Multi Object Tracking with UAVs using Deep SORT and YOLOv3 RetinaNet Detection Framework

Shivani Kapania
Bharati Vidyapeeth's College of Engineering
New Delhi, India
shivani.kapania@gmail.com

Dharmender Saini
Bharati Vidyapeeth's College of Engineering
New Delhi, India
dharmender.saini@bharatividyapeeth.edu

Sachin Goyal
Bharati Vidyapeeth's College of Engineering
New Delhi, India
sachingoyal2910@gmail.com

Narina Thakur
Bharati Vidyapeeth's College of Engineering
New Delhi, India
narina.thakur@bharatividyapeeth.edu

Rachna Jain
Bharati Vidyapeeth's College of Engineering
New Delhi, India
rachna.jain@bharatividyapeeth.edu

Preeti Nagrath
Bharati Vidyapeeth's College of Engineering
New Delhi, India
preeti.nagrath@bharatividyapeeth.edu

ABSTRACT

Over the years, object tracking and detection has emerged as one of the most important aspects of UAV applications such as surveillance, reconnaissance, etc. In our paper, we present a tracking-by-detection approach for real-time Multiple Object Tracking (MOT) of footage from a drone-mounted camera. Tracking-by-detection is the leading paradigm considering its computational effectiveness and improved detection algorithms. Our algorithm builds on the baseline Deep SORT algorithm implemented for MOT benchmarks. However, to circumvent the challenges posed by videos captured from a significant height we use a combination of YOLOv3 and RetinaNet for generating detections in each frame. The results of our experiment on the VisDrone 2018 dataset exhibit competitive performance in comparison to the existing trackers.

CCS CONCEPTS

• **Computing methodologies** → **Tracking**; *Object detection*; *Neural networks*.

KEYWORDS

Unmanned aerial vehicles, object tracking, object detection, neural networks

ACM Reference Format:
Shivani Kapania, Dharmender Saini, Sachin Goyal, Narina Thakur, Rachna Jain, and Preeti Nagrath. 2020. Multi Object Tracking with UAVs using Deep SORT and YOLOv3 RetinaNet Detection Framework. In *The 1st ACM Workshop on Autonomous and Intelligent Mobile Systems (AIMS '20), January 11, 2020, Bangalore, India.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3377283.3377284

AIMS '20, January 11, 2020, Bangalore, India

ACM ISBN 978-1-4503-7666-2/20/01…$15.00
https://doi.org/10.1145/3377283.3377284

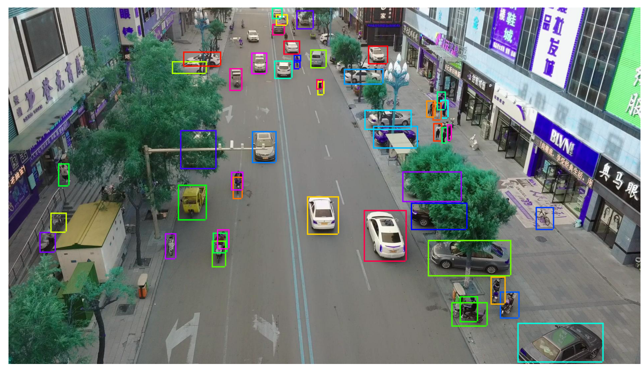

Figure 1: Frame from the VisDrone 2018 Multi Object Tracking dataset

1 INTRODUCTION

Applications of object tracking are becoming increasingly ubiquitous day-by-day, partly due to the vast (and ongoing) research in machine learning techniques, especially, convolutional neural networks – pothole detection, people counting to generate crowd statistics, automatic detection of vehicle number plates, facial recognition at security checkpoints, to name a few. Similarly, Object detection/tracking using unmanned aerial vehicles has also been gaining increasing interest from the Mobile Systems and Computer Vision community alike for its applications in search-and-rescue, surveillance, reconnaissance, and others. Most of these studies require detecting single/multiple pedestrians, vehicles, infrastructural components, etc.

