Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

ARTICLE TYPE

Transfer Learning Based Online Multi-Person Tracking with Gaussian Process Regression

Baobing Zhang[1] | Siguang Li*[2,3] | Zhengwen Huang[1] | Babak H. Rahi[1] | Qicong wang[4]

[1]Department of Electronic and Computer Engineering, Brunel University London, Uxbridge ,UB8 3PH, UK
[2]The Key Laboratory of Embedded Systems and Service Computing, Tongji University, Shanghai, China
[3]College of Electrical Engineering, Binzhou University, Binzhou, Shandong, China
[4]Department of Computer Science, Xiamen University, Xiamen, China

Abstract

Most existing tracking-by-detection approaches are affected by abrupt pedestrian pose changes, lighting conditions, scale changes, real-time processing, which leads to issues such as detection errors and drifts. To deal with these issues, we present a novel multi-person tracking framework by introducing a new observation model, which learns in a semi-supervised manner. The background information is taken into consideration to build the discriminative tracker, training samples are re-weighted appropriately to ease the impact of the potential sample misalignment and noisy during model updating. Unlabeled samples from the current frame provide rich information which is used for enhancing the tracking inference. Experimental results show that the proposed approach outperforms a number of state-of-the-art methods on some benchmark date sets.

KEYWORDS:
Multi-person tracking; Tracking-by-detection; Semi-supervised; Prior information; Kalman filter; Transfer learning; Regression

1 | INTRODUCTION

There are many high resolution cameras installed every day in different parts of the world for surveillance. This provides us with a large amount of image data. The demand for algorithms to automatically process such image data has surged. One particular research area is to deal with multi-person tracking. Even though this area has been intensively studied [1,2,3,4], robust and efficient tracking of multiple persons still remains unsolved taking into account pedestrian occlusions, dynamic background changing and real-time processing. Building on the tremendous progress of object detection, the tracking-by-detection paradigm has been widely used for multi-person tracking in recent years. Compared with background modeling-based trackers tracking-by-detection approaches are more robust to changing backgrounds. However, the object detector, which is used for tracking, usually yields false positive and missing detections making the association between targets and detections difficult to build. In addition, these methods always build trajectories based on two neighboring frames during a long-term of occlusion or abrupt human pose changes. As a result, the risk of the tracked target trends to drift increases.

To address these challenges, many tracking-by-detection approaches combine the dynamic and observation models. A dynamic model like a Kalman filter [5] or a particle filter [6], takes pedestrian behavior into account for estimating the current state of pedestrian and improves the data association. However, most existing dynamic models utilize only the previous one state for predicting without considering prior information leading to an incorrect estimation when the pedestrian walking direction changes abruptly. An observation model takes into account the pedestrian's appearance changes, particularly when an observation model updates adaptively in a causal way. Most observation models utilize the past appearance information over time but do not consider the current state information of the pedestrian's appearance leading to drifting problems.

In this paper we explore how the future information can be used to assist in generating a tracking decision. To achieve this goal, we employ a novel observation model which learns in a semi-supervised fashion. The current state of the tracking targets has a significant

influence on the final tracking decision. Because a dynamic model utilizes only the previous one state to estimate the current state of the tracking targets which tends to drift, we fuse the prior information to enhance tracking inference, thereby to alleviate drift. The new observation is update adaptively to avoid the loss of sample diversity. All of these features of the new observation model help alleviate the problem of drifting. The main contributions of the paper are as follows.

- It takes into account the background information in the tracking inference, which is more stable for dynamic background tracking.
- To deal with object occlusions, it fuses all the prior information for tracking decisions, rather than utilizes only the previous one state of the tracking targets.
- It features online transfer learning by utilizing the extracted knowledge from the current state of the tracking targets for generating tracking decisions.
- Experimental results show the proposed observation model outperforms a number of state-of-the-art methods on some benchmark data sets.

The rest of the paper is organized as follows. Section 2 give a review on related work. Section 3 details the design and implementation the proposed observation model for online multi-person tracking. Section 4 evaluates the performance of the observation model in comparison with a number of typical methods. Section 5 concludes the paper and points out some future work.

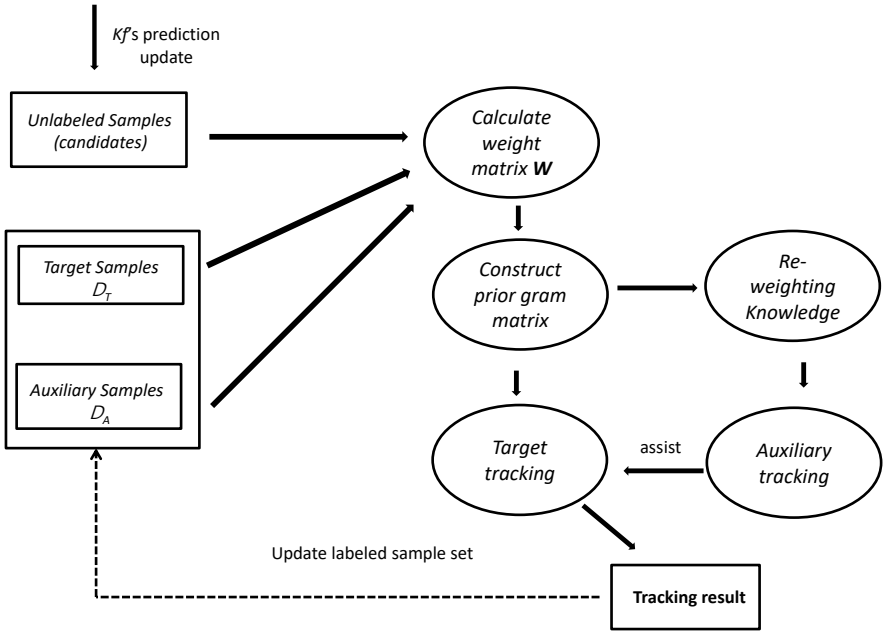

FIGURE 1 Overview of the Observation model for tracking

2 | RELATED WORK

Previous works such as a Kalman filter [5] and recently a particle filter [6] take into account only the previous one state information for current state estimation and optimize each trajectory independently. Many tracking-by-detection methods employ background subtractions from one or more cameras [7,8]. Based on the progress in the field of object detection, many researchers combine tracking and detection [9,10], which links the detections into trajectories. While other tracking-by-detection methods rely on the mean-shift tracker [11] which finds in the

detections that best match to the target appearance by computing a weight image via gradient ascent procedure. The work in [1] employs a weak classifier to differentiate the background foreground pixels and a strong classifier to generate a confidence map which is normally utilized by the mean-shift model in finding a target. Moreover the work in [12] proposes a part-based representation method by using head tracking instead of body tracking to handle the partial occlusion problems. An effective sampling method is proposed by [13] and applied to visual tracking. The observation model in these methods fully relies on the detected objects. Different from these methods, the novel observation model presented in this paper learned in a semi-supervised manner. More importantly, the model not only utilizes the past image patch information for tracking but also considers the current frame data. As a result, our work reduces the potential risk of drifting. Matching the newly detection to the established object is closely related to the data association problem [14] and classical approaches include [15,16,17]. The work in [15] considers several possible associations over multiple stages, but its computation complexity usually makes it unsuitable for multi-step analysis. The Hungarian algorithm [17] can be used to find the maximum matching of the detection-tracker pairs in a bipartite graph at runtime. In our work, we employ the Hungarian algorithm to deal with data association problems.

