Multiple Object Tracking by Bounding Boxes Without Using Texture Information and Optical Flow

Ágnes Lipovits, László Czúni, Katalin Tömördi, Zsolt Vörösházi
Image Processing Laboratory
Faculty of Information Technology, University of Pannonia
Egyetem street 10.
Hungary (H-8200), Veszprém
lipovitsa@almos.uni-pannon.hu

ABSTRACT
Object tracking is a key task in many applications using video analytics. While there is a huge number of algorithms to track objects, there is still a need for new methods to solve the correspondence problem under certain circumstances. In our article, we assume a very typical but still open scenario: a still image object detector has already identified the objects to be tracked; thus, we have object labels, confidence values, and bounding boxes in each video frame captured at a low sampling rate. That is, optical flow methods difficult to be applied (also due to bad lighting conditions, cluttered or homogeneous areas and strong ego-motion), and moreover, many objects look similar (having the same category labels). Our proposed approach is based on the Hungarian method and incorporates the above information into the cost function evaluating the possible pairings of objects. To consider the uncertainty of the detector, the elements of the confusion matrix also contribute to the cost of pairs, as well as the probability of spatial translations based on prior observations. As a use case, we apply the algorithm to a data-set, where images were captured from onboard cameras and traffic signs were detected by RetinaNet. We analyze the performance with different parameter settings.

Keywords
Object tracking, object detection, Hungarian method, RetinaNet

1 INTRODUCTION

While the goal of all object tracking algorithms is to estimate the trajectory of moving objects, there are significant differences between their nature, considering their prerequisites and tolerance against the different visual condition. For example, while many algorithms heavily rely on the optical flow or velocity and find the corresponding areas using the analysis of image features of candidate regions (e.g. [Bewley2016], [Ma2019]), others try to involve texture-based object detection more deeply (for example [Ning2017] and [Wang2019]). When the camera is moving in a 3D space, and the image features of the relevant objects to be tracked are similar, multiple object tracking becomes very difficult and even can require efforts to neutralize camera motion [Czúni2012]. That is a reason that under such circumstances, the second approach seems much more appealing.

In our article, we process outdoor videos captured from onboard cameras, and different objects, mostly traffic signs, are to be detected and tracked. Besides the problems encountered from various factors, such as the cluttered environment, possible occlusions, scale variation, image noise, and low contrast, the temporal sampling rate may be too low in the case of high-speed ego-motion. Moreover, the same object classes often have multiple appearances; thus, optical flow methods (e.g. CAMShift, Kanade-Lucas-Tomasi, Horn-Schunck, etc.) can not be applied efficiently. Fig. 1 illustrates when the car is turning, and the optical flow (generated by [Farneback2003]) looks turbulent due to the rotational and translational motion of the camera, the complicated structure of the 3D scene, low contrast regions, and camera independent object motion.

We assume a very typical but still open processing pipeline scenario for object tracking: a still image object detector has already identified the objects to be tracked; thus, we have object labels, confidence values, and bounding boxes in each video frame captured at a given sampling rate. The task is to find the correspondence between the bounding boxes (or declare there is no such match).

Our proposed approach is based on the Hungarian method [Kuhn1955] and incorporates the above information into the cost function evaluating the possible pairings of objects. To consider the uncertainty of the detection, the elements of the confusion matrix (error matrix) of the detector also contribute to the cost of pairs, as well as the probability of spatial translations based on prior observations.

In the next section, we overview related papers, then

Figure 1: Optical flow computed by [Farneback2003] when the car is turning.

in Section 3 we describe the data-set and the object detection method. Section 4 contains the theoretic details of our proposal, while in Section 5 we analyze the performance with different parameter settings.

2 ABOUT TRACKING METHODS

There are many aspects to categorize the large number of approaches such as stereo or monocular, model-based or model-free, multiple target or single-target, casual or bi-directional, and long-term or short-term trackers.

Since there is a larger variety and number of tracking algorithms, we can not give a comprehensive overview in our short article. We focus on those found the most similar or the recent ones lacking optical flow calculations but using neural networks.