Multiple Object Tracking (MOT) involves estimating the trajectory of several objects simultaneously over time in a series of video frames. This can be performed in an online or offline mode. Online object tracking is particularly useful for real-time applications because only detections from the previous and the current frame are available to the tracker, and thus requires significant computational speed for more complicated algorithms. In such scenarios, tracking-by-detection has emerged as the leading paradigm for MOT which uses an object detection algorithm to start, update or terminate a tracker. Simple Online And Realtime Tracking (SORT) [4] is one such relatively simple tracking-by-detection algorithm. It uses a combination of Kalman Filter and Hungarian algorithm to handle motion prediction and data association respectively, but ignores the appearance features in the association matrix, for its simplicity and speed, thus increasing the number of identity switches. To account for this loss in performance, Deep SORT [43] integrates object appearance information in an association matrix.

Although, object tracking has been researched for decades, a lot of challenges persist – noise in an image, difficult object motion, variation in illumination, object occlusion, complex object structures, and the loss of evidence caused by an estimate of the 3D realm on a 2D image. Furthermore, object tracking with aerial vehicles poses additional problems. Since the video is captured from a significant height, the objects (such as pedestrians and vehicles) in each frame are proportionately smaller – rendering most object detection algorithms practically ineffective.

Our contribution is as follows: an implementation of Deep SORT algorithm combined with a detection framework made of YoloV3 [36] and RetinaNet [40], on the VisDrone 2018 dataset. The "Vision Meets Drones" is a large-scale visual object tracking and detection benchmark [52] collected in China. For our work, we use the MOT portion captured by the drone-mounted cameras in a diverse range of scenarios. The dataset contains a total of 56 clips (24201 frames) for training, 7 clips (2819 frames) for validation and 16 clips (6333 frames) for testing.

2 RELATED WORK

Computer vision researchers have recognized object tracking as a crucial task with applications pertaining but not limited to human-computer interaction, automated surveillance, traffic monitoring, etc. A vast majority of the *offline learning* [7, 11, 23, 25, 31, 35] methods use a graph-based representation to establish MOT as a global optimization problem. Offline models can access past, as well as future frames from the entire video to extract information. The challenges in object detection are viewed as an optimization problem with an aim to minimize the global loss function. Offline models are likely to deliver better performance due to greater information access. On the other hand, *online learning* techniques solve the data association problem either determinatively (greedy association [7] or Hungarian algorithm [28]) or probabilistically [19, 32, 33], whose main component is a similarity function between detections and targets. It receives video input on a frame-by-frame basis and produces an output corresponding to each frame. Therefore, the input information is obtained from only current and past frames. This makes online models more suitable for real-time videos.

Although there has been extensive research on this topic, many visual object trackers experience difficulty in handling changes in appearances of the objects due to frequent occlusion, camera motion, and variation in illumination. The Kanade-Lucas-Tomasi algorithm generates useful local features for tracking. After deriving local features, we can apply them to multiple tasks such as estimating camera motion [3, 13], motion clustering [10], generating short trajectories [39, 49], and so on.

A global visual representation indicates the global statistical characteristics of object appearance. This can be modeled using raw pixels, optical flow, histograms, and active contours. Optical flow characterizes a dense field of displacement vectors for each pixel within an image patch. It is widely used in visual tracking algorithms for encoding motion information [12, 42], data association [18, 38, 44], and discovering crowd motion pattern in situations where ordinary features may be unreliable due to frequent occlusions. Region-based features may be of three types: zero-order which includes raw pixel representation [45], and color histogram [18, 27, 33]. The histogram of oriented gradients [6, 14, 21, 48] feature descriptor is also commonly used.

Typical appearance models may employ single [1, 34, 45] or multiple cues. Some significant work has been done to optimize multiple cue integration models. Foreground and background approximations are done based on previous and present data obtained. Optimization algorithms are put to use to improve the appearance model for classification margin optimization. Models utilizing multiple cues may be categorized according to their fusion strategy as follows:

- Boosting: These appearance models [21, 23, 46] select a portion of features from a pool of candidate features using a boosting-based algorithm using cues such as shape, covariance matrix, HOG, color, etc.
- Concatenation: Features such as optical flow, color, HOG, etc may be concatenated for appearance modeling [8].
- Summation: Appearance models [24, 26] may fuse cues such as LBP, correlogram, depth etc.

