Online Multi-object Visual Tracking using a GM-PHD Filter with Deep Appearance Learning

Nathanael L. Baisa*

***Abstract*—We propose a new online multi-object visual tracker based on a Gaussian mixture Probability Hypothesis Density (GM-PHD) filter in combination with a similarity Convolutional Neural Network (CNN). The GM-PHD filter estimates the states and cardinality of an unknown and time varying number of targets in the scene handling target birth, death, clutter (false alarms) and missing detections in a unified framework, and has a linear complexity with the number of targets. However, it lacks the identity of targets. We combine spatio-temporal and visual similarities obtained from object bounding boxes and deep CNN appearance features, respectively, to alleviate its shortcoming of labelling targets across frames. We apply this developed method for tracking multiple targets in video sequences acquired under varying environmental conditions and targets density using a tracking-by-detection approach. Finally, we carry out extensive experiments on Multiple Object Tracking 2016 (MOT16) and 2017 (MOT17) benchmark datasets and find out that our tracker significantly outperforms several state-of-the-art trackers in terms of tracking accuracy and precision.**

***Index Terms*—Visual tracking, GM-PHD filter, Gaussian mixture, CNN features, Re-identification, MOT challenge**

I. Introduction

Multi-target tracking is an active research field in computer vision with a wide variety of applications such as intelligent surveillance, autonomous driving, robot navigation and augmented reality. It essentially associates the detections corresponding to the same object over time i.e. it assigns consistent labels to the tracked targets in each video frame to generate a trajectory for each target. These can be performed using online [1] [2] or offline [3] [4] [5] approaches. Online methods estimate the target state at each time instant and depends on predictive models in case of miss-detections to carry on tracking, however, both past and future observations are used in offline (batch) methods to overcome miss-detections. Although offline trackers can generally outperform the online trackers, they are limited for time-critical real-time applications.

Traditionally, multi-target trackers have been developed by finding associations between targets and observations using Joint Probabilistic Data Association Filter (JPDAF) [6] and Multiple Hypothesis Tracking (MHT) [7]. However, these approaches have faced challenges not only in the uncertainty caused by data association but also in algorithmic complexity that increases exponentially with the number of targets and measurements. Recently, a unified framework which directly extends single to multiple target tracking by representing multi-target states and observations as Random Finite Sets (RFS) was developed by Mahler [8] which not only addresses the problem of increasing complexity, but also estimates the states and cardinality of an unknown and time varying number of targets in the scene by allowing for target birth, death, clutter (false alarms), and missing detections. It propagates the first-order moment of the multi-target posterior, called the Probability Hypothesis Density (PHD) [9], rather than the full multi-target posterior. This approach is flexible, for instance, it has been used to find the detection proposal with the maximum weight as the target position estimate for tracking a target of interest in dense environments by removing the other detection proposals as clutter [10] [11]. Furthermore, the standard PHD filter was extended to develop a novel N-type PHD filter ($N \geq 2$) for tracking multiple targets of different types in the same scene [12] [13]. However, this approach does not naturally include target identity in the framework because of the indistinguishability assumption of the point process; additional mechanism is necessary for labelling each target.

More recently, Convolutional Neural Network (CNN) features have demonstrated outstanding results on various recognition tasks [14] [15] [16]. It has also shown better performance on object detection [17] and person re-identification [18] [19] problems. Motivated by this, some deep learning based trackers [20] [21] have been developed mainly for a single object tracking due to its powerful capturing capability of discriminative appearance features of a target of interest against background and other similar objects. Even though most deep learning based methods focused on single object tracking, recently deep learning models have been applied to multi-target tracking, for instance, [3] replaced the hand-engineered features with the learned features using a Siamese CNN which has shown increased discrimination of targets but it works offline. However, the advantages of CNNs in Random Finite Set based filters have not been explored which works online and suitable for real-time applications.

In this work, we propose an online multi-object visual tracker using tracking-by-detection approach for real-time applications. Accordingly, we make the following four contributions. First, we apply the GM-PHD filter in combination with the similarity CNN for tracking multiple targets in video sequences acquired under varying environmental conditions and targets density. Second, we formulate how to combine (fuse) spatio-temporal and visual similarities obtained from

*N. L. Baisa is currently a research scientist at AnyVision. This work was done when the author was with the Department of Computer Science, University of Lincoln, Lincoln LN6 7TS, United Kingdom. (e-mail: NBaisa@lincoln.ac.uk).

bounding boxes of objects and their CNN appearance features, respectively, to construct a cost to be minimized (similarity maximized) by the Hungarian algorithm for labelling each target. Third, we use the developed visual similarity from CNN as a person re-identification method to re-identify lost objects for consistently labelling them. Finally, we make extensive experiments on Multiple Object Tracking 2016 (MOT16) and 2017 (MOT17) benchmark datasets using the public detections provided in the benchmark's test set.

The paper is organized as follows. In section II, the GM-PHD filter for video tracking context is described in detail. The similarity CNN is briefly explained in section III. In section IV and V, data association and person re-identification methods for labelling of each target, respectively, are described. The experimental results are analyzed and compared in section VI. The main conclusions and suggestions for future work are summarized in section VII.