3 | TRACKING BY OBSERVATION MODEL

This section presents the design details of the multi-person framework, especially the observation model as shown in Fig.1. We employ the new observation model proposed in [18], but we make several changes in order to better suit the multi-person tracking problem. At each frame I_{t}, we obtain $D_{\mathrm{t}} = \left\{d_{\mathrm{t}}^{\mathrm{i}}\right\}_{i=1}^{N}$ detections by using Dalal and Triggs' Histogram of Oriented Gradients (HOG) detector, which is one of the most successful detectors especially for pedestrian detection [19]. Here d_t^i refers to a bounding box at each frame I_{t}. A tracker T_{i} is defined as $\{kf_{\mathrm{t}}, X_{\mathrm{t}}, M_{\mathrm{t}}\}$, where kf_{t} is the Kalman filter used to model the tracker's dynamics. Tracker's center point position and the 2D velocity are used to define Kalman filter's state (x, y, dx, dy). We set process and measurement noise covariance matrices to $\left(\frac{w_{\mathrm{t}}^{\mathrm{i}}}{r_{\mathrm{f}}}\right)^2 \bullet diag\left(0.025, 0.025, 0.25, 0.25\right)$ and $\left(w_{\mathrm{t}}^{\mathrm{i}}\right)^2 \bullet diag\left(1, 1\right)$ respectively, here r_{f} refers to the frame rate of the input sequence, $X_{\mathrm{t}} = \left[x_{\mathrm{t}}^{\mathrm{i}}, y_{\mathrm{t}}^{\mathrm{i}}, w_{\mathrm{t}}^{\mathrm{i}}, h_{\mathrm{t}}^{\mathrm{i}}\right]$ is a bounding box which store the current state of the target estimated by the observation model at $\left(x_{\mathrm{t}}^{\mathrm{i}}, y_{\mathrm{t}}^{\mathrm{i}}\right)$ with a size of $\left(w_{\mathrm{t}}^{\mathrm{i}}, h_{\mathrm{t}}^{\mathrm{i}}\right)$. M_{t} is a set of templates collected through time used for tracker's maintenance.

3.1 | Gaussian Process Regression

From the perspective of Bayesian incremental learning for visual tracking, once the tracker is initialized, the state variable $\mathbf{X}_{\mathrm{t}}^{T_{\mathrm{i}}}$ which describes the location of a tracker T_{i} at time t can be inferred recursively:

$$p\left(\mathbf{X}_t^{T_i}|\mathcal{I}_t^{T_i}\right) \propto p\left(\mathbf{I}_t|\mathbf{X}_t^{T_i}\right) \int p\left(\mathbf{X}_t^{T_i}|\mathbf{X}_{t-1}^{T_i}\right) p\left(\mathbf{X}_{t-1}^{T_i}|\mathcal{I}_{t-1}^{T_i}\right) d\mathbf{X}_{t-1}^{T_i} \tag{1}$$

where $\mathcal{I}_{\mathrm{t}}^{T_{\mathrm{i}}} = \{\mathbf{I}_1, \cdots, \mathbf{I}_{\mathrm{t}}\}$ is a set of observed images up to the t-th frame of the tracker T_{i}, the distribution of the object location in the current frame $\mathcal{X}_{\mathrm{U}}^{T_{\mathrm{i}}} = \left\{\mathbf{X}_{\mathrm{t}}^{T_{\mathrm{i}},\mathrm{j}}, j = 1, 2, \cdots, n_{\mathrm{U}}\right\}$ is stochastically generated with Kalman filter's prediction as input , which we call n_{U} the tracking candidates of the tracker T_{i}. The tracking result of tracker T_{i} can be estimated by MAP:

$$\hat{\mathbf{X}_t^{\mathrm{T}_i}} = \underset{\mathbf{X}_t^{\mathrm{T}_i,j}}{\arg\max}\, \mathrm{P}\left(\mathbf{X}_t^{\mathrm{T}_i,j}|\mathcal{I}_t^{T_i}\right) \tag{2}$$

For each sample, we introduce an indicator variable $\mathrm{y_j} \in \{-1, +1\}$ to indicate a positive sample ($\mathrm{y_j} = +1$) or a negative sample ($\mathrm{y_j} = -1$) corresponding to $\mathbf{X}_{\mathrm{t}}^{T_{\mathrm{i}},\mathrm{j}}$. $\mathcal{X}_{\mathrm{U}}^{T_{\mathrm{i}}}$ is an unlabeled sample set of tracker T_{i} from the tracking result $\left\{\hat{\mathbf{X}_f^{T_{\mathrm{i}}}}, f = 1, 2, \cdots, t-1\right\}$ up to the (t-1)-th frame. For each tracker, we extract n_{L} labeled training samples with indicator variables, and then we divide the n_{L} labeled training samples into two groups - one is the target sample set including n_{T} samples gathered from the most recent frame denoted as $\mathcal{D}_{\mathrm{T}} = \left\{\left(\mathbf{X}^{\mathrm{j}}, y_{\mathrm{j}}\right), \mathrm{j} = 1, 2, \cdots, n_{\mathrm{T}}\right\}$ which is updated quickly and aggressively. The other auxiliary sample group collected at every few intervals denoted as $\mathcal{D}_{\mathrm{A}} = \left\{\left(\mathbf{X}^{\mathrm{j}}, y_{\mathrm{j}}\right), \mathrm{j} = n_{\mathrm{T}} + 1, n_{\mathrm{T}} + 2, \cdots, n_{\mathrm{T}} + n_{\mathrm{A}}\right\}$ which is updated slowly and carefully, where $n_{\mathrm{L}} = n_{\mathrm{T}} + n_{\mathrm{A}}$, and $\mathrm{y_j}$ refers to a label. Let $\mathbf{1} = [+1, +1, \cdots, +1]^\top$, $y_{\mathrm{U}} = [y_1, y_2, y_3, \cdots, y_{n_{\mathrm{U}}}]^\top$, then the regression function for the indicators of unlabeled samples y_{U} can be expressed as:

$$\mathcal{R} = P(y_U = 1|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T) \tag{3}$$

For each sample we have indicator variable $\mathrm{y_j} \in \{-1, +1\}$. Actually each prediction of a Gaussian process on an unlabeled sample set $\mathcal{X}_{\mathrm{U}}^{T_{\mathrm{i}}} = \left\{\mathbf{X}_{\mathrm{t}}^{T_{\mathrm{i}},\mathrm{j}}, j = 1, 2, \cdots, n_{\mathrm{U}}\right\}$ is a real-valued output of a mean vector in a fixed feature space $\mathcal{K}$, i.e. the distance to the hyperplane with the normal vector [20]. For the purpose of using Gaussian process regression for classification, we introduce two real valued latent vectors $l_{\mathrm{A}} \in \mathbb{R}^{n_{\mathrm{A}}}$ and $l_{\mathrm{U}} \in \mathbb{R}^{n_{\mathrm{U}}}$ corresponding to the label y_{A} and y_{U} respectively. Intuitively,the further away an unlabeled sample is from

the hyperplane (i.e. the larger the value of l), the more likely it is that the sample is from the class $y = \mathbf{sign}(l)$.We model this intuitive notion by

$$\mathrm{P}(y_i|l_i) = \frac{e^{\gamma l_i y_i}}{e^{\gamma l_i y_i} + e^{-\gamma l_i y_i}} = \frac{1}{1 + e^{-2\gamma l_i y_i}}, \qquad \forall_i = 1, 2, \ldots, \mathrm{n}_U \tag{4}$$

where $l_{\mathsf{u}} = [l_1, l_2, \cdots, l_{\mathsf{n_u}}]\top$, γ is the noise level of the sigmoid noise label output model. We set γ to 10 and for auxiliary samples we use the same label output model. First of all we feed the auxiliary data to the Gaussian process model, then get the corresponding latent real-valued l_{A} which is the output of Gaussian process regression. Furthermore, by using the sigmoid noise label generation model, we get the indicator label y_{A}. We construct the prior covariance matrix depending on all the samples, the correlated structure of the labeled samples and unlabeled samples has a significant effect on the latent real-valued output. The latent variable l_{A} is the re-weighted knowledge extracted from the regression which can be a soft replacement of the indicator label y_{A} The latent variable is better for ameliorating sample misalignment problems, and is less sensitive to noise comparing with the indicator variable.

