Co-operative Pedestrians Group Tracking in Crowded Scenes using an MST Approach

Anonymous WACV submission

Paper ID 394

Abstract

We address the problem of multiple pedestrian tracking in crowded scenes in videos recorded by a static uncalibrated camera. We propose an online multiple pedestrian tracking algorithm that utilizes group behaviour of pedestrians using minimum spanning trees (MST). We first divide pedestrians into several groups using the agglomerative hierarchical clustering, taking position and velocity of pedestrians as features, and then we track each group, represented by an MST, with the pictorial structures method. We also propose: (1) a method to detect and handle inter-pedestrian occlusions using a custom trained head detector for crowded scenes, and (2) an efficient method to detect newly entered pedestrians in the frame with help of a background subtraction method. Finally, we present experiments on two challenging and publicly available datasets and show improvements on multiple object tracking accuracy (MOTA) over other methods.

1. Introduction and Related Work

Multiple pedestrian tracking in crowded scenes remains an important problem in computer vision. In this paper, we address the problem of automatically detecting and tracking multiple pedestrians in videos recorded by a static camera. This problem is very challenging because of the following factors: (1) changing background, (2) varying pedestrians appearance and *inter-pedestrian occlusions*, and (3) image noise and illumination variation.

Most popular approach to solve this problem has been *tracking-by-detection* [24] because of availability of fast reliable pedestrian detectors [12, 13]. These approaches involve continuous application of pedestrian detector on every frame and association of detections across frames. These approaches are generally robust to changing background and pedestrians appearance. In our work also, we use the Histogram of Oriented gradients (HOG) [12] features and support vector machine (SVM) classifier based appearance model for detecting pedestrians in images.

Multiple pedestrian tracking algorithms can also be classified into two groups: online and offline. Several tracking algorithms use a large temporal window to store detections including detections from future frames and then perform association to get the final optimized trajectories of pedestrians [2, 4, 18]. These algorithms are offline tracking algorithms and they suffer with a small delay depending on the temporal window. On the other hand, online algorithms [24, 15, 21], including ours, consider only information available from past frames for tracking and are more suitable for time critical applications.

Many existing online algorithms that track multiple objects are based on tracking each object individually [8, 24, 21, 3, 15]. They use an appearance model to detect humans in each frame and perform associations across frames to get final results. The problem with these approaches is that the output of appearance models is unreliable when a pedestrian is occluded, and performing correct associations in this case is very difficult. We, on the other hand, propose an approach that relies on pedestrians group behaviour. We propose that if some of the pedestrians are moving in a direction then they form a certain structure. Hence, this gives us extra prior to effectively track them.

SPOT method by Zhang et. al. [26] also uses a group behaviour to track multiple objects robustly and shows promising results where there is a common movement of objects in the video (e.g. flower movement in presence of wind and camera movement). However, this method assumes that the objects have same size throughout the video and all objects move together in one particular direction. Hence, this method is hard to use for tracking multiple pedestrians because of change in their size as they move in the video frame and their uncertain movements in different directions.

We, on the other hand, first divide pedestrians into several groups and then perform group-wise tracking. To divide pedestrians into different groups, we use a greedy clustering method with our own custom metric that makes sure that the formed groups have a particular directions and are compact. And for tracking, we use method similar to SPOT [26] using the pictorial structure method [14] on the minimum spanning tree (MST) formed with members of a group.

Some multiple pedestrian tracking algorithm perform occlusion handling using part based appearance models [15, 21]. They detect which parts of pedestrians are occluded on basis of part detector scores and take necessary steps to improve tracking. However, training and running individual part detectors is a cumbersome task and it requires a lot of extra computation during tracking. We, on the other hand, first mark the occluded pedestrians based on their bounding box locations and then use a simple HOG-SVM based head detector to handle occlusions.

Our contributions are the following: (1) we propose an online tracking algorithm that utilizes group behavior of pedestrians using minimum spanning trees, (2) We exploit pedestrian location and directions for clustering to divide them into separate groups and then track each group separately, (3) we propose a method to handle inter-pedestrian occlusions in crowded scenes, and (4) we propose an efficient method for detection of newly entered pedestrians parallel to tracking.