II. The GM-PHD Filter

The Gaussian mixture implementation of the standard PHD (GM-PHD) filter [9] is a closed-form solution of the PHD filter that assumes a linear Gaussian system. It has two steps: prediction and update. Before stating these two steps, certain assumptions are needed: 1) each target follows a linear Gaussian model:

$$y_{k|k-1}(x|\zeta) = \mathcal{N}(x; F_{k-1}\zeta, Q_{k-1}) \qquad (1)$$

$$f_k(z|x) = \mathcal{N}(z; H_k x, R_k) \qquad (2)$$

where $y_{k|k-1}(.|\zeta)$ is the single target state transition probability density at time k given the previous state ζ and $f_k(z|x)$ is the single target likelihood function which defines the probability that z is generated (observed) conditioned on state x. $\mathcal{N}(.; m, P)$ denotes a Gaussian density with mean m and covariance P; F_{k-1} and H_k are the state transition and measurement matrices, respectively. Q_{k-1} and R_k are the covariance matrices of the process and the measurement noises, respectively. The measurement noise covariance R_k can be measured off-line from sample measurements i.e. from ground truth and detection of training data [22] as it indicates detection performance. 2) A current measurement driven birth intensity inspired by but not identical to [23] is introduced at each time step, removing the need for the prior knowledge (specification of birth intensities) or a random model, with a non-informative zero initial velocity. The intensity of the spontaneous birth RFS is a Gaussian mixture of the form

$$\gamma_k(x) = \sum_{v=1}^{V_{\gamma,k}} w_{\gamma,k}^{(v)} \mathcal{N}(x; m_{\gamma,k}^{(v)}, P_{\gamma,k}^{(v)}) \qquad (3)$$

where $V_{\gamma,k}$ is the number of birth Gaussian components, $w_{\gamma,k}^{(v)}$ is the weight accompanying the Gaussian component v, $m_{\gamma,k}^{(v)}$ is the current measurement and zero initial velocity used as mean, and $P_{\gamma,k}^{(v)}$ is birth covariance for Gaussian component v.

3) The survival and detection probabilities are independent of the target state: $p_{S,k}(x_k) = p_{S,k}$ and $p_{D,k}(x_k) = p_{D,k}$.

Adaptive birth: We use adaptive measurement-driven approach for birth of targets. Each detection $z_k \in Z_k$ is associated with detection confidence score $s_k \in [0, 1]$. We use more confident (strong) detections based on their score for birth of targets as they are more likely to represent a potential target. Confident detections used for birth of targets will be $Z_{b,k} = \{z_{b,k} : s_k \geq s_t\} \subseteq Z_k$ where s_t is a detection score threshold. In fact, s_t governs the relationship between the number of false positives (clutter) and miss-detections (false negatives). Increasing the value of s_t gives more miss-detections and less false positives, and vice versa. The initial birth weight $w_{\gamma,k}^{(v)}$ in Eq. (3) is also weighted by s_k to give high probability for more confident detections for birth of targets i.e. $w_{\gamma,k}^{(v)} = s_k w_{\gamma,k}^{(v)}$. However, all measurements Z_k are used for the update step.

Prediction: It is assumed that the posterior intensity at time $k-1$ is a Gaussian mixture of the form

$$\mathcal{D}_{k-1}(x) = \mathcal{D}_{k-1|k-1}(x) = \sum_{v=1}^{V_{k-1}} w_{k-1}^{(v)} \mathcal{N}(x; m_{k-1}^{(v)}, P_{k-1}^{(v)}), \qquad (4)$$

where V_{k-1} is the number of Gaussian components of $\mathcal{D}_{k-1}(x)$ and it equals to the number of Gaussian components after pruning and merging at the previous iteration. Under these assumptions, the predicted intensity at time k is given by

$$\mathcal{D}_{k|k-1}(x) = \mathcal{D}_{S,k|k-1}(x) + \gamma_k(x), \qquad (5)$$

where

$$\mathcal{D}_{S,k|k-1}(x) = \quad p_{S,k} \textstyle\sum_{v=1}^{V_{k-1}} w_{k-1}^{(v)} \mathcal{N}(x; m_{S,k|k-1}^{(v)}, P_{S,k|k-1}^{(v)}),$$

$$m_{S,k|k-1}^{(v)} = F_{k-1} m_{k-1}^{(v)},$$

$$P_{S,k|k-1}^{(v)} = Q_{k-1} + F_{k-1} P_{k-1}^{(v)} F_{k-1}^T,$$

where $\gamma_k(x)$ is given by Eq. (3).

Since $\mathcal{D}_{S,k|k-1}(x)$ and $\gamma_k(x)$ are Gaussian mixtures, $\mathcal{D}_{k|k-1}(x)$ can be expressed as a Gaussian mixture of the form

$$\mathcal{D}_{k|k-1}(x) = \sum_{v=1}^{V_{k|k-1}} w_{k|k-1}^{(v)} \mathcal{N}(x; m_{k|k-1}^{(v)}, P_{k|k-1}^{(v)}), \qquad (6)$$

where $w_{k|k-1}^{(v)}$ is the weight accompanying the predicted Gaussian component v, and $V_{k|k-1}$ is the number of predicted Gaussian components and it equals to the number of born targets and the number of persistent (surviving) components. The number of persistent components is actually the number of Gaussian components after pruning and merging at the previous iteration.