While approaches such as Multiple Hypothesis Tracking (MHT) [37] or the Joint Probabilistic Data Association (JPDA) [16] filters have remained popular for offline tracking, they delay decision-making until there is low uncertainty about assignments of detections to tracklets. MHT calculates probabilities that a particular measurement corresponds to a previously known target. Kalman filter [9] estimates the target states for the next time step. As the tracker receives more measurements, it recursively calculates the joint probabilities of the hypotheses by including information such as location uncertainty, the density of unknown and false targets. This approach allows for correlating measurements with the target based on past and subsequent data.

The quality of the detection algorithm is a crucial aspect of all tracking-by-detection models. [4, 44] highlighted the dependency of the tracker on the accuracy of the detection model. Conversely, this may reduce the performance in real-time object tracking. To avoid this problem, tracking is split into detection, prediction, and association of objects between the frames. Thus, [2] uses a pre-trained support vector machine (SVM) and optical flow-like equations to detect vehicles and associate detections among the frames. Bochinski et al. [5] presented a simple IOU tracker, whose performance increases with higher frame rates and increasing computational power. [22] formulated a Bayesian filtering framework conducted by a changing point detection algorithm that uses a KLT based motion detector to compute the foreground regions as detections in case of occlusion and drifts. Tjaden et al. [41] proposed an algorithm for real-time pose tracking of 3D objects. It uses the Gauss-Newton optimization scheme to optimize the region based cost functions, which was derived initially from local color histograms. Nam et al. [29] use a tree structure to model and propagate multiple CNNs, where multiple CNNs collaborate to determine the target state for

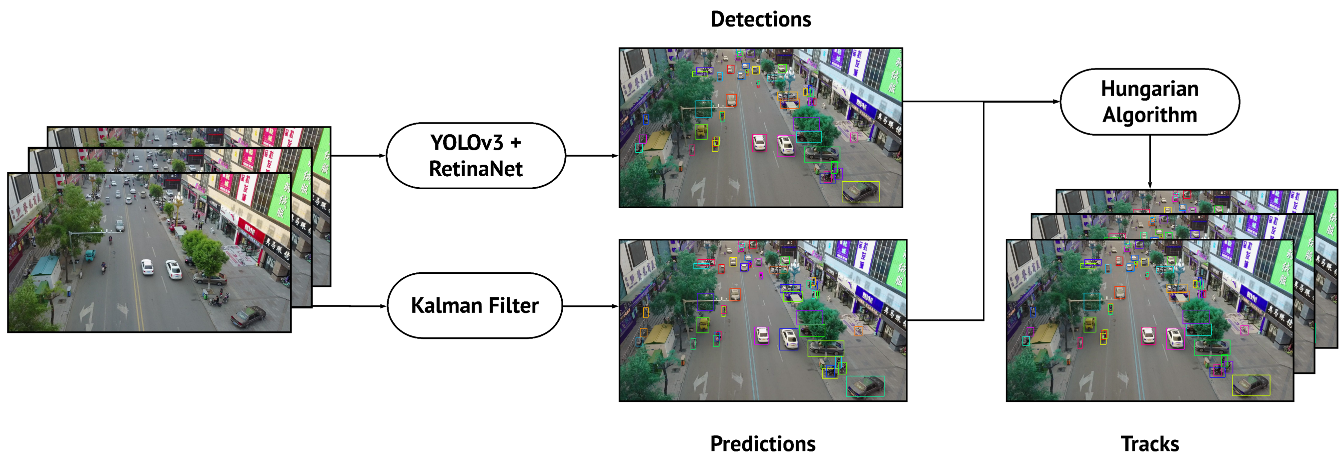

Figure 2: Our model's architecture

updating the paths in consecutive frames. An accurate understanding of the environment is necessary for tracking, and this becomes a more complicated task with moving cameras. Dias et al. [15] presented a solution for real-time multi-object tracking in highly dynamic environments. It can be used to plan tasks and control an autonomous robot. [47] proposed an approach to remove the effects of unexpected camera motion by using the motion contexts from multiple objects present in the frame. They constructed a Relative Motion Network (RMN) using the relative movement between the objects in the frame.