In the rest of this paper, we first present details of appearance model used in our algorithm in Section 2, followed by an overview and step by step details of our algorithm in Section 3, and then we presents details of pedestrian entry and exit detection method in Section 4. We later present the computational complexity of our algorithm in Section 5, experiments, results and comparison with others on two datasets in Section 6, and conclusion in Section 7.

2. The Appearance Model

An appearance model in any tracking system predicts the likelihood of an object present at a particular location in an image. We use the popular Dalal-Triggs detector's [12] appearance model in our algorithm. This detector uses Histogram-of-Oriented-Gradients (HOG) features to describe rectangular image patches and a linear Support Vector Machine (SVM) classifier to predict the likelihood of a pedestrian presence. Advantages of using HOG feature over others are: (1) they cover orientations other than only horizontal and vertical ones, (2) they are robust to changes in the illumination of the tracked objects because they allow more variations in the appearance and shape than other more rigid methods, and, (3) they can be summed on relatively smaller image regions. Together, they make HOG features highly sensitive to the object location in the image, which is very useful for tracking as pedestrians move continuously. For further details, we refer the reader to [12].

Let $l = [x, y, w, h]$ be a location in an image I, where x, y, w, h are the x, y coordinates, width and height of an image patch, and we are searching for a pedestrian at this particular location. Formally, the detector response is given by:

$$R(I|l) = w^T \phi(I|l) + bias \tag{1}$$

where $\phi(I|l)$ is an HOG feature vector of the image patch, and $w, bias$ are an SVM weight vector and bias respectively. The response of the SVM classifier is a real number and to convert it to a probability response, we use Platt scaling[20]:

$$P(I|l) = \frac{1}{1 + exp(-R)} \tag{2}$$

where $P(I|l)$ represents the probability of the presence of a pedestrian at a location l in the image.

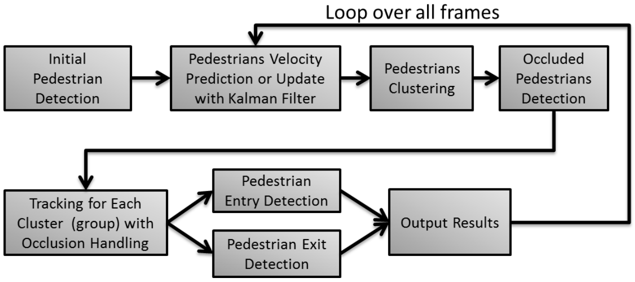
Loop over all frames
Initial Pedestrian Detection
Pedestrians Velocity Prediction or Update with Kalman Filter
Pedestrians Clustering
Occluded Pedestrians Detection
Tracking for Each Cluster (group) with Occlusion Handling
Pedestrian Entry Detection
Pedestrian Exit Detection
Output Results

Figure 1. Box diagram describing high level description of our tracking algorithm.

3. Tracking

We propose an online tracking algorithm that tracks multiple pedestrians over time in a video. Our key idea is that we exploit the group behavior of pedestrians through dividing pedestrians into one or more separate clusters (groups), and then we track each group separately. We divide pedestrians into several groups using agglomerative hierarchical clustering taking location and velocity of pedestrians as features, and then we track each group optimally, which is represented by an MST, with pictorial structures [14] method.

We show by experiments that exploiting group behavior of pedestrians is advantageous for robust tracking. It enables our algorithm to better predict the locations of some of the pedestrians when they are occluded or when the image is noisy. For example, let G be a group of pedestrians $X_i \in G, i = [1...n]$, and at some point of time a pedestrian $X_k \in G$ is occluded while the rest of the members of G are visible. Since we have prior information about the group velocity, we can predict the location of the occluded pedestrian X_k based on the velocities of other group members $X \subseteq G \setminus X_k$.

A brief overview of our approach is as follows. For the first frame, we use a trained appearance model as described in Section 2 to detect pedestrians. Then for the next few frames, we track them individually with the help of only the detector responses (described in Section 3.4). Then, for each pedestrian bounding box, once we have its history of locations, we use Kalman filtering to predict its velocity (Section 3.1). Subsequently, we perform clustering on the current set of pedestrian bounding boxes and cluster them into groups (Section 3.2). Then, we find the optimal location of each of the group members with the method specified in Section 3.3. The occluded pedestrians are handeled by with method described in Section 3.5. Apart from the tracked objects, one needs to check for entry and exit events of pedestrians (this is described in Sections 4.1 and 4.2 respectively). A box diagram showing the various steps in our algorithm is shown in Figure 1.