[Huang2008] describes a three-level hierarchical tracking method for multiple objects: at low-level, short tracks are generated for further analysis; at the middle-level, these tracks are further processed to form longer trajectories based on the Hungarian method; while at the highest level, a scene structure model (including three maps for entries, exits and scene occluders) is created. This high-level step implements scene knowledge-based reasoning to reduce trajectory fragmentation and prevent possible identity switches. The method only uses colour histograms, position and size, no other information from the detector itself.

[Henriques2011] also applies the Hungarian method but focuses on the ability to model multiple objects that are merged into a single measurement and track them as a group. The solution is based on a graph structure that encodes these multiple-match events. Since object identities are lost when objects merge, the problem of tracking individual objects across groups is posed as a standard optimal assignment problem.

A good recent example to combine different modalities in the tracking-by-detection domain is [Karunasekera2019], where a dissimilarity measure based on object motion, appearance (colour histogram), structure (Local Binary Pattern), and size was used; assignment is solved by the Hungarian method.

In contrast to [Karunasekera2019], in [Bewley2016] an efficient, while relatively simple and lightweight tracker, for multiple objects with constant velocity, was described using variations of Faster Region CNNs, and the plain old Kalman-filter and the Hungarian method. The main idea was to use the very efficient CNNs for the detection of objects, instead of using hand-crafted visual features of classical appearance-based trackers. Not surprisingly, it was found that the detection quality had a significant impact on tracking performance. Our proposal is also a simple tracker, but we incorporate the properties of detected objects (namely the confidence and the probability of confusions) into the solution of the correspondence problem. Moreover, since our tracker handles velocity in a probabilistic manner, linear or constant motion is not assumed.

There are approaches where both the spatial and temporal domains of tracking are fused by deep neural networks. For example [Ning2017] proposes a method to extend the deep neural network learning and analysis into the spatio-temporal domain by combining Yolo and LSTM networks. The spatially and temporally deep method applies regression and can effectively tackle problems of occlusions and motion blur. [Jiang2018] also applies Yolo and LSTMs but in a very different way. Each object has its own tracker, regarded as an agent, trained by utilizing deep reinforcement learning, where LSTM is used to predict parameters (motion and scale) of the observed object. Data association between the output of Yolo and the trackers is also done by an LSTM. The drawback of this technique is that besides Yolo the LSTMs should also be trained according to the environments. [Wang2019] introduces SiamMask performing both real-time visual object tracking and semi-supervised video object segmentation. Once trained, it relies on a single bounding box initialization and operates online, and can produce object segmentation masks and rotated bounding boxes at high-speed. Unfortunately, both methods can handle only a single object at a time. While most approaches follow a sequential strategy as detection then tracking, some new DNNs try to fuse the two steps. For example in [WangZ2019] a Feature Pyramid Network is used to find objects with their bounding boxes with a constraint that the distance between observations of the same identity in consecutive frames should be smaller than the distance between different identities. Unfortunately, this does not hold in many cases for our videos.

For readers with more interest to overview this field we propose to check earlier papers [Cannons2008],

[Smeulders2013] or [Fiaz2019] published more recently.
Our proposed solution will be a monocular, model-free, multiple target, causal, and short-term tracker. What is more important that we are not applying optical flow (only accumulated statistics about the motion of the camera), can utilize any object detection mechanism, which produces bounding boxes and confidence values, making our algorithm well-suited, as a post processing step, in many image processing pipelines.

3 THE BENCHMARK DATA-SET

Automotive applications require fast and accurate detection of street objects, traffic signs are among the key elements. While the detection and classification of traffic signs on some existing data-sets can reach good performance [Lim2017], there are many circumstances where results are still poor [Temel2019]. Also the number of classes is not really limited since there are many composite objects or unique designs which make existing methods easily fail.

3.1 The Hun158 and Hun169 Data-sets

The Hun158 benchmark data-set contains 158 different classes of Hungarian road traffic signs and some typical street objects (e.g. bus stops, dust-bins, etc.). It comprises pixel-wise segmented objects on 3440 image frames of size 1280×720 and 1920×1080 resolutions. The number of annotated objects is 13300, each has at least 20 appearances. The Hun169 data-set contains 20 videos of 37462 frames with 61274 annotated objects from 168 classes. In Hun169 objects are denoted by bounding boxes, videos were recorded at 15 and 30 FPS. There is no overlapping between Hun158 and Hun169.