Object detection using unmanned aerial vehicles suffers from its own set of challenges along with the inherited challenges of object detection. With the availability of faster and cheaper computing power, various advancements have taken place in this research area. Tracking and capturing the geolocation of a moving vehicle in real-time [50] has shown promise in the automatic supervision of a UAV. To tackle a large amount of visual data generated by drones [17] proposed a method to filter the relevant frames using a content-based segmentation technique, especially for construction sites. Furthermore, [20] proposed a method for early detection of forest fires and smoke by using onboard processing capability for a fixed and rotatory wing drone.

Another challenge, of associating noisy object detections to an existing track is handled by Markov Decision Process(MDP) [44]. Here, every detected object is modeled with a new MDP with four states Active, Tracked, Inactive and lost, thus making this challenge a decisive task for every object based on its current state and the learnt policy, via inverse reinforcement learning during its exhaustive training. It uses optical flow for predicting the future object track while they are in tracked state, while it uses association matrix to re-assign tracks.

3 METHODOLOGY

The effectiveness of any tracking-by-detection model depends primarily on its detection algorithm. By improving detections, it becomes easier for the data association technique to associate the newly generated detections to the existing tracks. To address the complexities associated with multi-object tracking by UAVs (at a considerable height), we used a detection framework which is a combination of YOLOv3 and RetinaNet. YOLOv3 is fast and performs well on usual objects, but is not suitable for small-sized and denser objects. On the other hand, RetinaNet performs well especially in cases where objects are small in size and are present in clusters. This framework returns all the detections from a given frame. The redundant detections were removed using non-max suppression (NMS) [30] giving a set of all the bounding boxes possible, which consist of all the new located objects in this frame.

The detections from each frame were fed into a pre-trained CNN model [43] on a person re-identification dataset (re-id). The Motion Analysis And Re-identification (MARS) [51] dataset contains 1,261 IDs with 200,000 tracklets. To create this dataset, they used six synchronized cameras and any pedestrian captured by any two cameras is added into the dataset. This model generates a deep association matrix related to each detection, which incorporates the appearance features of the objects. These appearance features were combined with the motion information of the detected objects in the matrix. This well-discriminating feature embedding is useful in tracking objects after a state of short term or long term occlusions by assigning the same identity to the object after the occlusion.

We used a Kalman Filter [9] to optimally estimate the state variables during motion. Kalman filter is a set of mathematical equations that estimate the state of a process, even if the precise location of the modeled system is unknown []. The state of the target (equation 1.) as the directly observed values is supplied to the Kalman Filter to predict the location of the target in the next frame. The tracked targets are represented by:

$$x = \left[u, v, s, r, \dot{u}, \dot{v}, \dot{s} \right]^T \tag{1}$$

Here u and v denote the horizontal and vertical pixel locations of the center of the target, and h and γ indicate the height and the aspect ratio respectively. Corresponding values $(\dot{u}, \dot{v}, \dot{s})$ denote the respective velocities in image coordinates of the base variable values.

Tracker	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓
DeepSORT_Y+RN*	45.8	0.219	73.6	266	348	13873	56741	802	2089
SORT	40.2	0.251	56.1	297	514	11838	74027	265	1380
DeepSORT	42.6	0.259	58	323	395	14722	68060	779	3717
GOG_EOC	36.9	0.242	46.5	205	589	5445	86399	354	1090
SCTrack	35.8	0.244	45.1	211	550	7298	85623	798	2042

Table 1: Evaluating the performance on VisDrone 2018 dataset

To find the equation governing the state, assume that we have a control signal $u \in R^l$ and a state $x \in \mathbb{R}^n$:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{2}$$

where w_k is the process noise, A is the state transition model and B is the control-input model. Moreover, the state is related to the observable variable $z \in \mathbb{R}^m$, but not directly and fully measurable,

$$z_k = Hx_k + v_k \tag{3}$$

where v_k is the observation noise and H is the observation model. Assume that the random variables w_k and v_k are normally distributed with zero mean and covariance Q and R respectively and independent. The optimal state estimate $\hat{x}$ is computed by the Kalman filter by recursively consolidating previous estimates with new observations. It consists of two well-defined phases: *predict*, during which we compute the optimal state $\hat{x}_k^-$ prior to observing z_k; and *update*, where optimal posterior state $\hat{x}_k$ is computed after observing z_k.

