the MOT2015 and MOT2016 datasets, we test our proposed method and compare it with state-of-the-art MOT methods[1] including SMOT [33], MDP [2], SCEA [3], CEM [34], RNN_LSTM [23], RMOT [11], TC_ODAL [38], CNNTCM [36], SiameseCNN [25], oICF [39], NOMT [37], CDA_DDAL [24]. The results of the compared methods are listed in Tables I and II. We focus on the MOTA value as the main performance indicator, which is a weighted combination

[1]Note that only methods in peer-reviewed publications are compared in this paper. ArXiv papers that have not undergone peer-review are not included.

TABLE I: Quantitative results by our method and state-of-the-art MOT methods on 2DMOT15 dataset. Bold numbers indicate the best results of online or offline methods respectively). ↑ denotes that higher is better and ↓ represents the opposite.

Tracking Mode	Method	**MOTA**↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	ID Sw↓	Frag↓
Offline	SMOT [33]	18.2%	71.2%	2.8%	54.8%	8780	40310	1148	2132
Offline	CEM [34]	19.3%	70.7%	8.5%	46.5%	14180	34591	813	1023
Offline	DCO_X [35]	19.6%	71.4%	5.1%	54.9%	10652	38232	521	**819**
Offline	SiameseCNN [25]	29.0%	71.2%	8.5%	48.4%	**5160**	37798	639	1316
Offline	CNNTCM [36]	29.6%	71.8%	11.2%	**44.0%**	7786	34733	712	943
Offline	NOMT [37]	**33.7%**	**71.9%**	**12.2%**	**44.0%**	7762	**32547**	**442**	823
Online	TC_ODAL [38]	15.1%	70.5%	3.2%	55.8%	12970	38538	637	1716
Online	RNN_LSTM [23]	19.0%	71.0%	5.5%	45.6%	11578	36706	1490	2081
Online	RMOT [11]	18.6%	69.6%	5.3%	53.3%	12473	36835	684	1282
Online	oICF [39]	27.1%	70.0%	6.4%	48.7%	7594	36757	**454**	1660
Online	SCEA [3]	29.1%	71.1%	8.9%	47.3%	6060	36912	604	**1182**
Online	MDP [2]	30.3%	**71.3%**	**13.0%**	38.4%	9717	**32422**	680	1500
Online	CDA_DDAL [24]	32.8%	70.7%	9.7%	42.2%	**4983**	35690	614	1583
Online	Proposed Method	**33.6%**	70.9%	10.4%	**37.6%**	5917	34002	866	1566

TABLE II: Quantitative results by our proposed method and state-of-the-art MOT methods on 2DMOT16 dataset. ↑ denotes that higher is better and ↓ represents the opposite.

Tracking Mode	Method	**MOTA**↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	ID Sw↓	Frag↓
Offline	TBD [40]	33.7%	76.5%	7.2%	54.2%	5804	112587	2418	2252
Offline	LTTSC-CRF [41]	37.6%	75.9%	9.6%	55.2%	11969	101343	481	1012
Offline	LINF [42]	41.0%	74.8%	11.6%	51.3%	7896	99224	430	963
Offline	MHT_DAM [43]	42.9%	**76.6%**	13.6%	46.9%	**5668**	97919	499	659
Offline	JMC [4]	46.3%	75.7%	15.5%	**39.7%**	6373	90914	657	1114
Offline	NOMT [37]	**46.4%**	**76.6%**	**18.3%**	41.4%	9753	**87565**	**359**	**504**
Online	OVBT [44]	38.4%	75.4%	7.5%	47.3%	11517	99463	1321	2140
Online	EAMTT_pub [45]	38.8%	75.1%	7.9%	49.1%	8114	102452	965	1657
Online	oICF [39]	43.2%	74.3%	11.3%	48.5%	6651	96515	**381**	1404
Online	CDA_DDAL [24]	43.9%	74.7%	10.7%	44.4%	6450	95175	676	1795
Online	Proposed Method	**44.8%**	**75.6%**	**14.1%**	**42.3%**	**5613**	**94125**	968	**1378**

of false negatives (FN), false positives (FP) and identity switches (ID Sw). Note that offline methods generally have higher MOTA than online methods because they can utilize not only past but also future information for object tracking and are only listed for reference here. Our proposed online MOT method outperforms all compared online methods and most offline methods [2], [3], [39], [24], [25]. As shown by the quantitative results, our proposed method is able to alleviate the difficulties caused by object mis-detections, noisy detections, and short-term occlusion. The qualitative results are shown in Fig. 6.