3.2 Object Detection

On all the frames of Hun158 we computed the bounding boxes and trained the RetinaNet [Lin2017]. See Fig. 2 for the illustration of two subsequent frames with detected objects.

RetinaNet [Lin2017], as a popular dense detector network, is one of the best single-stage object detection model that has been proven to work efficiently with dense and small scale objects. The main new features of RetinaNet were the application of Feature Pyramid Networks and Focal Loss. By these enhancements it could reach the performance of previous single-stage detectors (e.g. Yolo, SSD) while it exceeds the accuracy of the existing state-of-the-art two-stage detectors

Figure 2: Typical results of the detection on two consecutive frames.

(R-CNN variants). Beside the Feature Pyramid Network it has two subnetworks: one for regression of the precise allocation of the bounding boxes, and the other one for classification (labeling). We used VGG19 as the backbone of RetinaNet.
Testing on the Hun169 we could reach about 0.805mAP (considering only objects of size larger than 10 pixels in any dimension). Most of the errors came from small objects and detections of such traffic signs which were missing from the trained classes (appearing as FP - false positives).
On Fig. 3 our large sized confusion (158×158) matrix (CM) with a magnified part can be seen to allow visualization of the performance of the detector. Each column of the matrix represents the instances in an actual class (for identifying the traffic road signs) while each row represents the instances in a predicted class by detectors. Black colors denote values zero or near to zero, while lighter colors in the main diagonal of matrix denote a large number of instances. The values are the averaged values of the aggregated matrix which was generated from 18 confusion matrices and derived from testing of the 18 videos.

3.3 Data-set for Tracking

For multi-object tracking purposes we used 20 videos from Hun169; Table 1 summarizes information about the training and testing parts (objects larger than 10 pixels were filtered out). During training of the tracking algorithm, statistical information was gathered about the size and position of the objects in the Ground Truth

(GT annotation) data and the probabilities of possible confusions of the detector ran antecedently.

Videos (@15 or 30 FPS)	training	testing
Number of videos	18	2
Names of videos HUN_	V01 - V18	V19, V20
Length of video (min:sec)	22:59	6:30
Number of frames	20703	11741
Number of objects	728	361
Number of bounding boxes	31455	20416

Table 1: Training and testing data-set for tracking.

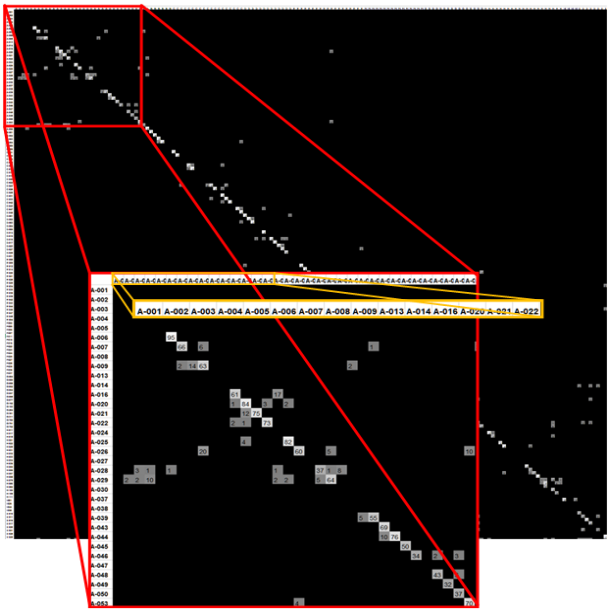

Figure 3: Illustration of the confusion matrix of the applied object detector at confidence threshold level 0.5 based on 18 videos of the Hun158 data-set.