3.1. Velocity Prediction using Kalman Filtering

As we have fresh pedestrians in the frame, we track them by detection with our appearance model for a few frames F_{win}, and then, for each subsequent frame, we recursively compute individual pedestrian velocity using a linear Kalman filter [16] taking all previous locations as input. Then, the predicted velocity for each pedestrian is used as input to the clustering algorithm.

We use the following Kalman filter model:

$$\begin{aligned} X(t) &= FX(t-1) + W(t), W \sim N(0,Q) \\ Y(t) &= HX(t) + V(t), V \sim N(0,R) \end{aligned} \tag{3}$$

where $X(t) = [x, y, dx, dy]^T$ is a state vector of position and velocity of a pedestrian bounding box for time t, $F(t)$ is a state transition matrix, and $W(t)$ is a vector containing the process noise terms for each parameter in the state vector. The process noise is assumed to be drawn from a zero mean multivariate normal distribution with covariance matrix Q. $Y(t)$ here denotes an observation at time t, $H(t)$ is a transformation matrix that maps the state vector parameters into the observation domain, and $V(t)$ is a vector containing the observation noise terms. Similar to the process noise, this observation noise is assumed to be a zero mean Gaussian white noise with covariance matrix R. For further details of Kalman filtering algorithm, the reader is referred to [16].

3.2. Clustering Pedestrians into Groups

After we have velocities of moving pedestrians in video frame, we perform clustering on them and group them into one or more separate clusters (groups). Each group consists of pedestrians moving in a similar direction and close to each other. The idea is that the motion of people in a group is most probably correlated to each other and that there is a high chance that they will continue to move together.

We use positions and directions of pedestrians excluding their speed as features for clustering. This is because that there is not much variation in the speed since the camera is mounted on a high platform and covers only small part of the scene. Although, pedestrians' speed can also be used as a feature for clustering in a scenario where there is high variance in the speed of pedestrians belonging to same group.

The following qualities are required for a good clustering algorithm: (1) it should be fast (2) the maximum difference in the directions of pedestrians in a cluster should be low, and (3) the maximum pairwise Euclidian distance in a cluster should be low.

Many clustering algorithms such as K-means take the number of clusters (k) as input and have a high computational complexity $O(n^{kd+1} \log n)$, where n is number of observations and d is the feature dimension, which makes it unsuitable for our purpose. However, fast greedy clustering algorithms, such as hierarchical agglomerative clustering, run in $O(n^3)$ and are suitable for our purpose.

Hierarchical agglomerative clustering is popular in data mining and works in a bottom up fashion. Each observation starts as a cluster and a pair of clusters is merged in moving up the hierarchy. The merges are determined in a greedy manner and the results of clustering are presented using a dendrogram. There are several measures for the distance between a pair of observations, also called clustering metric, for the purpose of clustering. The most commonly used metric is the Euclidian metric. The choice of metric in the clustering influences the shape of the clusters. Let i^{th} pedestrian have the set of features $O_i = (x_i, y_i, d_i)$, where x_i, y_i are the x and y coordinates, and $d_i \in [0, 2\pi)$ is the direction in which the pedestrian is moving. With these features, we use the following metric to compute the distance between any pair of observations O_i, O_j:

$$\begin{aligned} D(O_i, O_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ + A_w(1 - \cos(d\theta)) \end{aligned} \tag{4}$$

Here $d\theta \in (0, \pi)$ is the absolute angle difference measured as $d\theta = \min(|d_i - d_j|, 2\pi - |d_i - d_j|)$, and $D(O_i, O_j)$ is the distance measure between observation O_i and O_j. The first term of this metric is Euclidian that focuses on the location difference of the observations, and the second term denotes the directional difference of the observations. The weight of the direction difference is determined by the parameter A_w.