Update: The posterior intensity (updated PHD) at time k is also a Gaussian mixture and is given by

$$\mathcal{D}_{k|k}(x) = (1 - p_{D,k})\mathcal{D}_{k|k-1}(x) + \sum_{z \in Z_k} \mathcal{D}_{D,k}(x; z), \quad (7)$$

where

$$\mathcal{D}_{D,k}(x; z) = \sum_{v=1}^{V_{k|k-1}} w_k^{(v)}(z) \mathcal{N}(x; m_{k|k}^{(v)}(z), P_{k|k}^{(v)}),$$

$$w_k^{(v)}(z) = \frac{p_{D,k} w_{k|k-1}^{(v)} q_k^{(v)}(z)}{c_{s_k}(z) + p_{D,k} \sum_{l=1}^{V_{k|k-1}} w_{k|k-1}^{(l)} q_k^{(l)}(z)},$$

$$q_k^{(v)}(z) = \mathcal{N}(z; H_k m_{k|k-1}^{(v)}, R_k + H_k P_{k|k-1}^{(v)} H_k^T),$$

$$m_{k|k}^{(v)}(z) = m_{k|k-1}^{(v)} + K_k^{(v)}(z - H_k m_{k|k-1}^{(v)}),$$

$$P_{k|k}^{(v)} = [I - K_k^{(v)} H_k] P_{k|k-1}^{(v)},$$

$$K_k^{(v)} = P_{k|k-1}^{(v)} H_k^T [H_k P_{k|k-1}^{(v)} H_k^T + R_k]^{-1}$$

The clutter intensity due to the scene, $c_{s_k}(z)$, in Eq. (7) is given by

$$c_{s_k}(z) = \lambda_t c(z) = \lambda_c A c(z), \quad (8)$$

where $c(.)$ is the uniform density over the surveillance region A, and λ_c is the average number of clutter returns per unit volume i.e. $\lambda_t = \lambda_c A$.

After update, weak Gaussian components with weight $w_k^{(v)} < T = 10^{-5}$ are pruned, and Gaussian components with Mahalanobis distance less than $U = 4$ pixels from each other are merged. These pruned and merged Gaussian components are predicted as existing (persistent) targets in the next iteration. Finally, Gaussian components of the pruned and merged intensity with means corresponding to weights greater than 0.5 as a threshold are selected as multi-target state estimates (we use the pruned and merged intensity rather than the posterior intensity as it gives better results).

III. Visual-similarity CNN

Visual cues are very crucial for associating tracklets with detections (in our case current filtered outputs) for robust online multi-object tracking. In this work, we propose a visual-similarity CNN for computing visual affinities between image patches cropped at bounding box locations. We adopted the ResNet [16] as the network structure (ResNet50) in a Siamese topology by replacing the topmost layer to output same-object confidence. We train the network with a binary cross-entropy loss which outputs the confidence whether the input pair represents the same object or not. We use a transfer learning approach i.e. it is pre-trained on the ImageNet dataset [24] rather than training the network from scratch. Basically, we use two steps transfer learning. In the first step, we fine-tune the pre-trained network on the ImageNet dataset [24] to DukeMTMC dataset [25] for same/different classification. This helps us to adapt the network for a 2-way classification of persons for re-identification. This DukeMTMC dataset has 702 person identities for training from which we extract positive and negative samples. In the second step, we fine-tune again the pre-trained network on DukeMTMC dataset to MOT15 [26] and MOT16/17 [27] training datasets (MOT16 and MOT17 have the same training dataset though MOT17 is claimed to have more accurate ground truth and is used in our experiment). From all the training dataset of MOT15 (TUD-Stadmitte, TUD-Campus, PETS09-S2L1, ETH-Bahnhof and ETH-Sunnyday) and MOT16/17 (5 sequences), we produce about 521 person identities from which we extract positive and negative samples for training our network. This helps the network more adapt to the MOT benchmark test sequences as well as the network can learn the inter-frame variations. 10% of this training set (of each person identity) is used for validation and 2 video sequences of MOT16/17 training dataset (02 and 09) are used for testing of the similarity network. For this testing set, we produce about 66 person identities. In our analysis, these two steps transfer learning gives better results than merging the two datasets (DukeMTMC and MOT) together.

We resize all the training images to 256×256 and then subtract the mean image from all the images which is computed from all the training images. During training, we randomly crop all the images to 224×224 and then mirror horizontally. We use a random order of images by reshuffling the dataset. The positive pairs are generated by randomly sampling image patches from the same person identities whereas the negative pairs are the image patches corresponding to different classes/identities. We set the ratio between the positive and negative pairs to 1:1 initially and then multiply it by a factor of 1.01 every epoch until it reaches 1:4. This is due to the limited number of positive pairs that may cause the network to over-fit.

The inputs and outputs of the network are illustrated in Fig. 1 (for tracking phase). We use this Siamese topology, StackNet, since it outperforms the other Siamese topologies such as those combined at cost function and in-network [3]. After cropping the image patches for inputs to the network, they are resized to $224 \times 224 \times 3$ and then concatenated along the channel dimension to be used as the input to the visual-similarity CNN i.e. the input to the network becomes $224 \times 224 \times 6$. Then, the filter size of the first convolutional layer is changed from $7 \times 7 \times 3$ to $7 \times 7 \times 6$, and the last fully-connected layer models a 2-way classification problem (the same and different identities). The rest of the ResNet50 architecture remains the same except adding a dropout with a rate of 0.75 after the last pooling layer for reducing a possible over-fitting. During testing (as shown in Fig. 1), given a pair of images, the visual-similarity CNN produces the probability of the pair being the same or different identity by a forward pass.