Data association is the task of determining which detection corresponds to which prediction of an object from the previous frame, or alternatively if this detection represents a new object. Inaccuracies arise when new objects enter the frame, tracked objects are not detected or they exit the frame. In this case, the detection algorithm may produce *false positives* – i.e. "detecting" an object that does not exist – or when the predicted positions differ greatly from the actual positions.

Let each detection response be, $x_i = (x_i, s_i, a_i, t_i)$, where x_i is the position, a_i is the appearance, s_i is the scale, and t_i is the frame number/timestep of the object. And, $\mathcal{X} = x_i$ be a set of object observations. An ordered list of object observations, i.e. $T_k = x_{k1}, x_{k2}, ..., x_{k_i}$ constitutes a single trajectory hypothesis, where $x_{k_i} \in \mathcal{X}$. An association hypothesis $\mathcal{T}$ is defined as a set of single trajectory hypotheses, i.e. $\mathcal{T} = T_k$. The objective of data association is to maximize the posterior probability of $\mathcal{T}$ given the observation set $\mathcal{X}$:

$$\mathcal{T}^* = \mathcal{T} P(\mathcal{T} \mid \mathcal{X}) \tag{4}$$

$$\mathcal{T}^* = \mathcal{T} P(\mathcal{X} \mid \mathcal{T}) P(\mathcal{T}) \tag{5}$$

$$\mathcal{T}^* = \mathcal{T} \prod_i P(x_i \mid \mathcal{T}) P(\mathcal{T}) \tag{6}$$

assuming that the likelihood probabilities are conditionally independent given the hypothesis $\mathcal{T}$.

To this end, we employed the Hungarian algorithm to optimally associate the bounding boxes (detections) with the existing tracks (predictions). The assignment cost matrix is computed as Intersection-Over-Union (IoU) distances with the purpose of maximizing the overlap between predictions and detections. With a time complexity of $O(n^3)$ where n is the number of agents (and tasks), the Hungarian algorithm solves the assignment problem in polynomial time. The input to the algorithm is a cost matrix C, where $C(i, j)$ is the cost c_{ij}. To compute the cost matrix C between predictions and detections, we have to define what it means for two bounding boxes to be (dis-)similar. A common technique is to compute the Jaccard index, also known as Intersection over Union (IoU), between two bounding boxes A and B as

$$IoU = \frac{| A \cap B |}{| A \cup B |} \tag{7}$$

The detections with an IoU score less than IoU_{min} are neglected. A new id is assigned to every new tracked object. To minimise false positives, detections must be associated with tracks for a threshold number of times or the new objects are not tracked. Similarly, if the object is not associated with any detection for T_{lost} frames, then the tracklet corresponding to that object is terminated. This is unless it reappears after a period of occlusion, in which case, the same id may be reassigned to that object.

4 EXPERIMENTS

The detection framework composed of YOLOv3 and RetinaNet separately obtains detections within a frame and the redundant detections were removed by applying non-max suppression. These detections (vectors of length 10) were passed to the pre-trained CNN model to incorporate appearance information. The CNN used to generate the deep association matrix was pre-trained on the MARS dataset. It generated a NumPy file of vectors containing 138 values each containing appearance and motion information encoded in them, which is used for tracking and re-identification of an object after long and short term occlusions. The Kalman filter takes this vector as input to predict the location of the bounding boxes in the future frames. The Hungarian algorithm used the *IOU* score to generate an assignment cost matrix to associate predicted bounding boxes with previously generated tracks.

Training of the complete model on an NVIDIA Tesla P100 GPU took about 14-16 hours. We evaluated the performance of our tracker on a diverse set of sequences in the VisDrone dataset [52]. For tuning the initial Kalman filter covariances, IoU_{min}, and T_{Lost} parameters, we use the same training/validation split.

4.1 Evaluation

Considering that it is difficult to use a single score to evaluate the performance of a multi-object tracker, we utilized the standard MOT evaluation metrics and report these for our implementation and other baseline trackers. To evaluate our implementation against other trackers, we used the py-motmetrics library which supports CLEAR-MOT metrics and ID metrics. Py-motmetrics tracks all the relevant per-frame events such as correspondences, misses, false alarms and switches. We report the following MOT metrics. The **MOTA** (multi-object tracking accuracy) combines three error sources, i.e., false positive, false negative, and identity switches. **MOTP** (multi-object tracking precision) is the mean dissimilarity between ground truths and all true positives. **IDF1** represents the global minimum cost F1 score. The **MT** (mostly tracked trajectories) and **ML** (mostly lost trajectories) metrics measure the number of tracked objects less than 20% and more than 80% of the life span based on the ground truth respectively. Both the **IDS** (identity switches) and **FM** (number of fragmentations) describe the accuracy of the tracker to follow object trajectories. Identity switches indicate the number of times that the matched identity of a tracked trajectory changes from one id to another, while FM is the number of times that the trajectories are disconnected, that is, from tracked to not tracked.