4 TRACKING OF BOUNDING BOXES

We employ the Hungarian method to find the optimal correspondence between detected objects of two frames using the following cost function. Let us define the cost of pairing detections i and j from frame k and $k+1$:

$$\begin{aligned} C_{i,j} = \alpha(1 - CM(\mathscr{L}(d_{k,i}), \mathscr{L}(d_{k+1,j}))) \\ + \beta \frac{\phi_{\Delta A}(\Delta A)}{\max(\phi_{\Delta A})} + \gamma \frac{\phi_{\Delta y}(\Delta y)}{\max(\phi_{\Delta y})} \\ + \delta\,(1 - p_i p_j)\,, \end{aligned} \tag{1}$$

where $\mathscr{L}(d_{k,i})$ is the class label of the i^{th} detection on the k^{th} frame, CM is the confusion matrix (for illustration see Fig. 3), ΔA and Δy are the differences between bounding boxes' area and y coordinates in consecutive frames, $\phi_{\Delta A}$ and $\phi_{\Delta y}$ are the corresponding probability density function assuming normal distributions, and p_i is the confidence value of the i^{th} detection. All terms are between 0 and 1, grid-search is used to find their optimal settings, resulting in the best MOTA (Multiple Object Tracking Accuracy) [Stiefelhagen2006] value. The algorithm used for multiple object tracking is given in Algorithm 1.

Algorithm 1: Multiple Bounding Box Tracking algorithm

Input: $D = \{D_0, D_1, \cdots, D_{F-1}\}$
Initialize: $AT = \emptyset, FT = \emptyset$
for $f = 0$ *to* $F-1$ **do**
  while $at \in AT$ **do**
    $t = at_{last}$
    $d = Hun\_Method(AT_{last}, D_f, t), d \in D_f$
    if $d < Th$ **then**
      push d to at
      remove d from D_f
    else
      add at to FT
      remove at from AT
  while $d \in D_f$ **do**
    start new track list with d and insert into AT
while $at \in AT$ **do**
  add at to FT
Result: FT

The proposed algorithm requires a training phase to determine its main parameters, however, these parameters are easy to obtain (and are based on a larger set of image sequences as given in the next section). Since no motion flow is computed between the consecutive frames, we can rely purely on information given by the detector (coordinates of bounding boxes, labels, confusion matrix, and the detection's confidence values). The proposed algorithm has the following parameters:

- F: the number of frames in the sequence,
- D_f: list of detections on the frame f, $0 \leq f < F$,
- D: list of D_i, $0 \leq i < F$,
- AT: active tracks; $at \in AT$,
- at_{last}: last element of AT,
- FT: finished tracks,
- Th: threshold of the costs.

5 EXPERIMENTS

Demonstrating the usability of our algorithm, we used the benchmark data-set described in Section 3.3. Normalized histograms of the area changes and y-direction displacements of the 37163 bounding boxes of the 18 training videos (see Table 1) were computed to fit the normal distributions, as shown in Fig. 4, and in Table 2.

To measure the accuracy we use MOTA [Stiefelhagen2006], which is a widely used met-