Linkage criteria also plays an important role in clustering. It determines the distance between clusters as a function of the pairwise distance between observations. In order to get compact clusters, inter-cluster distance should be a function of the maximum distance between two observations of members of the two clusters. Hence, we use a maximum linkage criterian that takes the maximum distance between two clusters as the distance measure:

$$Dist(A, B) = \max\{D(a, b) : a \in A, b \in B\} \tag{5}$$

where A, B are different clusters, and a, b are the observations belonging to cluster A and B respectively. Two clus-

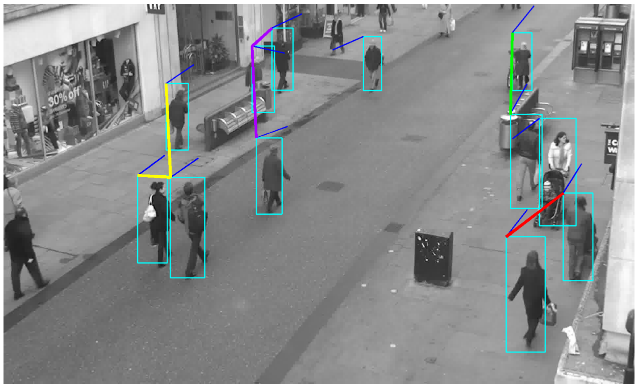

Figure 2. MSTs of clusters (groups) after clustering on a sample frame from Oxford Town Center dataset [3]. The bounding boxes are connected with colored lines that represents the tree and the blue lines show the direction of pedestrians.

ters are paired if the distance between them is less than a threshold C_{th}.

Apart from clustered (grouped) pedestrians, there are newly arrived pedestrians that have unknown velocity. Those are made as singleton clusters and are tracked individually until their velocity is estimated.

3.3. Group Tracking

After clustering, we have one or more groups of pedestrians moving in different directions. To track each group, we use a method similar to SPOT[26]. We assume that the pedestrians in a group maintain a structure in their movement, and the relative movement of pedestrians of a group is less compared to the other groups.

We represent the pedestrian bounding boxes in a group by $B \in V$ where each bounding box $B_i = (x_i, y_i, w_i, h_i)$ is represented by x and y coordinates, width, and height respectively. Subsequently, we define a graph $G = (V, E)$ over pedestrian bounding boxes of a group, where E denotes the set of edges in the graph. The edges in the graph can be viewed as springs that represents spatial constraints between pedestrians. Next, we define the score of the structure $C = \{B_1, \dots B_{|V|}\}$ as a sum of two terms: (1) the appearance scores of individual group member which is the likelihood of the image patch being the pedestrian, and (2) a deformation score that measures how much the structure changes in tracking from the previous to the current frame:

$$s(C|I,\Theta) = \sum_{i \in V} P(I|B_i) \\ - \sum_{(i,j)\in E} \lambda_{ij} \sqrt{(x_i - x_j - e_{ijx})^2 + (y_i - y_j - e_{ijy})^2} \tag{6}$$

where, $P(I|B_i)$ is the probability of the i^{th} pedestrian being present at location B_i which is same as defined in the Section 2, $e_{ij} = (e_{ijx}, e_{ijy})$ is a vector that represents the length and orientation of the connection between the i^{th} and j^{th} pedestrians, and $\Theta = \{w, e_{ij}\}$ denotes the set of parameters.

The parameter λ_{ij} decides the importance given to the deformation between each pair of connected pedestrians. The higher the λ_{ij}, the stiffer is the constraint for the $(i,j)^{th}$ pair. Then, we set this parameter depending on the proximity of pedestrians in a pair:

$$\lambda_{ij} = \frac{\lambda}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \tag{7}$$

The purpose behind this is to a allow higher deformation between farther pedestrians than the closer ones. This follows from the observation that farther objects have lessor correlation in the motion compared to closer ones.

The graph structure of a group plays an important role in inference. Ideally, one should employ the fully connected graph structure, but this would make the inference intractable. Therefore, we use a minimum spanning tree (MST) model similar to [27] as the graph structure. This retains connections only between the closest objects. Examples of different groups MSTs that are formed after clustering in a sample frame are shown in Figure 2.

Given an MST of pedestrians, we estimate the optimal configuration of each single group by maximizing Equation 6 over configuration C. We use the pictorial structure method [14] over a tree structure to maximize the same. This method uses dynamic programming (DP) and is exact and very efficient. The inference on each frame can be performed in linear time depending on the number of pedestrians present in frame. For further details, we refer the reader to [14].