3.2 | Gaussian Latent Variable Model

In order to exploit latent variables l_{U} and l_{A}, we marginalize over all their possible values $(l_{\mathsf{U}}, l_{\mathsf{A}})$:

$$\begin{aligned} P(y_{\mathsf{U}} = 1|\mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}}) &= \int\int P(y_{\mathsf{U}} = 1, l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}})\, dl_{\mathsf{A}}\, dl_{\mathsf{U}} \\ &= \int\int P(y_{\mathsf{U}} = 1|l_{\mathsf{U}}) P(l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}})\, dl_{\mathsf{A}}\, dl_{\mathsf{U}} \end{aligned} \tag{5}$$

The latent variable l and labels $y \in (-1, +1)$ are connected via the sigmoid noise label output model, applying the posterior $P(l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}})$ by Bayes theorem we have

$$P(l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}}) = P(l_{\mathsf{A}}, l_{\mathsf{U}}|y_{\mathsf{A}}, \mathcal{X}_{\mathsf{U}}, \mathcal{X}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}}) \tag{6}$$

$$= \frac{P(y_{\mathsf{A}}|l_{\mathsf{U}}, l_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}}, \mathcal{X}_{\mathsf{A}}, \mathcal{X}_{\mathsf{U}}) \bullet P(l_{\mathsf{U}}, l_{\mathsf{A}}|\mathcal{D}_{\mathsf{T}}, \mathcal{X}_{\mathsf{A}}, \mathcal{X}_{\mathsf{U}})}{P(y_{\mathsf{A}}|\mathcal{D}_{\mathsf{T}}, \mathcal{X}_{\mathsf{A}}, \mathcal{X}_{\mathsf{U}})} \tag{7}$$

The Gaussian process model $P(l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{A}}, \mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{T}})$ is restricted to the auxiliary data and unlabeled data with the mode μ and the covariance matrix $\tilde{\mathbf{\Delta}}^{-1} \in \mathbb{R}^{(n_{\mathsf{A}}+n_{\mathsf{U}})\times(n_{\mathsf{A}}+n_{\mathsf{U}})}$ leading to the regression of the latent variables l_{A} and l_{U} :

$$P(l_A, l_U|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T) \sim \mathcal{N}(\mu, \tilde{\Delta}^{-1}) \tag{8}$$

3.3 | Constructing Prior Gram Matrix

To prevent the binary classification $y \in \{-1, +1\}$ from losing generality, we need to define the kernel matrix $\mathbf{G}_{\mathsf{all}}$ properly which has to be symmetric with non-negative entries. The Gram matrix $\mathbf{G}_{\mathsf{all}}$ is based on the weighted graph $\mathcal{G} = (V, E)$ with node set V corresponding to all the samples includes target samples, auxiliary samples and unlabeled samples. Intuitively the weighted matrix W of $\mathcal{G}$ specifies the 'local similarity' .

Following this intuition, we use the method proposed in [21] to construct the weighted matrix. We further explore the manifold structure between all the samples as suggested in [22]. Gaussian random fields are equivalent to Gaussian processes that are restricted to a finite set of points [23]. Following this connection, we establish the link between the graph Laplacian and kernel method in general. We compute sparse matrix W here, which empirically tends to have a good performance. Eq. (8) can be viewed as a Gaussian process restricted to the auxiliary and unlabeled data. The Laplacian is defined as $\mathbf{\Delta} \equiv \mathbf{D} - W$, degree matrix $\mathbf{D}$ is the row sum of W. Because the Laplacian $\mathbf{\Delta}$ has a zero eigenvalue with constant eigenvector $\mathbf{1}$ and as covariance matrix is an improper prior. To get rid of the eigenvalues, a regular Laplacian is obtained by $\tilde{\mathbf{\Delta}} = \mathbf{\Delta} + \mathbf{I}/\tau^2$, where τ is a small smoothing parameter, $\tilde{\mathbf{\Delta}}^{-1}$ is the inverse function of the Laplacian $\tilde{\mathbf{\Delta}}$. Therefore the covariance between any two points i , j in general depends on all data. This is the way how semi-supervised learning ~~working~~ and how unlabeled data influences the Prior Knowledge can be viewed as a transfer learning strategy.

3.4 | Laplace Approximation for Gaussian Processes

We use $\mathbf{G}$=$\tilde{\mathbf{\Delta}}^{-1}$ to denote the covariance matrix (the gram matrix). Considering the sigmoid noise label output model, the $P(l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}})$ is no longer a Gaussian and has no closed form solution, assuming $P(l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{A}}, \mathcal{D}_{\mathsf{T}})$ is a uni-modal function with mode $(l_{\mathsf{A}}, l_{\mathsf{U}}) \in \mathbb{R}^{\mathsf{n_A}+\mathsf{n_U}}$. We use its Laplace approximation with the mode $\mu' \in \mathbb{R}^{\mathsf{n_A}+\mathsf{n_U}}$ and covariance $\mathbf{\Sigma} \in \mathbb{R}^{(n_{\mathsf{A}}+n_{\mathsf{U}})\times(n_{\mathsf{A}}+n_{\mathsf{U}})}$ instead the correct density.

~~Taking the logarithm will not change the maximum but render optimization easier ,take~~ the logarithm of Eq.(7) ,we have the following objective function to maximize

$$\mathcal{J}(l_A, l_U) = \underbrace{\ln(\mathrm{P}(y_{\mathsf{A}}|l_{\mathsf{A}}))}_{\mathcal{Q}_1(l_A)} + \underbrace{\ln(\mathrm{P}(l_{\mathsf{A}}, l_{\mathsf{U}}|\mathcal{X}_{\mathsf{A}}, \mathcal{X}_{\mathsf{U}}, \mathcal{D}_{\mathsf{T}}))}_{\mathcal{Q}_2(l_A, l_U)} - \ln(\mathrm{P}(l_A|\mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T)) \tag{9}$$

The last term is normalization constant has no influence on the maximization can be omitted from the optimization.

Let's focus first on $\mathcal{Q}_2(l_{\mathsf{A}}, l_{\mathsf{U}})$, which builds the link between Gaussian processes regression and classification. According to Eq. (8) this term is given by

$$Q_2(l_A, l_U) = -\frac{1}{2}(\ln(2\pi)^{n_A+n_U} + \ln|\mathbf{G}| + (l-\mu)^\top \mathbf{G}^{-1}(l-\mu)) \tag{10}$$