This proposed network is trained using a binary cross-entropy loss and batch Stochastic Gradient Descent (SGD) with momentum. The mini-batch size is set to 20. We trained our model on a NVIDIA GeForce GTX 1050 GPU for 75 epochs for the first fine-tuning and for 110 epochs for the second fine-tuning, after which it generally converges, using MatConvNet [28]. We initialize the learning rate to 10^{-4} for the first 40 epochs, to 10^{-5} for the next 35 epochs and to 10^{-6} for the last 35 epochs (1 epoch is one sweep over all the training data). In addition, we augment the training samples by random flipping horizontally as well as randomly shifting the cropping positions by no more than ± 0.2 of detection box of width or height for x and y dimensions respectively to increase more variation and thus reduce possible over-fitting.

For evaluating the fine-tuned networks, we randomly sample about 800 positive pairs (the same identities) and 3200 negative pairs (different identities) from ground truth of MOT16/17-02 and MOT16/17-09 training dataset (used as a testing set for our network). We use this larger ratio of negative pairs to mimic the positive/negative distribution during tracking. We use verification accuracy as a an evaluation metric. Given a pair of images, the probability of the pair containing the same person is estimated from the network. If the estimated probabilities of positive pairs are greater than 0.5, they are assumed as correctly classified pairs. Similarly, if the estimated probabilities of negative pairs are less than 0.5, they are assumed as correctly classified pairs. Accordingly, the network fine-tuned on DukeMTMC dataset gives about 81% accuracy. More fine-tuning the network to MOT15 and MOT16/17 training dataset increases the accuracy to about 92%.

IV. Tracklet-detection Association

The GM-PHD filter distinguishes between true and false targets, however, this does not distinguish between two different targets, so an additional step is necessary to identify different targets between consecutive frames. We use both the spatio-temporal and visual similarities between the estimated tracklet and detection (filtered output) boxes in frames $k-1$ and k, respectively, to label each object across frames. Let $b^t_{i,k-1}$ be the i^{th} tracklet's estimated location and $b^d_{j,k}$ be the j^{th} detection box at frame k. Their visual similarity, $V_{s,k}(b^t_{i,k-1}, b^d_{j,k})$, is computed using the visual-similarity CNN. Similarly, their spatio-temporal similarity is calculated using Euclidean distance $D_k(b^t_{i,k-1}, b^d_{j,k})$ between their centers. We use Euclidean distance rather than Jaccard distance (1 - Intersection-over-Union) as it gives slightly better result. The spatio-temporal (motion or distance) relation has been commonly used, in different forms, in many multi-object tracking works [1] [29] [30]. Euclidean distance $D_k(b^t_{i,k-1}, b^d_{j,k})$ between the centers of the bounding boxes $b^t_{i,k-1}$ and $b^d_{j,k}$ is given by

$$\begin{aligned} &D_k(b^t_{i,k-1}, b^d_{j,k}) = \\ &\sqrt{(b^t_{i,x,k-1} - b^d_{j,x,k})^2 + (b^t_{i,y,k-1} - b^d_{j,y,k})^2}, \end{aligned} \tag{9}$$

where $(b^t_{i,x,k-1}, b^t_{i,y,k-1})$ and $(b^d_{j,x,k}, b^d_{j,y,k})$ are the center locations of their corresponding bounding boxes at frames $k-1$ and k, respectively.

We use the Munkres's variant of the Hungarian algorithm [31] to determine the optimal associations in case a detection box is tried to be associated with multiple tracklets using the following overall association cost

$$\mathbf{C}_k = (1-\eta)\mathbf{D}_{n,k} + \eta \mathbf{V}_{d,k}, \tag{10}$$

where $\mathbf{V}_{d,k} = \mathbf{1} - \mathbf{V}_{s,k}$ is the visual difference used as a cost where each of its element $V_{d,k} \in [0,1]$, $\mathbf{D}_{n,k}$ is the normalized (by maximum) version of $\mathbf{D}_k$ where each element $D_{n,k} \in [0,1]$, and η is the weight balancing the two costs. $\mathbf{C}_k \in \mathbb{R}^{N \times M}$, $\mathbf{D}_{n,k} \in \mathbb{R}^{N \times M}$ and $\mathbf{V}_{d,k} \in \mathbb{R}^{N \times M}$ are matrices where N and M are the number of tracklets and detections (filtered outputs) at time k; $\mathbf{1}$ is a matrix of $1's$ of the same dimension as $\mathbf{V}_{s,k} \in \mathbb{R}^{N \times M}$. Spatio-temporal relation gives useful information for tracklet-detection association of targets that are in close proximity, however, its importance starts to decrease as targets become (temporally) far apart. In contrast, visual similarity obtained from CNN allows long-range association as it is robust to large temporal and spatial distance. These combination of spatio-temporal and visual information helps to solve target ambiguities which may occur due to either targets motion or their visual content as well as allows long-range association of targets. The associated detection (filtered output) boxes are appended to their corresponding tracklets to generate longer ones up to time k.

V. Person re-identification for tracking

Person re-identification in the context of multi-target tracking is very challenging due to occlusions (inter-object and background), cluttered background and inaccurate bounding box localization. Inter-object occlusions are very challenging in video sequences containing dense targets, hence, object detectors may miss some targets in some consecutive frames. Re-identification of lost targets due to miss-detections is crucial to keep track of identity of each target.