3.4. Individual Tracking

After clustering, there may be some pedestrians that are already detected and are not assigned to any group. We, then, track these pedestrians individually through tracking by a detection method similar to [2] using only their detector response. For each such Pedestrian, we search around a neighbourhood ($\pm delta_{scale}$ and $\pm\delta_{px}$) of its size and select the location with the maximum detector response in the current frame. The scale step for δ_{scale} is taken to be 1.05 as used in most of the detection algorithms. This δ_{px} neighbourhood depends on the video resolution. Higher resolution requires a higher δ_{px} because the pedestrian movement is higher. Please note that we only run the detector in the neighbourhood of the pedestrians and not on the full frame, which takes much less time per pedestrian and the overall running time is linear in the number of pedestrians present in the frame.

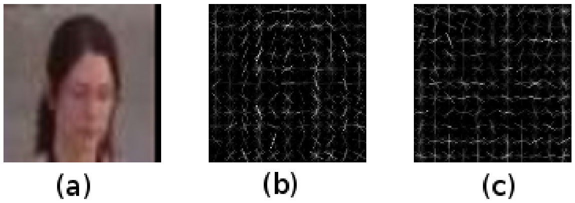

Figure 3. Our trained head detector on Coffee Break Head Orientation dataset [11], a sample positive image is on the left, positive and negative weights of the detector are in middle and right respectively.

3.5. Occlusion Handling

There are several instances when one or more pedestrians are occluded by other pedestrians during tracking. Occlusions happen when two or more pedestrians are travelling in different directions and they overlap at some point of time. Most detection-based tracking methods fail to maintain consistent trajectories for occluded pedestrians and result in drifting. Hence, occlusion handling is essential for better accuracy in tracking.

In case of overhead surveillance cameras, the general pattern in occlusion is that the foreground pedestrian occludes the lower part of the background pedestrian, and the head of the background pedestrian stays visible. Now, with this pattern, we first mark the occluded pedestrians. This, we do by comparing each pair of existing bounding boxes. Then, if two bounding boxes overlap, the one with the lower y coordinate is marked as the occluded pedestrian.

The full body apperance model yields bad scores for the occluded pedestrians. However, their head stays visible. So, we use our separately trained head detector for computing appearance score for occluded pedestrians. This gives reliable location as compared to full body appearance model. Our head detector is trained on the positive images from the Coffee Break Head Orientation dataset [11] and random negative images from outdoor images from [12]. We use the same HOG features and linear SVM classifier as mentioned in Section 2. An example of a positive head image with positive and negative features is shown in Figure 3.

Formally, in a graph $G = (V, E)$, if we denote the set of visible pedestrians by V_v and the set of occluded pedestrians by V_o, then the configuration score given in Equation 6 becomes:

$$s(C|I,\Theta) = \sum_{i \in V_v} P(I|B_i) + \sum_{i \in V_o} P'(I|B_i) - \sum_{(i,j) \in E} \lambda_{ij} \sqrt{(x_i - x_j - e_{ijx})^2 + (y_i - y_j - e_{ijy})^2} \tag{8}$$

Here P' is the probability score after Platt scaling [20] of the response of our head detector. The graph structure and the inference method remains the same as in Section 3.3.

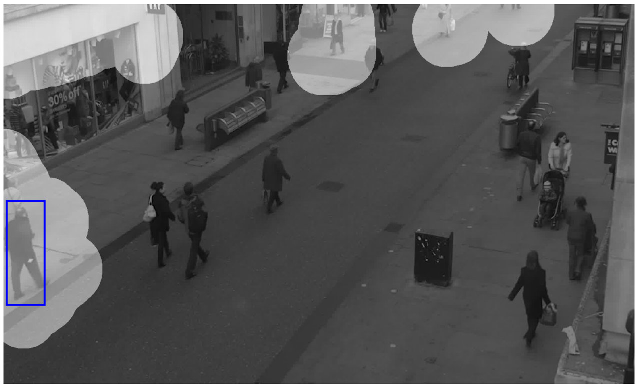

Figure 4. Bright regions contain possible new pedestrians which are not being tracked. A newly entered pedestrian is shown with blue bounding box.