We define $l^\top = (l_{\mathsf{A}}^\top l_{\mathsf{U}}^\top)$, $\mathbf{y}^\top = (\mathbf{y}_{\mathsf{T}}^\top l_{\mathsf{A}}^\top)$, $\mathbf{y}_{\mathsf{T}} = [\mathsf{y}_1, \mathsf{y}_2, ..., \mathsf{y}_{\mathsf{n}_\mathsf{T}}]$. Moreover $\mathbf{G}_{\mathsf{all}} = \begin{pmatrix} \mathbf{G}_{\mathsf{LL}} & \mathbf{G}_{\mathsf{LU}} \\ \mathbf{G}_{\mathsf{UL}} & \mathbf{G}_{\mathsf{UU}} \end{pmatrix} = \begin{pmatrix} \mathbf{G}_{\mathsf{TT}} & \mathbf{G}_{\mathsf{TZ}} \\ \mathbf{G}_{\mathsf{ZT}} & \mathbf{G}_{\mathsf{ZZ}} \end{pmatrix}$ is $(\mathsf{n}_\mathsf{L} + \mathsf{n}_\mathsf{U}) \times (\mathsf{n}_\mathsf{L} + \mathsf{n}_\mathsf{U})$ Gram matrix (symmetric, non-singular), which is defined over all samples. Its inverse is $\mathbf{G}_{\mathsf{all}}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{M} \end{pmatrix}$ (according to the Partitioned Matrix Inversion Theorem) derivate form the latter one. Where $\mathbf{A} = \mathbf{G}_{\mathsf{TT}}^{-1} + \mathbf{G}_{\mathsf{TT}}^{-1}\mathbf{G}_{\mathsf{TZ}}\mathbf{M}\mathbf{G}_{\mathsf{ZT}}\mathbf{G}_{\mathsf{TT}}^{-1}$, $\mathbf{B} = -\mathbf{G}_{\mathsf{TT}}^{-1}\mathbf{G}_{\mathsf{TZ}}\mathbf{M}$, $\mathbf{M} = (\mathbf{G}_{\mathsf{ZZ}} - \mathbf{G}_{\mathsf{ZT}}\mathbf{G}_{\mathsf{TT}}^{-1}\mathbf{G}_{\mathsf{TZ}})^{-1}$, according to[20,23], we can determine μ and $\mathbf{G}$ in eq.(8) as : $\mu = -\mathbf{M}^{-1}\mathbf{B}^\top\mathbf{y}_{\mathsf{T}}$, $\mathbf{G} = \mathbf{M}^{-1}$. Hence the equation (10) can ~~derivate~~ as follows :

$$\begin{aligned}
\mathsf{Q}_2(l_{\mathsf{A}}, l_{\mathsf{U}}) &= -\frac{1}{2}(\ln(2\pi)^{\mathsf{n}_\mathsf{A}+\mathsf{n}_\mathsf{U}} + \ln|\mathbf{G}| + (l-\mu)^\top\mathbf{G}^{-1}(l-\mu)) \\
&= -\frac{1}{2}(\ln|\mathbf{G}_{\mathrm{all}}| + \mathbf{y}_{\mathsf{T}}^\top\mathbf{A}\mathbf{y}_{\mathsf{T}} + l^\top\mathbf{B}^\top\mathbf{y}_{\mathsf{T}} + \mathbf{y}_{\mathsf{T}}^\top\mathbf{B}l + l^\top\mathbf{M}l) + \mathsf{c}_1 \\
&= -\frac{1}{2}(\ln|\mathbf{G}_{\mathrm{all}}| + (\mathbf{y}_{\mathsf{T}}^\top l^\top)\mathbf{G}_{\mathrm{all}}^{-1}\begin{pmatrix}\mathbf{y}_{\mathsf{T}} \\ l\end{pmatrix}) + \mathsf{c}_1 \qquad (11) \\
&= -\frac{1}{2}(\ln|\mathbf{G}_{\mathrm{all}}| + (\mathbf{y}^\top l_{\mathsf{U}}^\top)\mathbf{G}_{\mathrm{all}}^{-1}\begin{pmatrix}\mathbf{y} \\ l_U\end{pmatrix}) + \mathsf{c}_1 \qquad (12)
\end{aligned}$$

where $\mathsf{c}_1 = -\frac{1}{2}(\mathbf{y}_{\mathsf{T}}^\top(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^\top - \mathbf{A})\mathbf{y}_{\mathsf{T}} + \mathbf{ln}|\mathbf{G}| - \mathbf{ln}|\mathbf{G}_{\mathrm{all}}| + \mathbf{ln}(2\pi)^{\mathsf{n}_\mathsf{A}+\mathsf{n}_\mathsf{U}})$ summarizes all terms independent of l, we can see that $\mathsf{Q}_1(l_{\mathsf{A}})$ does not depend on l_{U}. Thus we can derive the optimal value $\hat{l_{\mathsf{U}}}$ of l_{U} by maximizing $\mathsf{Q}_2(l_{\mathsf{A}}, \bullet)$, taking the derivative of $\mathsf{Q}_2(l_{\mathsf{A}}, \bullet)$ w.r.t. l_{U}, setting this function to zero. According to[23] the optimal value $\hat{l_{\mathsf{U}}}$ can be derived as:

$$\hat{l_U} = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathbf{y}_{\mathsf{T}} \\ \hat{l_{\mathsf{A}}}\end{pmatrix} = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\mathbf{y} \tag{13}$$

Substituting this expression into eq. (12) shows that this term equals

$$\begin{aligned}
\mathsf{Q}_2(l_{\mathsf{A}}, l_{\mathsf{U}}) &= -\frac{1}{2}(\mathbf{y}_{\mathsf{T}}^\top l_{\mathsf{A}}^\top)\mathbf{G}_{\mathsf{LL}}^{-1}\begin{pmatrix}\mathbf{y}_{\mathsf{T}} \\ l_{\mathsf{A}}\end{pmatrix} + \mathsf{c}_1 - \frac{1}{2}\ln|\mathbf{G}_{\mathrm{all}}| \\
&= -\frac{1}{2}(\mathbf{y}_{\mathsf{T}}^\top l_{\mathsf{A}}^\top)\mathbf{G}_{\mathsf{LL}}^{-1}\begin{pmatrix}\mathbf{y}_{\mathsf{T}} \\ l_{\mathsf{A}}\end{pmatrix} + \mathsf{c}_2 \qquad (14)
\end{aligned}$$

Let's turn our attention to the first term $\mathsf{Q}_1(l_{\mathsf{A}})$ of $\mathcal{J}(l_{\mathsf{A}}, l_{\mathsf{U}})$. We define $\pi(l_{\mathsf{j}}) = (1 + \mathrm{e}^{-2\gamma l_{\mathsf{j}}})^{-1}$, where $j = n_{\mathsf{T}}+1, n_{\mathsf{T}}+2, \ldots, n_{\mathsf{T}}+n_{\mathsf{A}}$, the sigmoid noise label generation model can be written as :

$$\begin{aligned}
P(y_{\mathsf{j}}|l_{\mathsf{j}}) &= \frac{e^{\gamma l_{\mathsf{j}} y_{\mathsf{j}}}}{e^{\gamma l_{\mathsf{j}} y_{\mathsf{j}}} + e^{-\gamma l_{\mathsf{j}} y_{\mathsf{j}}}} \\
&= \left(\frac{e^{\gamma l_{\mathsf{j}}}}{e^{\gamma l_{\mathsf{j}}} + e^{-\gamma l_{\mathsf{j}}}}\right)^{\frac{1+y_{\mathsf{j}}}{2}}\left(1 - \frac{e^{\gamma l_{\mathsf{j}}}}{e^{\gamma l_{\mathsf{j}}} + e^{-\gamma l_{\mathsf{j}}}}\right)^{\frac{1-y_{\mathsf{j}}}{2}} \\
&= \pi(l_{\mathsf{j}})^{\frac{1+y_{\mathsf{j}}}{2}}(1 - \pi(l_{\mathsf{j}}))^{\frac{1-y_{\mathsf{j}}}{2}} \qquad (15)
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathsf{Q}_1(l_{\mathsf{A}}) &= \ln(\mathrm{P}(\mathbf{y}_{\mathsf{A}}|l_{\mathsf{A}})) \\
&= \sum_{j=n_{\mathsf{T}}+1}^{n_{\mathsf{L}}} \ln(\mathrm{P}(y_{\mathsf{j}}|l_{\mathsf{j}})) \\
&= \gamma(\mathbf{y}_{\mathsf{A}} - \mathbf{1})^\top l_{\mathsf{A}} - \sum_{j=n_{\mathsf{T}}+1}^{n_{\mathsf{L}}} \ln(\mathbf{1} + \mathrm{e}^{-2\gamma l_{\mathsf{j}}}) \qquad (16)
\end{aligned}$$