5 CONCLUSION

In this paper, we present a multiple object tracker with an improved object detection framework comprising of YOLOv3 and RetinaNet. RetinaNet detects objects from a significant height more accurately, as YOLO performs sub-optimally in cases where objects are of smaller size and are in clusters. As demonstrated in SORT and keeping in line with Occam's Razor, we select a simple filter (for motion prediction) and data association algorithm. The deep association matrix is generated by a CNN model pre-trained on the MARS dataset. Incorporating appearance features in the deep association matrix along with the motion information improves the accuracy of the trajectories by reducing the number of fragmentations and identity switches. This allows for re-identification in cases of short and long term occlusions. Generating tracks during online tracking requires fast computation and easy-to-run algorithms. As evidenced by experiments, the quality of detection remains extremely important. Thus, future work may investigate the trade-off between performance and speed in online tracking by training the tracker in offline mode for the initial optimization of its parameters.

REFERENCES

[1] Saad Ali and Mubarak Shah. 2008. Floor fields for tracking in high density crowd scenes. In *European conference on computer vision*. Springer, 1–14.
[2] Shai Avidan. 2004. Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence* 26, 8 (2004), 1064–1072.
[3] Ben Benfold and Ian Reid. 2011. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*. IEEE, 3457–3464.
[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3464–3468.
[5] Erik Bochinski, Volker Eiselein, and Thomas Sikora. 2017. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 1–6.
[6] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. 2009. Robust tracking-by-detection using a detector confidence particle filter. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 1515–1522.
[7] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. 2011. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE transactions on pattern analysis and machine intelligence* 33, 9 (2011), 1820–1833.
[8] William Brendel, Mohamed Amer, and Sinisa Todorovic. 2011. Multiobject tracking as maximum weight independent set. In *CVPR 2011*. IEEE, 1273–1280.
[9] Ted J Broida and Rama Chellappa. 1986. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1 (1986), 90–99.
[10] Gabriel J Brostow and Roberto Cipolla. 2006. Unsupervised bayesian detection of independent motion in crowds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 1. IEEE, 594–601.
[11] Asad A Butt and Robert T Collins. 2013. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1846–1853.
[12] Wongun Choi. 2015. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*. 3029–3037.
[13] Wongun Choi and Silvio Savarese. 2010. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *European Conference on Computer Vision*. Springer, 553–567.
[14] Wongun Choi and Silvio Savarese. 2012. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*. Springer, 215–230.
[15] Ricardo Dias, Bernardo Cunha, Eduardo Sousa, José Luís Azevedo, João Silva, Filipe Amaral, and Nuno Lau. 2017. Real-time multi-object tracking on highly dynamic environments. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 178–183.
[16] Thomas E Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. 1980. Multi-target tracking using joint probabilistic data association. In *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*. IEEE, 807–812.
[17] Youngjib Ham and Mirsalar Kamari. 2019. Automated content-based filtering for enhanced vision-based documentation in construction toward exploiting big visual data from drones. *Automation in Construction* 105 (2019), 102831.
[18] Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah. 2012. 2 T: multiple people multiple parts tracker. In *European Conference on Computer Vision*. Springer, 100–114.
[19] Zia Khan, Tucker Balch, and Frank Dellaert. 2005. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE transactions on pattern analysis and machine intelligence* 27, 11 (2005), 1805–1819.
[20] Diyana Kinaneva, Georgi Hristov, Jordan Raychev, and Plamen Zahariev. 2019. Early Forest Fire Detection Using Drones and Artificial Intelligence. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 1060–1065.
[21] Cheng-Hao Kuo, Chang Huang, and Ramakant Nevatia. 2010. Multi-target tracking by on-line learned discriminative appearance models. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 685–692.
[22] Byungjae Lee, Enkhbayar Erdenee, Songguo Jin, Mi Young Nam, Young Giu Jung, and Phill Kyu Rhee. 2016. Multi-class multi-object tracking using changing point detection. In *European Conference on Computer Vision*. Springer, 68–83.
[23] Yuan Li, Chang Huang, and Ram Nevatia. 2009. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2953–2960.
[24] Ye Liu, Hui Li, and Yan Qiu Chen. 2012. Automatic tracking of a large number of moving targets in 3d. In *European Conference on Computer Vision*. Springer, 730–742.
[25] Anton Milan, Stefan Roth, and Konrad Schindler. 2014. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence* 36, 1 (2014), 58–72.
[26] Dennis Mitzel, Esther Horbert, Andreas Ess, and Bastian Leibe. 2010. Multi-person tracking with sparse detection and continuous segmentation. In *European Conference on Computer Vision*. Springer, 397–410.
[27] Dennis Mitzel and Bastian Leibe. 2011. Real-time multi-person tracking with detector assisted structure propagation. In *2011 IEEE International Conference on*