4. Pedestrian Entry and Exit

In our tracking algorithm, we also detect entering and leaving pedestrians in parallel to tracking. For new pedestrians detection, a naive approach is to run the pedestrian detector on full frames and include only new high scoring detections that do not overlap with current bounding boxes. However, this takes significant amount of time to run on a full frame. We use an efficient method to detect new pedestrians that uses background subtraction. Details of new pedestrian detection method is given in Section 4.1. As pedestrians move, they leave the video frame at some point of time. We track pedestrians as soon as they enter the frame until they leave. Details of detecting the exit of pedestrians is given in Section 4.2.

4.1. New Pedestrian Detection

The overall idea for detection of new pedestrians is that background subtraction [22] outputs large blob-like regions where moving pedestrians are present, and we run the pedestrian detector in the close vicinity of these regions to get new high scoring detections.

For each foreground pixel location, we first compute its distance from the nearest bounding box with distance transform [9]. Then we reject the pixels that have less than a minimum foreground distance d_f from the current bounding boxes. This yields locations of foreground pixels that do not overlap with current set of bounding boxes. Now, these selected foreground pixels form several connected components and are potential places where the new pedestrians might exist. However, some of the components may be caused by noise, and hence, we reject small components that are less than a minimum component size d_c. Then we use our human detector in the nearby areas of the selected foreground region, and select only those candidates that have a high scores and are in the local neighbourhood d_t from this foreground region. In the final step, we define a region of interest (ROI) on the image, where we allow entry of new pedestrians only if it is inside that ROI.

$$ROI_{entry} = [30, 30, w_I - w_D, h_I - h_D] \quad (9)$$

where w_I, h_I are the width and height of the image and w_D, h_D are the width and height of the pedestrian detector. An example image showing regions where new pedestrians might exist is presented with high brightness regions in Figure 4.

4.2. Leaving Pedestrian Detection

Since, we have the velocity of each moving pedestrian, we can predict when it is going to leave the frame. Also, the pedestrian detector will produce a good score surly till a pedestrian is completely in the frame. However, as soon as the pedestrian is partially out of the frame, the detector score becomes unreliable. So, we define an ROI for the leaving pedestrians in the image.

$$ROI_{exit} = [10, 10, w_I - w_D, h_I - h_D] \quad (10)$$

where w_I, h_I, w_D, h_D are same as defined in Section 4.1 We stop tracking a pedestrian as soon as its predicted location in the next frame is out of this ROI.

5. Computational Complexity

The complexity of our tracking algorithm depends on the following steps: (1) velocity prediction, (2) clustering, (3) individual and group tracking (4) new pedestrian detection, and (5) leaving pedestrian detection. For n pedestrians, all steps in our algorithm has linear time complexity $O(n)$ except clustering, which is $O(n^3)$. However, the constant term of clustering algorithm is quite small. On a single core 3.0 GHz processor with MATLAB code, it takes about 1-2 seconds to process a frame of 1920×1080 resolution.

6. Experiments

We demonstrate our online multiple human tracking algorithm on the two publicly available datasets: (1) Oxford Town Center [3], and (2) PETS 2009 L1. These datasets have challenges such as inter-person occlusion, cluttered background, linear and non-linear motion, and crowded scenarios. We describe the evaluation measures used for comparison in brief in Section 6.1, while in Section 6.2, we present details of the specific parameters used for each dataset, results, and comparison with other methods.

6.1. Evaluation Measures

We evaluate our tracking results with the standard CLEAR MOT metrics [7] and detection-detection precision and recall. The standard 50% overlap criteria is used for detection-detection precision-recall. The CLEAR MOT metrics includes two metrics: (1) Multiple Object Tracking Precision (MOTP) measures the precision with which objects are located using the intersection of the estimated region with the ground truth region, (2) Multiple Object Tracking Accuracy (MOTA) measures the accuracy of the estimated object regions with the ground truth regions, including false positives, false negatives, and identity switches. For further details of these metrics, the reader is referred to [7].