Combine ~~$\mathsf{Q}_1(l_\mathsf{A})$~~ and $\mathsf{Q}_2(l_\mathsf{A}, l_\mathsf{U})$ together, we obtain the following revised objective function $\mathcal{J}(l_\mathsf{A})$. Maximize ~~$\mathcal{J}(l_\mathsf{A})$~~ over $l_\mathsf{A} \in \mathbb{R}^{\mathrm{n_A}}$ we get

$$\mathcal{J}(l_A) = \gamma(\mathbf{y}_A - \mathbf{1})^\top l_A - \sum_{j=n_T+1}^{n_L} \ln(1 + \mathrm{e}^{-2\gamma l_j}) - \frac{1}{2}(\mathbf{y}_T^\top l_A^\top)\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathbf{y}_\mathsf{T}\\ l_\mathsf{A}\end{pmatrix} + c_2 \tag{17}$$

The gradient vector $\hat{l_\mathsf{A}}$ ~~given~~ by a straightforward calculation

$$\left.\frac{\partial \mathcal{J}(l_A)}{\partial l_A}\right|_{l_A=\hat{l_A}} = \gamma(\mathbf{y}_A - \mathbf{1}) + 2\gamma(1 - \pi(\hat{l_A})) - \mathrm{G}_{LL}^{-1}\hat{l_A} \tag{18}$$

Furthermore let $\mathbf{G}_{\mathsf{LL}}^{-1} = \begin{pmatrix}\mathbf{B}_{\mathsf{TT}} & \mathbf{B}_{\mathsf{TA}}\\ \mathbf{B}_{\mathsf{AT}} & \mathbf{B}_{\mathsf{AA}}\end{pmatrix}$, Eq. (18) can be written as

$$\left.\frac{\partial \mathcal{J}(l_A)}{\partial l_A}\right|_{l_A=\hat{l_A}} = \gamma(\mathbf{y}_A - \mathbf{1}) + 2\gamma(1 - \pi(\hat{l_A})) - \mathrm{B}_{AA}\hat{l_A} - \mathrm{B}_{AT}\mathrm{y}_T \tag{19}$$

Where $\pi(\hat{l_\mathsf{A}}) = (\pi(\hat{l_{\mathrm{n_T}+1}}), \ldots, \pi(\hat{l_{\mathrm{n_L}}}))^\top$. We can see from this expression, the term $\pi(\hat{l_\mathsf{A}})$ make it impossible to compute $\hat{l_\mathsf{A}}$ in a closed form, we use the Newton-Raphson method.

$$l_A^{i+1} \leftarrow l_A^i - \eta\mathbf{H}^{-1} \cdot \left.\frac{\partial \mathcal{J}(l_A)}{\partial l_A}\right|_{l_A=l_A^i} \tag{20}$$

Where $\mathbf{H}$ is $\mathsf{n_A} \times \mathsf{n_A}$ Hessian matrix defined as :

$$\mathbf{H}_{\hat{l_A}} = \begin{pmatrix} \left.\frac{\partial^2 \mathcal{J}(l_\mathsf{A})}{\partial l_{\mathrm{n_T}+1}\partial l_{\mathrm{n_T}+1}}\right|_{l_{\mathrm{n_T}+1}=\hat{l_{\mathrm{n_T}+1}}} & \cdots & \left.\frac{\partial^2 \mathcal{J}(l_\mathsf{A})}{\partial l_{\mathrm{n_T}+1}\partial l_{\mathrm{n_L}}}\right|_{l_{\mathrm{n_T}+1}=\hat{l_{\mathrm{n_T}+1}}, l_{\mathrm{n_L}}=\hat{l_{\mathrm{n_L}}}} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial^2 \mathcal{J}(l_\mathsf{A})}{\partial l_{\mathrm{n_L}}\partial l_{\mathrm{n_T}+1}}\right|_{l_{\mathrm{n_L}}=\hat{l_{\mathrm{n_L}}}, l_{\mathrm{n_T}+1}=\hat{l_{\mathrm{n_T}+1}}} & \cdots & \left.\frac{\partial^2 \mathcal{J}(l_\mathsf{A})}{\partial l_{\mathrm{n_L}}\partial l_{\mathrm{n_L}}}\right|_{l_{\mathrm{n_L}}=\hat{l_{\mathrm{n_L}}}} \end{pmatrix} = -\mathbf{P} - \mathbf{B}_{AA} \tag{21}$$

Where $\mathbf{P}$ is a diagonal matrix with elements $P_{\mathsf{ii}} = \mathsf{4}\gamma^2\pi(l_i)(1 - \pi(l_i))$ and $\eta \in \mathbb{R}^+$ has to be chosen such that $\mathcal{J}(l_\mathsf{A}^{\mathsf{i+1}}) > \mathcal{J}(l_\mathsf{A}^{\mathsf{i}})$. We set η ~~0.4~~ and the number of iterations used in Newton-Raphson method is 7.

As the latent variable l can be the soft substitution of label y, we build two trackers by using auxiliary and target samples respectively. For each tracker based on the derivation above we compute the soft label vector l_U, hence we get two candidate set V_A and V_T. We check the similarity of two sets, if the similarity is high we can use any one, if the similarity is low we rely more on the target decision to ensure the tracking consistency. When the similarity is zero, we use the auxiliary decision to handle the heavy occlusion or the severe appearance ~~change.~~

Algorithm 1 Gaussian Process Regression for Tracking

Input: Labeled training set , candidate set $\mathcal{X}_\mathsf{U}$

Output: tracking results(average location of n most likely candidate)

once the tracker initialized construct the sample set
if n < Threshold **then**
    calculate weight matrix W based on the target and unlabeled samples
    construct prior matrix $\mathbf{G}_{\mathsf{all}}^{\mathsf{t}}$ according to the analysis above
    use target samples compute the $\hat{l_\mathsf{U}}$ vector
    track results by averaging the n most likely sample's location indexed in $\hat{l_\mathsf{U}}$
else
    calculate weight matrix W base on all the target, auxiliary ,unlabeled samples
    calculate re-weighting knowledge $\hat{l_\mathsf{A}}$ according eq.(20)
    calculate soft label vector $\hat{l_\mathsf{U}^\mathsf{A}}$ by using re-weighting knowledge $\hat{l_\mathsf{A}}$
    calculate soft label vector $\hat{l_\mathsf{U}^\mathsf{T}}$ by using target label y_T
    make a tracking decision by comparing the similarity of $\hat{l_\mathsf{U}^\mathsf{A}}$ and $\hat{l_\mathsf{U}^\mathsf{T}}$
end if
return results

4 | TRACKING STRATEGY

The tracking process is carried out by alternating between Gaussian process regression and Kalman filtering. Once a tracker is initialized, we use the prediction from Kalman filter to start the observation model which employs the Gaussian Process Regression to estimate the new targets location. The output of Gaussian Process Regression is then used as the measurement input to the Kalman filter.