Computer Vision Workshops (ICCV Workshops). IEEE, 974–981.

[28] James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics* 5, 1 (1957), 32–38.

[29] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. 2016. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242* (2016).

[30] Alexander Neubeck and Luc Van Gool. 2006. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 3. IEEE, 850–855.

[31] Juan Carlos Niebles, Bohyung Han, and Li Fei-Fei. 2010. Efficient extraction of human motion volumes by tracking. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 655–662.

[32] Songhwai Oh, Stuart Russell, and Shankar Sastry. 2009. Markov chain Monte Carlo data association for multi-target tracking. *IEEE Trans. Automat. Control* 54, 3 (2009), 481–497.

[33] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. 2004. A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*. Springer, 28–39.

[34] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 261–268.

[35] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*. IEEE, 1201–1208.

[36] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[37] Donald Reid. 1979. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control* 24, 6 (1979), 843–854.

[38] Mikel Rodriguez, Saad Ali, and Takeo Kanade. 2009. Tracking in unstructured crowded scenes. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 1389–1396.

[39] Daisuke Sugimura, Kris M Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. 2009. Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 1467–1474.

[40] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. *arXiv preprint arXiv:1902.09212* (2019).

[41] Henning Tjaden, Ulrich Schwanecke, Elmar Schömer, and Daniel Cremers. 2018. A region-based gauss-newton approach to real-time monocular multiple object tracking. *IEEE transactions on pattern analysis and machine intelligence* (2018).

[42] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. 2010. New features and insights for pedestrian detection. In *2010 IEEE Computer society conference on computer vision and pattern recognition*. IEEE, 1030–1037.

[43] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3645–3649.

[44] Yu Xiang, Alexandre Alahi, and Silvio Savarese. 2015. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*. 4705–4713.

[45] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. 2011. Who are you with and where are you going?. In *CVPR 2011*. IEEE, 1345–1352.

[46] Bo Yang and Ram Nevatia. 2012. Online learned discriminative part-based appearance models for multi-human tracking. In *European Conference on Computer Vision*. Springer, 484–498.

[47] Ju Hong Yoon, Ming-Hsuan Yang, Jongwoo Lim, and Kuk-Jin Yoon. 2015. Bayesian multi-object tracking using motion context from multiple objects. In *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 33–40.

[48] Ting Yu, Ying Wu, Nils O Krahnstoever, and Peter H Tu. 2008. Distributed data association and filtering for multiple target tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

[49] Xuemei Zhao, Dian Gong, and Gérard Medioni. 2012. Tracking using motion patterns for very crowded scenes. In *European Conference on Computer Vision*. Springer, 315–328.

[50] Xiaoyue Zhao, Fangling Pu, Zhihang Wang, Hongyu Chen, and Zhaozhuo Xu. 2019. Detection, Tracking, and Geolocation of Moving Vehicle From UAV Using Monocular Camera. *IEEE Access* 7 (2019), 101160–101170.

[51] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. 2016. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*. Springer, 868–884.

[52] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. 2018. Vision meets drones: a challenge. *arXiv preprint arXiv:1804.07437* (2018).