The tracklet-detection association using the Hungarian algorithm given in section IV can also provide unassigned tracklet(s) and detection(s). If a past tracklet is not associated to any detection box at frame k, the tracked target might be occluded or temporally missed by the object detector. If an object detection box is not associated to any tracklet, it is used for initializing a new tracklet if it is not created earlier by checking it within the last $m = 10$ frames (from lost targets) using visual similarity $V_{s,k}(b^t_{i,k-r}, b^d_{j,k})$ for re-identification, where $r \in [2, m+1]$. We use a visual similarity threshold of 0.8 for the re-identification of targets. If the tracklet is not associated to any detection box for more than a time window of $m = 10$ frames, it is terminated. Re-identification using the visual similarity has increased the performance of the combination of the spatio-temporal and visual similarities to construct the cost for labelling of targets as shown in Fig. 2.

VI. Experimental Results

Our state vector includes the centroid positions, velocities, width and height of the bounding boxes, i.e. $x_k = [p_{cx,xk}, p_{cy,xk}, \dot{p}_{x,xk}, \dot{p}_{y,xk}, w_{xk}, h_{xk}]^T$. Similarly, the measurement is the noisy version of the target area in the image

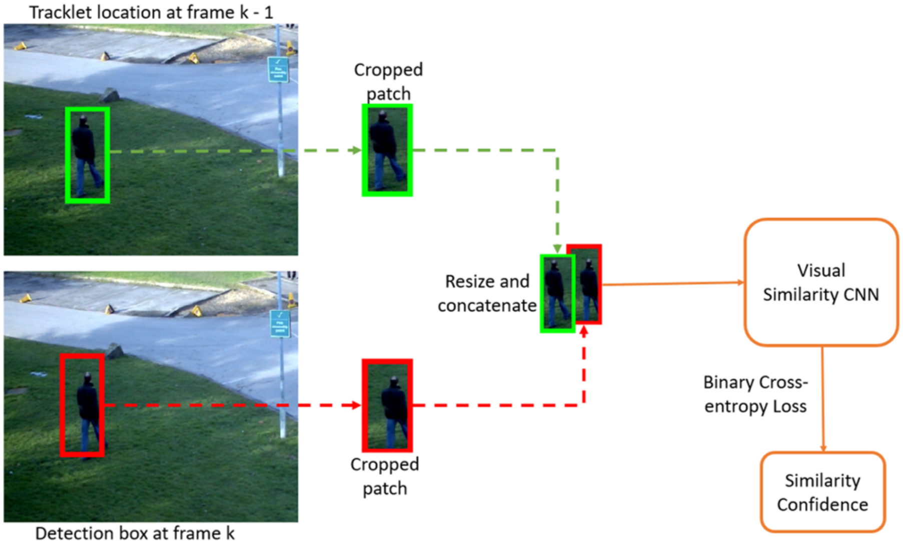

Fig. 1: Illustration of visual-similarity CNN (during tracking). The two image patches corresponding to the estimated tracklet and detection (filter output) boxes are cropped at frames $k-1$ and k, respectively.

plane approximated with a w x h rectangle centered at $(p_{cx,xk}, p_{cy,xk})$ i.e. $z_k = [p_{cx,zk}, p_{cy,zk}, w_{zk}, h_{zk}]^T$.

We set the survival probability $p_S = 0.99$, and we assume the linear Gaussian dynamic model of Eq. (1) with matrices taking into account the box width and height at the given scale.

$$F_{k-1} = \begin{bmatrix} I_2 & \Delta I_2 & 0_2 \\ 0_2 & I_2 & 0_2 \\ 0_2 & 0_2 & I_2 \end{bmatrix},$$

$$Q_{k-1} = \sigma_v^2 \begin{bmatrix} \frac{\Delta^4}{4} I_2 & \frac{\Delta^3}{2} I_2 & 0_2 \\ \frac{\Delta^3}{2} I_2 & \Delta^2 I_2 & 0_2 \\ 0_2 & 0_2 & I_2 \end{bmatrix}, \tag{11}$$

where F and Q denote the state transition matrix and process noise covariance, respectively; I_n and 0_n denote the n x n identity and zero matrices, respectively, and $\Delta = 1$ second is the sampling period defined by the time between frames. $\sigma_v = 5$ pixels/s^2 is the standard deviation of the process noise.

Similarly, the measurement follows the observation model of Eq. (2) with matrices taking into account the box width and height,

$$H_k = \begin{bmatrix} I_2 & 0_2 & 0_2 \\ 0_2 & 0_2 & I_2 \end{bmatrix},$$

$$R_k = \sigma_r^2 \begin{bmatrix} I_2 & 0_2 \\ 0_2 & I_2 \end{bmatrix}, \tag{12}$$

where H_k and R_k denote the observation matrix and the observation noise covariance, respectively, and $\sigma_r = 6$ pixels is the measurement standard deviation. The probability of detection is assumed to be constant across the state space and through time and is set to a value of $p_D = 0.95$. The false positives are independently and identically distributed (i.i.d), and the number of false positives per frame is Poisson-distributed with mean $\lambda_t = 10$ (false alarm rate of $\lambda_c = 4.8 \times 10^{-6}$; dividing the mean λ_t by frame resolution A, refer to Eq. (8)).