4.1 | Tracker establishment and destruction

The lifecycle of a tracker is based on the average matching rate $\bar{\Omega}$ among all the established trackers. The average matching rate is defined as :

$$\bar{\Omega} = \frac{\psi + \sum_{i=0}^{N_{trackers}} \Delta N_i^{matched}}{\chi + \Delta t \cdot N_{trackers}} \tag{22}$$

where ψ and χ are the parameters of the Poisson distribution among all the trackers which were set to 30 and 5 respectively. Once a tracker candidate is activated, we keep it in the waiting list until its detection rate is above the ξ_{init}. On the contrary a tracker will be killed when its detection rate is less than ξ_{term}.

$$\xi_{init} = \Omega - \gamma_1\sqrt{\Omega} \tag{23}$$

$$\xi_{term} = \Omega - \gamma_2\sqrt{\Omega} \tag{24}$$

where γ_1 and γ_2 is the scale factor, we set to 1 and 2 respectively, for each tracker's detection rate is defined as :

$$\Omega_i = \frac{\Delta N_i^{matched}}{\Delta t} \tag{25}$$

where $\Delta N_i^{matched}$ refers to the number of detections matched with T_{i} in a sliding window of length Δt.

A tracker's confidence is proportional to the number of template it possesses, following this rule we divided trackers into two groups based on the template it owes. Once a tracker is born we call it **tyro**, it will accumulate templates throughout the tracking process, after **K** template accumulated over a period of robust tacking time, a tyro would be promoted to ~~**veteran**~~ Conversely~~,~~ a veteran ~~demoted~~ to a tyro when it loses template less than K, we set K to 5. Veteran ~~correct~~ its Kalman filter every step~~,~~ ~~on~~ the contrary, when a new detection is assign to a tyro T_{i}~~.~~ ~~Posterior~~ state of T_{i} is replace by the location of the new detection without updating, a tyro jumps directly to the newly detection's location. This allows a tyro ~~recover~~ its tracking during a short-term occlusion or the abrupt appearance change, and then can be viewed as an adaptive Kalman filter mechanism. Each tracker ~~keep~~ at most N_{max} reliable templates by discarding the lower score template, we set N_{max} ~~10~~. In order to deal with the scale change problem we introduce a learning rate Θ_{s} ~~each~~ tracker updates the size of its bounding box $(w_{\text{t}}^{\text{i}}, h_{\text{t}}^{\text{i}})$ with the following equation $(w_{\text{t}}^{\text{i}}, h_{\text{t}}^{\text{i}}) = \Theta_{\text{s}}(w_{\text{det}}, h_{\text{det}}) + (1 - \Theta_{\text{s}})(w_{\text{t}-1}^{\text{i}}, h_{\text{t}-1}^{\text{i}})$~~. Where~~ $(w_{\text{det}}, h_{\text{det}})$ is the size of the new assignment, we set learning rate Θ_{s} to 0.4.

4.2 | Detection Association

The traditional ~~"~~data association~~"~~ problem can be interpreted as from the tracking literature, in which new "observation" must be associated with already-established "tracks" [14]. At each frame we have D_{t} new detections, which need to be assigned to already-established tracker. We solve the assignment problem by finding the maximum matching in the bipartite graph. We use Hungarian Algorithm [17] with cost matrix.

$$c_{ij} = \begin{cases} dist_{\text{ij}}, & dist_{\text{ij}} < K_{\text{i}} \\ \infty, & ohterwise \end{cases}$$

where $K_{\text{i}} = \alpha_{\text{i}} w_{\text{i}}$, α_{i} is calculated as $\alpha_{\text{i}} = min(\phi_1 \frac{\sigma_{kf_{\text{i}}}}{w_{\text{i}}} + \phi_2, \alpha_{\text{max}})$ and $\sigma_{kf_{\text{i}}}$ refer to the square root of the posteriori covariance kf_{t}, w_{i} is the width of the bounding box. We set $(\phi_1, \phi_2, \alpha_{\text{max}})$ to (0.5, 1.5, 4), $dist_{\text{ij}}$ is the distance between center point of the detection d_{t}^{i} and the current target position.

5 | EXPERIMENTS

We ~~test~~ our new approach on a 2.8GHz octa-core CPU,16GB memory computer~~, our system is~~ implemented in C++ using OpenCV and Eigen library. There is no unified accepted benchmark available for multi-person tracking. Most ~~the~~ recent ~~publication~~ ~~has~~ tested

Algorithm 2 Multi-Person Tracking

Input: Tracker pools T_t, Detections D_t, Current frame I_t
Output: Updated tracker pool T_t

associate new detections to the already-established tracker
tracker initialization
for candidate $V_i \in$ waiting list **do**
  if V_i's detection rate exceed ξ_{init} **then**
    tracker V_i initialized
  end if
end for
waiting list updated
tracker termination
for tracker $T_t^i \in T_t$ **do**
  if T_t^i has detection rate less than ξ_{term} **then**
    remove T_t^i from T_t
  end if
end for
T_t updated
if a tyro tracker T_t^i accumulated template number more than or equal **K then**
  promote T_t^i to veteran
end if
if a veteran tracker T_t^i loss template and the number less than **K then**
  demote T_t^i to tyro
end if
Matching rate update

their ~~approach~~ on their own ~~sequence~~, which varies from viewpoint, density of pedestrian, and ~~amount~~ of occlusion. We combined them together for the evaluation of our algorithm, these sequences include: TUD-Campus [10] and TUD-Stadtmitte [24], PETS'09 S2.L1 - S2.L2 - S2.L3 [25], TownCenter [12]. ~~Runtime performance depend on different sequence, most of the sequences can achieve real-time performance. In the following section we will detail the different several experiments.~~

5.1 | Evaluation Metrics

~~To measure performance of our system, we~~ employed the CLEAR MOT metrics proposed by [26].The Multiple Object Tracking Accuracy (MOTA) ~~consider~~ false positive, missed targets and identity switches. The Multiple Object Tracking Precision (MOTP) takes ~~account into~~ the average distance between estimated location and ~~ground~~ truth. Note that ~~higher~~ value of these metrics ~~stand for~~ ~~better~~ performance. Furthermore we also compute the metrics described in [27], which considers the partially tracked (PT), the counts the number of mostly tracked (MT), mostly lost (ML) trajectories, number of track fragmentations (FM) and identity switches (IDS).

5.2 | Example Results

The PETS 2009 Dataset [25] is proposed by the Computational Vision Group University of Reading. This dataset ~~include~~ four ~~subset~~ used for different ~~purpose~~ of visual ~~analysis,~~ ~~among~~ them ~~S2~~ subset sequence is designed for testing the performance of the tracking algorithm. We ~~test~~ our algorithm on the S2L1, L2, and L3 ~~sequence~~, the density of pedestrian ~~is raising~~ according to the order of the dataset. S2L1 sequence exhibits a randomly walking sparse crowd. S2L2 sequence exhibits a randomly walking dense crowd. S2L3 sequence shows two individuals which are bystanders in an empty scene and later join a moving crowd which walking in the same direction. While this sequence has a very crowded crowd, the crowd is occlude heavily even for pedestrian detection. A brief description of the PETS sequences is ~~as~~ shown in Table 1.