6.2. Results

Oxford Town Center Dataset [3]: This dataset has a resolution of 1920×1080 at 25 fps and has 4500 (ground-truth) frames with an average of 16 pedestrians visible at any time. We use the following parameters for this dataset: frame window for Kalman filtering $F_{win} = 20$, direction difference weight $A_w = 500$, clustering cut-off threshold $C_{th} = 500$, deformation constant $\lambda = 0.1$. For new pedestrians detection: minimum foreground distance $d_f = 40$, minimum component size $d_c = 300$, and search neighbourhood size $d_t = 100$. We show and compare our results on this dataset in Table 1.

In comparison to other methods, we get better MOTA score. However, the MOTP score is lower. This is because of the extra margin in the Dalal-Triggs human detector that outputs larger bounding box than the ground truth locations. Izadinia et. al. [15] used part-based human detector as the appearance model for tracking, which yields tight bounding boxes, but requires high computation for this frame size. This is why their MOTP score is highest. Our MOTA score is higher because of the occlusion handeling and the group tracking. The deformation constant λ plays a very important role for tracking occluded pedestrians. If there are many instances of occlusion, higher value of λ gives better accuracy. A_w also affects the results in great extent: lower value results in groups that includes people moving in many direction and vice-versa. This affects the tracking and causes large deformation in the tree structure over the time. Tracking group with people moving in a single direction results in better accuracy. Some sample results from this dataset are shown in the first row of Figure 5.

PETS 2009 Dataset L1: This dataset has a resolution of 756×576 and has a total of 795 frames. Annotation for this data is provided by **TUD GRIS** group. We use the following parameters for this dataset: frame window for Kalman filtering $F_{win} = 10$, direction difference weight $A_w = 500$,

Table 1. Quantitative comparison of tracking performance on Oxford Town Center dataset [3] with other methods.

Method	MOTP	MOTA	Prec.	Rec.
Benfold[3]	80.3	61.3	82.0	79.0
Yamaguchi[23]	70.9	63.3	71.1	64.0
Pellegrini[19]	70.7	63.4	70.8	64.1
Zhang[25]	71.5	65.7	71.5	66.1
Leal-Taixe[17]	71.5	67.3	71.6	67.6
Izadinia[15]	71.6	75.7	93.6	81.8
Ours	65.7	76.5	87.1	70.4

Table 2. Quantitative comparison of tracking performance on PETS 2009 L1 dataset with other methods.

Method	MOTP	MOTA	Prec.	Rec.
Breitenstein[8]	59.0	74.0	89.0	60.0
Berclaz[6]	62.0	78.0	78.0	62.0
Conte[10]	57.0	81.0	85.0	58.0
Berclaz[5]	52.0	83.0	82.0	53.0
Alahi[1]	52.0	83.0	69.0	53.0
Izadinia[15]	76.0	93.7	96.8	95.2
Ours	58.1	95.8	83.6	83.1

clustering cut-off threshold $C_{th} = 250$, deformation constant $\lambda = 1.0$. For new pedestrians detection: minimum foreground distance $d_f = 40$, minimum component size $d_c = 200$, and search neighbourhood size $d_t = 50$. We show and compare our results on this dataset in Table 2.

This dataset has low resolution images. So, we had to retrain both appearance models, the full body and head detector, on low resolution images to get better results. However, similar to the previous dataset, we get better MOTA score than other methods, but we get lower MOTP score because our appearance model outputs larger bounding box than the ground truth locations. Some sample results from this dataset are shown in the second row of Figure 5.

7. Conclusion

We proposed an online multiple pedestrian tracking algorithm that utilizes group behaviour using minimum spanning trees (MST) and performs tracking after dividing pedestrians into several groups. We used position and velocity of pedestrians as features for agglomerative hierarchical clustering algorithm with our own custom metric for clustering pedestrians into compact groups, and we used the pictorial structures method to track each group MST optimally. We handled inter-pedestrian occlusions using a custom trained head detector. Also, we proposed an efficient method to detect newly entered pedestrians in the frame parallel to tracking with help of background subtraction. We performed experiments on two challenging publicly available datasets and showed improvements on multiple object tracking accuracy (MOTA) over other methods.

References

[1] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst. Sparsity-driven people localization algorithm: Evaluation in crowded scenes environments. In *Proc. PETS-Winter, 2009*. IEEE, 2009.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proc. CVPR 2008*. IEEE, 2008.