Nothing is known about the appearing targets before the first observation. The distribution after the observation is determined by the current measurement and zero initial velocity used as a mean of the Gaussian distribution and using a predetermined initial covariance given in Eq. (13) for birthing of targets.

$$P_{\gamma,k} = diag([100, 100, 25, 25, 20, 20]). \tag{13}$$

The birth weight $w_{\gamma,k}$ that any potential observation represents an appearing target in Eq. (3), detection score threshold s_t and whether using detection score s_k along with (multiplied by) $w_{\gamma,k}$ depends on the application as they govern the relationship between false positives and miss-detections i.e. they are hyper-parameters that require tuning. For instance, we evaluated on $w_{\gamma,k} \in \{0.10, 0.02, 0.0001\}$, $s_t \in \{0.0, 0.10, 0.15, 0, 20, 25\}$, and with and without using s_k along with $w_{\gamma,k}$. Given $w_{\gamma,k} = 0.10$ and $s_t = 0.10$, using s_k along with $w_{\gamma,k}$ ($w_{\gamma,k} = 0.10 s_k$) gives better MOTA value than without using s_k. Reducing $w_{\gamma,k}$ to 0.0001 reduces MOTA value slightly, however, it greatly decreases false positives at the expense of increased miss-detections. In our experiment, we find $w_{\gamma,k} = 0.02$, $s_t = 0.0$ and without using s_k along with $w_{\gamma,k}$ gives better MOTA value at the expense of increased false positives. The influence of s_k and s_t partly depends on the hype-parameter value of $w_{\gamma,k}$, and thus, all these hyper-parameters need to be tuned well for the application at hand. Furthermore, after evaluating on $\eta \in \{0.0, 0.4, 0.6, 1.0\}$ (in Eq. (10)), we set it to 0.6 as this gives better result. The implementation parameters and their values are summarized in Table I.

We validate our proposed tracker, GM-PHD-DAL, and compare it against state-of-the-art online and offline tracking methods (GM-PHD-HDA [2], DP-NMS [5], SMOT [32], CEM [4], JPDA-m [33], EAMTT [1], GMPHD-KCF [34] and GM-PHD [35]) on the MOT16 and MOT17 benchmark datasets [27]. We use the *public detections* provided by the MOT benchmark with a non-maximum suppression (NMS) of 0.3 for DPM detector (for both MOT16 and MOT17) and 0.5 for FRCNN and SDP detectors (for MOT17). We use the following evaluation measures:

- Multiple Object Tracking Accuracy (MOTA): A summary of overall tracking accuracy in terms of false positives, false negatives and identity switches, which gives a measure of the tracker's performance at detecting objects as well as keeping track of their trajectories.
- Multiple Object Tracking Precision [36] (MOTP): A summary of overall tracking precision in terms of bounding box overlap between ground-truth and tracked location, which shows the ability of the tracker to estimate precise object positions.
- Mostly Tracked targets (MT): Percentage of mostly tracked targets (a target is tracked for at least 80% of its life span regardless of maintaining its identity) to the total number of ground truth trajectories.
- Mostly Lost targets (ML) [37]: Percentage of mostly lost targets (a target is tracked for less than 20% of its life span) to the total number of ground truth trajectories.
- False Positives (FP): Number of false detections.
- False Negatives (FN): Number of miss-detections.
- Identity Switches (IDSw): Number of times the given identity of a ground-truth track changes.
- Fragmented trajectories (Frag): Number of times a track is interrupted (compared to ground truth trajectory) due to miss-detection.

True positives are detections which have at least 50% overlap with their corresponding ground truth bounding boxes. For more detailed description of each metric, please refer to [27].

Quantitative evaluation of our proposed method with other trackers is compared in Table II on MOT16 benchmark dataset. The Table shows that our algorithm outperforms both online and offline trackers listed in the table in terms of MOTA, MOTP and FP; our algorithm provides the 4^{th} lowest FP of all listed algorithms on MOT16 benchmark website. The number of MT percentage is overall higher than many of the online and offline trackers except one offline tracker (i.e. second to CEM). The number of ML and FN percentage are also lower than many of the online and offline trackers (i.e. second to SMOT). Our tracker also gives promising results on MOT17 benchmark dataset as is quantitatively shown in Table III. It outperforms all other trackers in the table in all MOTA, MOTP, MT, ML and FN measures. The number of FP percentage is also lower than many of the online and offline trackers (i.e. second to DP-NMS). The number of IDSw and Frag is still significant compared to the other online and offline trackers which is due to the fact that spawning targets are not modelled in our tracker, therefore, identity switches are more likely to occur in such crowded scenes. In this work, the spawned targets are treated as new-born targets.

The most important to notice here is that the comparison of our algorithm to the GM-PHD-HDA [2]. These both trackers use the GM-PHD filter but with different approaches for labelling of targets from frame to frame. While our tracker uses the Hungarian algorithm for labelling of targets by postprocessing the output of the filter using a combination of spatio-temporal and visual similarities along with visual similarity for re-identification, the GM-PHD-HDA uses the approach in [38] at the prediction stage by also including appearance features for re-identification to label targets. In addition to the GM-PHD-HDA tracker, our proposed tracker outperforms the other GM-PHD filter-based trackers such as GMPHD-KCF and GM-PHD as shown in Tables II and III in all of the evaluation metrics except IDSw and Frag.