	Frame Rate	Number of frames	Number of Id	Our Method	
				Precision	Recall
PETS2009 S2L1	7	795	19	0.87	0.81
PETS2009 S2L2	7	436	43	0.90	0.59
PETS2009 S2L3	7	240	44	0.88	0.33

TABLE 1 A brief description of the PETS S2 sequence

Sequence	Tracker	Rcll(%)	Prcn(%)	FP	IDSW	MOTA(%)	MOTP(%)
PETS2009-S2L1	DP_NMS[28]	83.3118	80.9783	910	348	56.2581	71.119
	Ours	81.957	87.6495	537	126	67.6989	62.5369
	TC_ODAL[29]	81.6559	83.4139	755	31	64.7527	70.4317
	TBD[30]	81.2903	84.2434	707	239	60.9462	71.1903
	SMOT[31]	75.1828	91.5663	322	99	66.129	71.5962
	[32][32]	61.2043	99.0257	15	15	60.2796	68.2216
	[33][33]	-	-	-	-	67	-
PETS2009-S2L2	[1][32]	25.2235	98.8199	31	47	24.4656	61.3475
	MotiCon[34]	54.8387	90.4224	560	238	46.5616	67.6273
	Ours	59.9883	90.2896	664	420	49.4559	52.9748
	SNM	57.048	85.6298	923	240	44.985	68.4288
	JPDA_m[35]	49.611	82.4797	1016	139	37.631	65.9038
	SORT[36]	38.2014	82.045	806	240	27.3519	67.361
	MPTDLPF	50.8142	81.8957	1083	380	35.6395	67.5972
	RMOT[37]	50.8039	81.3081	1126	190	37.1538	67.6956
	Otakudj	45.4932	79.2985	1145	709	26.2628	68.9981
	Stitiching	45.7007	78.8475	1182	715	26.0243	68.9675
	GSCR[38]	35.5461	78.3673	946	162	24.0535	67.5983
	MPT_CNNPF	22.6118	73.9986	766	351	11.0258	65.416
PETS2009-S2L3	[32][32]	26.4168	96.2531	45	30	24.7029	57.1675
	Ours	33.0439	88.5487	187	38	27.9022	53.0151

TABLE 2 Results comparison of selected PETS dataset

We compare our algorithm with many different approaches, the results of these approaches are obtained from the MOTChallenge, which is as a part of the famous VideoNet challenge and public available[1]. A comprehensive comparison is ~~as~~ shown in Table 2.

As can be seen from Table 2, our method ~~output~~ higher recall and precision ~~among~~ all three datasets. For the S2L1 sequence the MOTA score success surpass all other methods, we also get comparable MOTP score. We believe the slightly lower MOTP score was owing to the update of the sample set not perfectly adapt the scale change over time, our method have less missing detection and potentially increases the number of false positive. We also ~~compare~~ our method ~~with~~[33] ~~when available~~, our method ~~get~~ higher MOTA score ~~than~~[33]. For S2L2 sequence we have the best MOTA score and comparable MOTP score. Besides we also ~~test~~ our algorithm on the S2L3 dataset, which featur~~ed~~ very crowded ~~crowd occlude each other~~, only a few persons can be tracked accurately. Note that the results ~~of~~[32] are slightly different from the original paper, we ~~do~~ the ~~test~~ ourselves, and it may be influenced by the parameter tuning, pretreatment optimization and other factors. We use the same evaluation metrics ~~as~~[32] and we get comparable performance. It is noticed that with the increase of density of people in the ~~sense~~ there are few veterans, the ratio of veterans is much higher in sequence S2L1 than S2L2, it can be explained ~~by~~ ~~there is~~ more occlusion issues ~~in S2L2~~ than S2L1.

We also ~~test~~ our algorithm on the TownCenter[12] dataset , which ~~initially~~ designed for head tracking, but nowadays widely used for the performance measurement corresponding to multi-person tracking algorithms. This video is high definition (1920×1080 at 25fps), with

[1] https://motchallenge.net/

Sequence	Tracker	Rcll(%)	Prcn(%)	FP	IDSW	MOTA(%)	MOTP(%)
TownCenter	Ours	82.1854	85.5537	6528	247	67.7827	55.827
	sort_pr	28.5814	82.0812	446	87	21.1248	68.995
	GSCR[38]	23.2233	79.1607	437	42	16.5221	67.8075
	SiameseCNN	31.0716	76.0877	698	142	19.3201	68.9768
	ARM	52.6301	75.8621	1197	421	29.9944	68.8219
	SORT[36]	45.0196	74.3359	1111	162	27.2104	67.4241
	MPT_CNNPF	35.7303	72.7221	958	413	16.5501	66.0305
	EAMTTpub[39]	32.8903	72.0723	911	201	17.3335	68.549
	OMT_DFH[40]	39.6335	69.3513	1252	52	21.3906	67.805
	RNN_LSTM[41]	34.4992	67.1569	1206	299	13.4443	68.7955
	TSDA_OAL[42]	45.0196	64.489	1772	105	18.7605	67.3565
	oICF[43]	38.2065	63.8233	1548	82	15.4029	67.542

TABLE 3 Results Comparison of TownCenter dataset

Sequence	Tracker	Rcll(%)	Prcn(%)	FAR	GT	MT	PT	ML	FP	FN	IDSW	FM	MOTA(%)	MOTP(%
TUD-Stadtmitte	DP_NMS[28]	79.58	81.56	1.16	10	8	2	0	208	236	40	25	58.13	69.87
	Ours	54.8	77.4	1.03	10	2	8	0	185	522	6	32	38.3	61.6
	TBD[30]	82.26	82.91	1.09	10	8	2	0	196	205	28	13	62.88	69.51
	TC_ODAL[29]	78.46	86.38	0.79	10	7	3	0	143	249	6	17	65.57	69.91
	CEM[45]	74.91	93.11	0.35	10	6	4	0	64	290	11	9	68.42	69.65
	SMOT[31]	71.10	92.98	0.34	10	4	6	0	62	334	16	26	64.35	70.16
TUD-Campus	DP_NMS[28]	72.14	75.29	1.19	8	3	5	0	85	100	44	22	36.21	74.17
	Ours	27	67.4	0.66	8	0	5	3	47	262	4	18	12.8	66.7
	TBD[30]	76.32	86.43	0.60	8	5	3	0	43	85	9	12	61.83	74.85
	TC_ODAL[29]	58.77	86.83	0.45	8	1	7	0	32	148	6	14	48.18	74.02
	CEM[45]	65.18	89.65	0.38	8	3	5	0	27	125	15	8	53.48	74.73

TABLE 4 Results comparison of our algorithm and others

an average of sixteen people visible at any time[12]. Both the ground truth for this sequence and dataset ~~is public~~ available[2]. The results show that our algorithm outperforms other algorithms in recall, precision and MOTA categories, while more ~~detection~~ increase the risk of false positive.

PETS'09 S2.L1 - S2.L2 and TownCenter ~~dataset, all of these~~ datasets ~~is as~~ part of the MOT challenge 2015 benchmark[44]. In addition, we also selected two additional datasets from this benchmark, which ~~is~~ TUD-Campus[10] and TUD-Stadtmitte[24]. These two datasets ~~is public~~ available[3], there is also a development kit ~~public available~~[4], which is used for fair comparison.

TUD-Campus sequence filmed at a horizontal view with people walking in two opposite ~~direction~~ ~~occlude~~ each other, TUD-Stadtmitte sequence filmed at a lower view, pedestrian dressed in similar dresses , walking disorderly and ~~occlusion~~ happen a lot~~, this~~ makes the color-based observation model very hard to track people accurately. Results comparison is ~~as~~ shown in Table 4. The results show that our algorithm ~~output~~ comparable performance compared with others.

[2]http://www.robots.ox.ac.uk/ActiveVision/Publications/benfold_reid_cvpr2011/benfold_reid_cvpr2011.html
[3]https://motchallenge.net/data/2D_MOT_2015/
[4]https://bitbucket.org/amilan/motchallenge-devkit/

6 | CONCLUSION AND FUTURE WORK

In this paper we ~~introduced~~ a semi-supervised tracking algorithm with a new observation model adopt graph Laplacian. Furthermore the prior gram matrix is constructed based on all samples, by this way future information have strong influence on the tracking decision can be viewed as a transfer learning strategy. We devise multi-person tracking by using a tracker hierarchy. Trackers are classified into two groups based on the template they owe, different type of tracker adopt different update strategy during the tracking process. In the future work we will incorporate re-identification scheme in our algorithm to help account for people re-identification problem, we will extend this framework to Multi-Target, Multi-Camera Tracking.