[3] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Proc. CVPR 2011*. IEEE, 2011.

[4] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *Proc. CVPR 2006*. IEEE, 2006.

[5] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Proc. PETS-Winter, 2009*. IEEE, 2009.

[6] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Proc. PAMI 2011*, 2011.

[7] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *Proc. EURASIP Journal on Image and Video Processing*, 2008.

[8] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Proc. PAMI 2011*, 2011.

[9] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time euclidean distance transform algorithms. *Proc, PAMI 1995*, 1995.

[10] D. Conte, P. Foggia, G. Percannella, and M. Vento. Performance evaluation of a people tracking system on pets2009 database. In *Proc. AVSS 2010*. IEEE, 2010.

[11] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. D. Bue, G. Menegaz, and V. Murino. Social interaction discovery by statistical analysis of f-formations. In *Proc. BMVA 2011*. British Machine Vision Association, 2011.

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR 2005*. IEEE, 2005.

[13] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR 2008*, 2008.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV 2005*, 2005.

[15] H. Izadinia, I. Saleemi, W. Li, and M. Shah. (mp)2t: Multiple people multiple parts tracker. In *Proc. ECCV 2012*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.

[17] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *Proc. ICCV Workshops 2011*. IEEE, 2011.

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

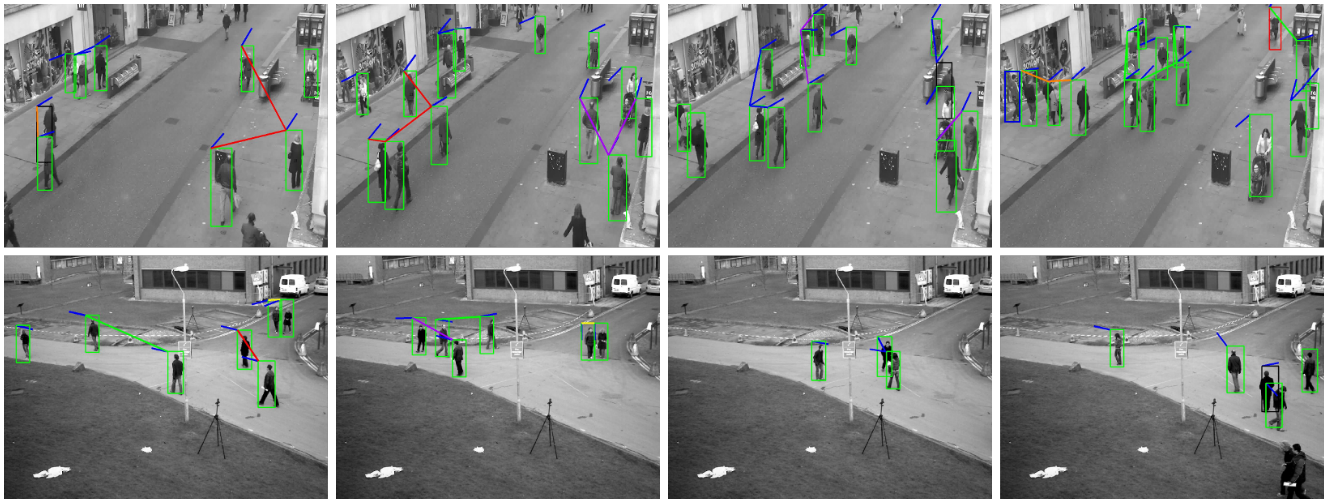

Figure 5. Sample results from Oxford Town Center dataset [3] in first row, and from PETS 2009 L1 dataset in second row. The green rectangles show pedestrians being tracked, the red show leaving pedestrians, the black show occluded, and the blue show newly entered pedestrians.

[19] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. ICCV 2009*. IEEE, 2009.

[20] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.

[21] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *Proc. CVPR 2012*. IEEE, 2012.

[22] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. CVPR 1999*. IEEE, 1999.

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

[24] B. Yang and R. Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Proc. ECCV 2012*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012.

[25] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proc. CVPR 2008*. IEEE, 2008.

[26] L. Zhang and L. van der Maaten. Structure preserving object tracking. In *Proc. CVPR 2013*. IEEE, 2013.

[27] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. CVPR 2012*. IEEE, 2012.