The qualitative comparison of our proposed tracker (GM-PHD-DAL) and our tracker without re-identification is given in Fig. 2 for frames 4, 17 and 36. Due to the detection failures, some labels of targets are not consistent for our tracker without re-identification (top row), for instance, label 5 in frame 4 is changed to label 9 in frame 17; label 10 in frame 17 is changed to label 14 in frame 36; label 2 in frame 17 is changed to label 11 in frame 36. However, the labels of the targets are consistent using the GM-PHD-DAL tracker (bottom row).

In our evaluations, the association cost constructed using only visual similarity CNN gives better result than using only spatio-temporal relation, however, their combination using Eq. (10) gives better result than each of them. Furthermore, weighted summation of the costs according to Eq. (10) gives slightly better result than the Hadamard product (element-wise multiplication) of the two costs. Our proposed tracking algorithm is implemented in Matlab (not well optimized) on a i7 2.80 GHz core processor with 8 GB RAM. We use the MatConvNet [28] for CNN feature extraction where its forward propagation computation is transferred to a NVIDIA GeForce GTX 1050 GPU, and our tracker runs at about 3.5 frames per second (fps). The forward propagation for feature extraction step is the main computational load of our tracking algorithm, specially for constructing the cost due to the visual content in Eq. (10). For very time-critical applications, the spatio-temporal relation-based data association with the visual-similarity based re-identification algorithm might be an option as this is much faster with a little loss of performance.

VII. Conclusions

We have developed a novel multi-target visual tracker based on the GM-PHD filter and the similarity deep CNN. We apply this method for tracking multiple targets in video sequences acquired under varying environmental conditions and targets density. We followed a tracking-by-detection approach using the public detections provided in the MOT16 and MOT17 benchmark datasets. We integrate spatio-temporal similarity from the object bounding boxes and the appearance information from the learned similarity CNN (using both motion and

TABLE I: Implementation values of the parameters used in our experiment.

Parameters	η	s_t	$w_{\gamma,k}$	σ_r	σ_v	p_D	p_S	λ_t	U	T	m
Values	0.6	0.0	0.02	6 pixels	5 pixels$/s^2$	0.95	0.99	10	4 pixels	10^{-5}	10 frames

TABLE II: Tracking performance of representative trackers developed using both online and offline methods. All trackers are evaluated on the test dataset of the **MOT16** [27] benchmark using public detections. The first and second highest values are highlighted by bold and underline, respectively. Evaluation measures with (↑) show that higher is better, and with (↓) denote lower is better.

Tracker	Mode	MOTA↑	MOTP↑	MT (%)↑	ML (%)↓	FP↓	FN↓	IDS↓	Frag↓
CEM [4]	offline	$\underline{33.2}$	75.8	**7.8**	54.4	6,837	114,322	642	$\underline{731}$
DP-NMS [5]	offline	26.2	$\underline{76.3}$	4.1	67.5	$\underline{3,689}$	130,557	**365**	**638**
SMOT [32]	offline	29.7	75.2	5.3	**47.7**	17,426	**107,552**	3,108	4,483
JPDF-m [33]	offline	26.2	$\underline{76.3}$	4.1	67.5	$\underline{3,689}$	130,549	**365**	**638**
GM-PHD-HDA [2]	**online**	30.5	75.4	4.6	59.7	$\underline{5,169}$	120,970	$\underline{539}$	$\underline{731}$
GM-PHD-DAL (ours)	**online**	**35.1**	**76.6**	$\underline{7.0}$	$\underline{51.4}$	**2,350**	$\underline{111,886}$	4,047	5,338

TABLE III: Tracking performance of representative trackers developed using both online and offline methods. All trackers are evaluated on the test dataset of the **MOT17** benchmark using public detections. The first and second highest values are highlighted by bold and underline, respectively. Evaluation measures with (↑) show that higher is better, and with (↓) denote lower is better.

Tracker	Mode	MOTA↑	MOTP↑	MT (%)↑	ML (%)↓	FP↓	FN↓	IDS↓	Frag↓
DP-NMS [5]	offline	$\underline{43.7}$	$\underline{76.9}$	12.6	46.5	**10,048**	302,728	$\underline{4,942}$	**5,342**
EAMTT [1]	**online**	42.6	76.0	$\underline{12.7}$	$\underline{42.7}$	30,711	288,474	**4,488**	$\underline{5,720}$
GMPHD-KCF [34]	**online**	40.3	75.4	8.6	43.1	47,056	$\underline{283,923}$	5,734	7,576
GM-PHD [35]	**online**	36.2	76.1	4.2	56.6	23,682	328,526	8,025	11,972
GM-PHD-DAL (ours)	**online**	**44.4**	**77.4**	**14.9**	**39.4**	$\underline{19,170}$	**283,380**	11,137	13,900