ACKNOWLEDGMENTS

This work is supported by the Principal Foundation of Xiamen University, No.20720180075

References

1. Avidan Shai. Ensemble tracking. *IEEE transactions on pattern analysis and machine intelligence.* 2007;29(2).

2. Breitenstein Michael D, Reichlin Fabian, Leibe Bastian, Koller-Meier Esther, Van Gool Luc. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE transactions on pattern analysis and machine intelligence.* 2011;33(9):1820–1833.

3. Andriyenko Anton, Roth Stefan, Schindler Konrad. An analytical formulation of global occlusion reasoning for multi-target tracking. In: :1839–1846IEEE; 2011.

4. Shitrit Horesh Ben, Berclaz Jerome, Fleuret Francois, Fua Pascal. Tracking multiple people under global appearance constraints. In: :137–144IEEE; 2011.

5. Black James, Ellis Tim, Rosin Paul. Multi view image surveillance and tracking. In: :169–174IEEE; 2002.

6. Breitenstein Michael D, Reichlin Fabian, Leibe Bastian, Koller-Meier Esther, Van Gool Luc. Robust tracking-by-detection using a detector confidence particle filter. In: :1515–1522IEEE; 2009.

7. Song Xuan, Cui Jinshi, Zha Hongbin, Zhao Huijing. Vision-based multiple interacting targets tracking via on-line supervised learning. In: :642–655Springer; 2008.

8. Berclaz Jerome, Fleuret Francois, Fua Pascal. Robust people tracking with global trajectory optimization. In: :744–750IEEE; 2006.

9. Huang Chang, Wu Bo, Nevatia Ramakant. Robust object tracking by hierarchical association of detection responses. In: :788–801Springer; 2008.

10. Andriluka Mykhaylo, Roth Stefan, Schiele Bernt. People-tracking-by-detection and people-detection-by-tracking. In: :1–8IEEE; 2008.

11. Comaniciu Dorin, Ramesh Visvanathan, Meer Peter. Real-time tracking of non-rigid objects using mean shift. In: :142–149IEEE; 2000.

12. Benfold Ben, Reid Ian. Stable multi-target tracking in real-time surveillance video. In: :3457–3464IEEE; 2011.

13. Blake Andrew, Isard Michael. The condensation algorithm-conditional density propagation and applications to visual tracking. In: :361–367; 1997.

14. Huang Timothy, Russell Stuart. Object identification in a bayesian context. In: :1276–1282; 1997.

15. Reid Donald. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control.* 1979;24(6):843–854.

16. Shalom Bar. *Fortmann, Tracking and Data Association.* 1988.

17. Munkres James. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics.* 1957;5(1):32–38.

18. Gao Jin, Ling Haibin, Hu Weiming, Xing Junliang. Transfer learning based visual tracking with gaussian processes regression. In: :188–203Springer; 2014.

19. Dalal Navneet, Triggs Bill. Histograms of oriented gradients for human detection. In: :886–893IEEE; 2005.

20. Herbrich Ralf. *Learning Kernel classifiers: theory and algorithms (adaptive computation and machine learning)*. MIT press; 2002.

21. Hu Weiming, Li Xi, Luo Wenhan, Zhang Xiaoqin, Maybank Stephen, Zhang Zhongfei. Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2012;34(12):2420–2440.

22. Zhu Xiaojin, Ghahramani Zoubin, Lafferty John D. Semi-supervised learning using gaussian fields and harmonic functions. In: :912–919; 2003.

23. Zhu Xiaojin, Lafferty John, Rosenfeld Ronald. Semi-supervised learning with graphs. PhD thesisCarnegie Mellon University, language technologies institute, school of computer science2005.

24. Andriluka Mykhaylo, Roth Stefan, Schiele Bernt. Monocular 3d pose estimation and tracking by detection. In: :623–630IEEE; 2010.

25. Ferryman J, Shahrokni A. Pets2009: Dataset and challenge. In: :1–6IEEE; 2009.

26. Kasturi Rangachar, Goldgof Dmitry, Soundararajan Padmanabhan, et al. 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.* 2009;31(2):319–336.

27. Bernardin Keni, Stiefelhagen Rainer. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *Journal on Image and Video Processing.* 2008;2008:1.

28. Pirsiavash Hamed, Ramanan Deva, Fowlkes Charless C. Globally-optimal greedy algorithms for tracking a variable number of objects. In: :1201–1208IEEE; 2011.

29. Bae Seung-Hwan, Yoon Kuk-Jin. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: :1218–1225; 2014.

30. Geiger Andreas, Lauer Martin, Wojek Christian, Stiller Christoph, Urtasun Raquel. 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence.* 2014;36(5):1012–1025.

31. Dicle Caglayan, Camps Octavia I, Sznaier Mario. The way they move: Tracking multiple targets with similar appearance. In: :2304–2311IEEE; 2013.

32. Zhang Jianming, Presti Liliana Lo, Sclaroff Stan. Online multi-person tracking by tracker hierarchy. In: :379–385IEEE; 2012.

33. Leal-Taixé Laura, Pons-Moll Gerard, Rosenhahn Bodo. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: :120–127IEEE; 2011.

34. Leal-Taixé Laura, Fenzi Michele, Kuznetsova Alina, Rosenhahn Bodo, Savarese Silvio. Learning an image-based motion context for multiple people tracking. In: :3542–3549; 2014.

35. Rezatofighi Seyed Hamid, Milan Anton, Zhang Zhen, Shi Qinfeng, Dick Anthony R, Reid Ian D. Joint Probabilistic Data Association Revisited.. In: :3047–3055; 2015.

36. Bewley Alex, Ge Zongyuan, Ott Lionel, Ramos Fabio, Upcroft Ben. Simple online and realtime tracking. In: :3464–3468IEEE; 2016.

37. Yoon Ju Hong, Yang Ming-Hsuan, Lim Jongwoo, Yoon Kuk-Jin. Bayesian multi-object tracking using motion context from multiple objects. In: :33–40IEEE; 2015.

38. Fagot-Bouquet Low, Audigier Romaric, Dhome Yoann, Lerasle Frédéric. Online multi-person tracking based on global sparse collaborative representations. In: :2414–2418IEEE; 2015.

39. Sanchez-Matilla Ricardo, Poiesi Fabio, Cavallaro Andrea. Online multi-target tracking with strong and weak detections. In: :84–99Springer; 2016.

40. Ju Jaeyong, Kim Daehun, Ku Bonhwa, Han David K, Ko Hanseok. Online multi-object tracking with efficient track drift and fragmentation handling. *JOSA A.* 2017;34(2):280–293.

41. Milan Anton, Rezatofighi Seyed Hamid, Dick Anthony R, Reid Ian D, Schindler Konrad. Online Multi-Target Tracking Using Recurrent Neural Networks.. In: :4225–4232; 2017.

42. Ju Jaeyong, Kim Daehun, Ku Bonhwa, Han David K, Ko Hanseok. Online multi-person tracking with two-stage data association and online appearance model learning. *IET Computer Vision.* 2016;11(1):87–95.

43. Kieritz Hilke, Becker Stefan, Hübner Wolfgang, Arens Michael. Online multi-person tracking using integral channel features. In: :122–130IEEE; 2016.

44. Leal-Taixé L., Milan A., Reid I., Roth S., Schindler K.. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv:1504.01942 [cs].* 2015;. arXiv: 1504.01942.

45. Milan Anton, Roth Stefan, Schindler Konrad. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence.* 2014;36(1):58–72.