Compared with SCEA [3], which also models inter-object interactions and speed differences to handle mis-detections caused by global camera motion, our learned DCCRF shows better performance, especially in FN for our more accurate displacement prediction which is able to recover more mis-detections. Our proposed method also outperforms MDP [2] in terms of MOTA and FP by a large margin. MDP learns to predict four target states (active, tracked, lost and inactive) for each tracked object. However, it only models tracked object's movement patterns with a constant speed assumption, which is likely to result in false tracklet-detection associations and thus increases FP. CDA_DDAL [24] focuses on using discriminative visual features by a siamese CNN for tracklet-detection associations, which is not robust for occlusions and is easy to increase FN. Compared with other algorithms DCO_X [35] and LTTSC-CRF [41] which also use conditional random field approximation to solve MOT problems, the results show that our proposed DCCRF has great advantages over other CRF-based methods in MOTA.

However, our method produces more ID switches than some compared methods, which is due to long-term occlusions that cannot be solved by our method.

TABLE III: Component analysis of our proposed DCCRF on 2DMOT2016 dataset. ↑ denotes that higher is better and ↓ represents the opposite.

Method	**MOTA**↑	FP↓	FN↓	ID Sw↓
Proposed DCCRF	44.8%	5613	94125	968
Unary-only	41.9%	7392	97618	876
Unary-only+L_1-loss (reg)	34.2%	12089	104810	3134
DCCRF w/o size-asym	43.6%	8063	93724	1035
DCCRF w/o cfd-asym	43.8%	7353	94163	969
DCCRF w/ symmetry	43.4%	9100	93076	1104

D. Component analysis on 2DMOT16

To analyze the effectiveness of different components in our proposed framework, we also design a series of baseline methods for comparison. The results of these baselines and our final method are reported in Table III. Similar to the above experiments, we focus on MOTA value as the main performance indicator. 1) Unary-only: this baseline utilizes only our unary terms in DCCRF, i.e., the visual-displacement CNN, with our overall MOT algorithm. Such a baseline model considers only tracked objects' appearance information. Compared with our proposed DCCRF, it has a 3% MOTA drop, which denotes

(a) MOT1603 (b) MOT1606 (c) MOT1607

(d) MOT1608 (e) MOT1612 (f) MOT1614

Fig. 6: Example tracking results by our proposed method on 2DMOT16 dataset.

that the inter-object relations are crucial for regularizing each object's estimated displacement and should not be ignored. 2) Unary-only+L_1-loss (reg): since our visual-displacement CNN is trained with proposed cross-entropy loss instead of conventional L_1 or L_2 losses for regression problems, we train a visual-displacement CNN with smooth L_1-loss and test it in the same way as the above unary-only baseline. Compared with unary-only baseline, unary-only+L_1-loss has a significant 7% MOTA drop, which demonstrates that our proposed cross-entropy loss results in much better displacement estimation accuracy. 3) DCCRF w/o cfd-asym and DCCRF w/o size-asym: the weighting functions of the pairwise term in our proposed DCCRF have two terms, a confidence-asymmetric term and a size-asymmetric term. We test using only one of them in our DCCRF's pairwise terms. The results show more than 1% drop in terms of MOTA for both baseline methods compared with our proposed DCCRF, which validates the need of both terms in the weighting functions. 4) DCCRF w/ symmetry: this baseline method replaces the asymmetric pairwise term in our DCCRF with a symmetric one,

$$(1 - w_{i,1}) \sum_k \exp\left(-\frac{(l_i - l_j)^2}{2a_2^{(k)2}}\right)(\Delta d_{ij} - \Delta s_{ij})^2, \quad (15)$$

where l_i is the coordinates of ith object's center position and $a_2^{(k)}$ are learnable Gaussian kernel bandwidth parameters. Such a symmetric term assumes that the speed differences between close-by objects should be better maintained across time, while those between far-away objects are less regularized. There is a 1% MOTA drop compared with our proposed DCCRF, which shows our asymmetric term is beneficial for the final performance. We also try to directly replace the sigmoid function in Eq. (5) with a Gaussian-like function in the weighting function (Eq. (15)), which results in even worse performance.