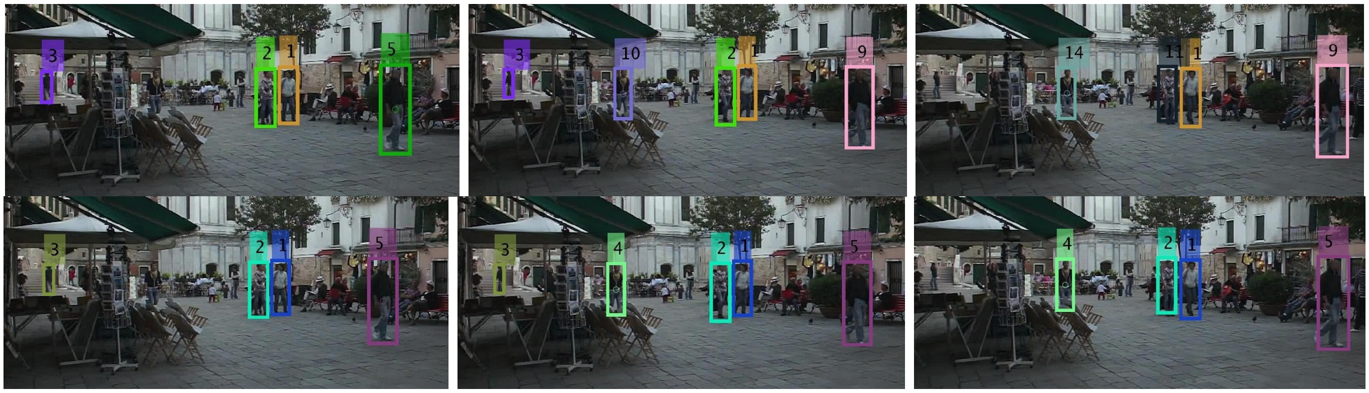

Fig. 2: Sample results on 3 frames of MOT16-01 dataset for our proposed tracker (GM-PHD-DAL) without re-identification (top row for frames 4, 17 and 36 from left to right) and with re-identification (bottom row for frames 4, 17 and 36 from left to right). Bounding boxes represent the tracking results with their color-coded identities; small numbers are also shown on top of each bounding box for better clarity.

appearance cues) for the labelling of each target in consecutive frames. Results show that our method outperforms state-of-the-art trackers developed using both online and offline approaches on the MOT16 and MOT17 benchmark datasets in terms of tracking accuracy and precision. In the future work, we will include inter-object relations model for tackling the interactions of different objects.

References

[1] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, 2016, pp. 84–99.

[2] Y. Song and M. Jeon, "Online multiple object tracking with the hierarchically adopted GM-PHD filter using motion and appearance," in *IEEE/IEIE The International Conference on Consumer Electronics (ICCE) Asia*, 2016.

[3] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR). DeepVision: Deep Learning for Computer Vision.*, 2016.

[4] 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, Jan 2014.

[5] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *CVPR 2011*, June 2011, pp. 1201–1208.

[6] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 560–576, 2001.

[7] T.-J. Cham and J. M. Rehg, "A multiple hypothesis approach to figure tracking." in *CVPR*. IEEE Computer Society, 1999, pp. 2239–2245.

[8] R. P. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[9] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4091–4104, Nov 2006.

[10] N. L. Baisa, D. Bhowmik, and A. Wallace, "Long-term correlation tracking using multi-layer hybrid features in sparse and dense environments," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 464 – 476, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1047320318301536

[11] N. L. Baisa, "Single to multiple target, multiple type visual tracking," Ph.D. dissertation, Heriot-Watt University, 06 2018.

[12] N. L. Baisa and A. Wallace, "Multiple target, multiple type filtering in the RFS framework," *Digital Signal Processing*, vol. 89, pp. 49 – 59, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200418303166

[13] ——, "Development of a N-type GM-PHD filter for multiple target, multiple type visual tracking," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 257 – 271, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1047320319300343

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1097–1105.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[17] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016.

[18] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned CNN embedding for person re-identification," *CoRR*, vol. abs/1611.05666, 2016.

[20] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3074–3082.

[19] B. Lavi, M. F. Serj, and I. Ullah, "Survey on deep learning techniques for person re-identification task," *CoRR*, vol. abs/1807.05284, 2018. [Online]. Available: http://arxiv.org/abs/1807.05284

[21] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," *CoRR*, vol. abs/1606.09549, 2016.

[22] G. Welch and G. Bishop, "An introduction to the kalman filter," 2006.

[23] B. Ristic, D. E. Clark, B.-N. Vo, and B.-T. Vo, "Adaptive target birth intensity for PHD and CPHD filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1656–1668, 2012.

[24] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 248–255.

[25] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[26] L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *CoRR*, vol. abs/1504.01942, 2015. [Online]. Available: http://arxiv.org/abs/1504.01942

[27] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831 [cs]*, Mar. 2016, arXiv: 1603.00831.

[28] A. Vedaldi and K. Lenc, "MatConvNet – convolutional neural networks for matlab," in *Proceedings of the 25th annual ACM international conference on Multimedia*, 2015.

[29] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3701–3710.

[30] 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 *ECCV Workshops*, 2016.

[31] F. Bourgeois and J.-C. Lassalle, "An extension of the munkres algorithm for the assignment problem to rectangular matrices," *Commun. ACM*, vol. 14, no. 12, pp. 802–804, Dec. 1971.

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

[33] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3047–3055.

[34] T. Kutschbach, E. Bochinski, V. Eiselein, and T. Sikora, "Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2017, pp. 1–5.

[35] V. Eiselein, D. Arp, M. Pätzold, and T. Sikora, "Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors," in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, Sep. 2012, pp. 325–330.

[36] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, Feb 2009.

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

[38] K. Panta, D. E. Clark, and B.-N. Vo, "Data association and track management for the gaussian mixture probability hypothesis density filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1003–1016, 2009.