In addition to the above, we also conduct experiments to analysize the effects of different hyper-parameters to show our DCCRF robustness. 1) The λ controls the weight between the visual-similarity term and the DCCRF location prediction term for tracklet-detection association in Eq. (13). We test three different values of λ and the results of different λ are reported in Table IV, which the final performance is not sensitive to the λ value. 2) The k is the length of a candidate tracklet to create an actual tracklet in section III-B3. We additionally test $k = 8$ in Table V, which shows slightly performance drop, because larger k will cause more low-confidence detections to be ignored. 3) The m denotes the number of consecutive frames of missing objects to terminate its associated tracklet in section III-B4. We additionally test $m = 8$ and the results in Table VI show the peformance is not sensitive to the choice of m.

TABLE IV: Effects of different λ parameter.

λ	0.5	1	1.5
MOTA	43.8%	44.8%	43.5%

TABLE V: Results by different tracklet initialization parameter k.

k	MOTA	FP	FN
4	44.8%	5613	94125
8	43.0%	4837	98433

V. Conclusion

In this paper, we present the Deep Continuous Conditional Random Field (DCCRF) model with asymmetric inter-object constraints for solving the MOT problem. The unary terms are modeled as a visual-displacement CNN that estimates object displacements across time with visual information. The asymmetric pairwise terms regularize inter-object speed

TABLE VI: Results by different tracklet termination parameter m.

m	MOTA	FP	FN
5	44.8%	5613	94125
8	44.7%	6861	92976

differences across time with both size-based and confidence-based weighting functions to weight more on high-confidence tracklets to correct tracking errors. By jointly training the two terms in DCCRF, the relations between objects' individual movement patterns and complex inter-object constraints can be better modeled and regularized to achieve more accurate tracking performance. Extensive experiments demonstrate the effectiveness of our proposed MOT framework as well as the individual components of our DCCRF.

Acknowledgment: This work is supported in part by SenseTime Group Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants CUHK14213616, CUHK14206114, CUHK14205615, CUHK419412, CUHK14203015, CUHK14239816, CUHK14207814, CUHK14208417, CUHK14202217, in part by the Hong Kong Innovation and Technology Support Programme Grant ITS/121/15FX, in part by the National Natural Science Foundation of China under Grant 61671125, Grant 61201271, and Grant 61301269, and in part by the China Postdoctoral Science Foundation under Grant 2014M552339.

REFERENCES

[1] W. Luo, J. Xing, X. Zhang, X. Zhao, and T. K. Kim, "Multiple object tracking: A literature review," *arXiv preprint arXiv:1409.7618*, 2014.

[2] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4705–4713.

[3] J. Hong Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, "Online multi-object tracking via structural constraint event aggregation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1392–1400.

[4] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," *arXiv preprint arXiv:1608.05404*, 2016.

[5] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 261–268.

[6] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[7] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn, "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 120–127.

[8] X. Chen, Z. Qin, L. An, and B. Bhanu, "Multiperson tracking by online learned grouping model with nonlinear motion context," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 12, pp. 2226–2239, 2016.

[9] L. Zhang and L. Van Der Maaten, "Preserving structure in model-free tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 756–769, 2014.

[10] G. Duan, H. Ai, S. Cao, and S. Lao, "Group tracking: Exploring mutual relations for multiple object tracking," *Computer Vision–ECCV 2012*, pp. 129–143, 2012.

[11] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 33–40.

[12] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1285–1292.

[13] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1529–1537.

[14] B. Yang and R. Nevatia, "An online learned crf model for multi-target tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2034–2041.

[15] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1926–1933.

[16] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[17] H. Li, Y. Li, and F. Porikli, "Robust online visual tracking with a single convolutional neural network," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 194–209.

[18] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems*, 2013, pp. 809–817.

[19] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International Conference on Machine Learning*, 2015, pp. 597–606.

[20] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[21] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1420–1429.

[22] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," *arXiv preprint arXiv:1606.09549*, 2016.

[23] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks." in *AAAI*, 2017, pp. 4225–4232.

[24] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[25] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese cnn for robust target association," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 33–40.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[27] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 749–765.

[28] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[29] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[30] H. Roberto, L.-T. Laura, C. Daniel, and R. Bodo, "A novel multi-detector fusion framework for multi-object tracking," *arXiv preprint arxiv:1705.08314*, 2017.

[31] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *European Conference on Computer Vision Workshops*, 2016, pp. 36–42.

[32] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang *et al.*, "Crafting gbd-net for object detection," *arXiv preprint arXiv:1610.02579*, 2016.

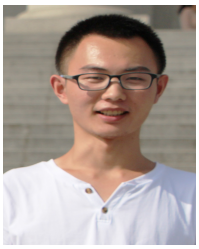
[33] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2304–2311.

[34] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, 2014.

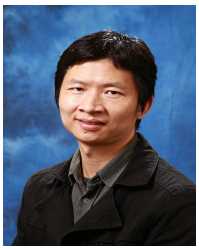
[35] A. Milan, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2054–2068, 2016.

[36] B. Wang, L. Wang, B. Shuai, Z. Zuo, T. Liu, K. Luk Chan, and G. Wang, "Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association," in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1–8.
[37] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3029–3037.
[38] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1218–1225.
[39] H. Kieritz, S. Becker, W. Hübner, and M. Arens, "Online multi-person tracking using integral channel features," in *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*. IEEE, 2016, pp. 122–130.
[40] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.
[41] N. Le, A. Heili, and J.-M. Odobez, "Long-term time-sensitive costs for crf-based tracking by detection," in *Computer Vision-Eccv 2016 Workshops, Pt Ii*, vol. 9914, no. EPFL-CONF-221401. Springer Int Publishing Ag, 2016, pp. 43–51.
[42] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle, "Improving multi-frame data association with sparse representations for robust near-online multi-object tracking." in *ECCV (8)*, 2016, pp. 774–790.
[43] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4696–4704.
[44] Y. Ban, S. Ba, X. Alameda-Pineda, and R. Horaud, "Tracking multiple persons based on a variational bayesian model," in *ECCV Workshop on Benchmarking Mutliple Object Tracking*, 2016.
[45] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections." in *ECCV Workshops (2)*, 2016, pp. 84–99.



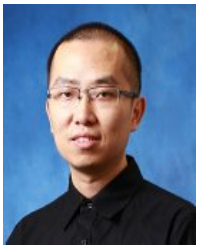
Hui Zhou received the bachelor degree at university of science and electronic technology of china(UESTC) in 2015. He is currently pursuing the master's degree at UESTC. His research interests include computer vision and machine learning..



Wanli Ouyang obtained Ph.D from the Dept. of Electronic Engineering , the Chinese University of Hong Kong. He is now a Senior Lecturer at the University of Sydney. His research interests include deep learning and its application to computer vision and pattern recognition, image and video processing.

Jian Cheng received the Ph.D. degree in Pattern Recognition and Intelligent System from Shanghai Jiao Tong University in 2006. From 2006 to 2007, he was an assistant researcher at the Chengdu Information Technology of Chinese Academy of Sciences Co., Ltd. He is currently a professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. His main research interests include machine learning, computer vision, remote sensing image analysis, multimodal image classification, video surveillance and scene understanding, human behavior analysis, etc.

Xiaogang Wang received the PhD degree from the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology in 2009. He is currently an Associate Professor in the Department of Electronic Engineering at The Chinese University of Hong Kong. His research interests include computer vision and machine learning.



Hongsheng Li received the bachelors degree in automation from East China University of Science and Technology, and the masters and doctorate degrees in computer science from Lehigh University, Pennsylvania, in 2006, 2010, and 2012, respectively. From 2013-2015, he was an associate professor in the School of Electronic Engineering at University of Electronic Science and Technology of China. He is currently a research assistant professor in the department of Electronic Engineering at the Chinese University of Hong Kong.