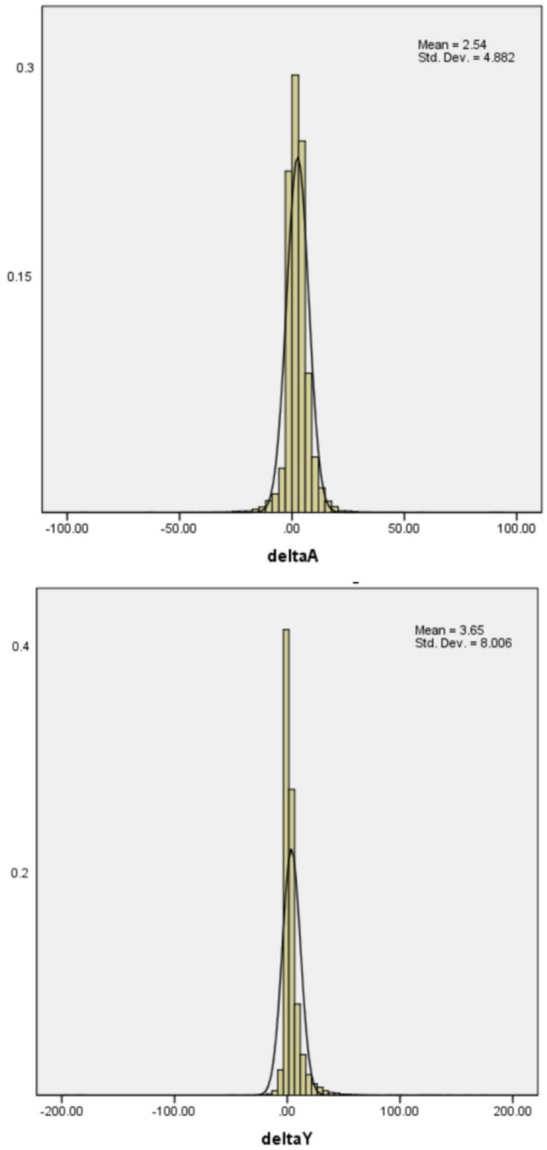

Figure 4: Normal distributions and their parameters fitted to the ΔA's and Δy's normalized histograms, measured on 18 training videos.

ric to evaluate the performance of multiple object tracking algorithms. It combines three sources of errors as FP (false positive), FN (false negative), and $IDSW$ (identity switch):

$$MOTA = 1 - \frac{\sum_k (FN_k + FP_k + IDSW_k)}{\sum_k GT_k}, \qquad (2)$$

where GT_k denotes the number of Ground Truth objects at frame k. The grid-based parameter search range is from 0 to 1 for all four parameters, using a step size of 0.2. The average MOTA value of the test videos was maximal at parameter values $\alpha = 1, \beta = 0.6, \gamma = 0.6$, and $\delta = 0.2$. The evaluation of tracking on videos shows that changes in parameters influence tracking outcomes, but the standard deviation is relatively small

	$\phi_{\Delta A}$	$\phi_{\Delta y}$
mean	2.5363	3.6472
standard deviation	4.88180	8.00638
max	0.2953	0.4168

Table 2: Parameters estimated on the 18 training sequences.

as seen in Table 3 above. The standard deviation is calculated at the parameters for the best MOTA values per video over the entire grid. Analyzing the diagrams in Fig. 5, we note that δ should be set to a relatively low value to reach high MOTA, thus the confidence value of detection should not be given much weight. The method is less sensitive to changes in α, β and γ.

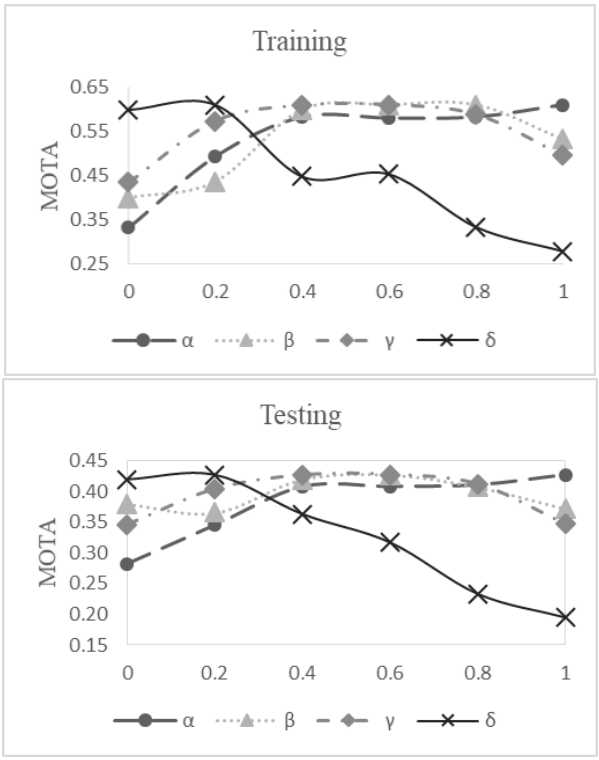

Figure 5: Parameter tests for MOTA with three fixed values. ($\alpha = 1, \beta = 0.6\,\gamma = 0.6, \delta = 0.2$)

6 CONCLUSIONS AND FUTURE WORK

In our article we described a tracking-by-detection method based on the outputs of an arbitrary object detector. Instead of using explicit textural features we rely on the confusion matrix of the detector to involve the possible similarity in appearance (which is very common for traffic signs). Motion is represented by the distribution of the vertical motion component of the detected bounding boxes accumulated in a training phase, thus no optical flow is needed, and there can be large changes in coordinates. This fits well to situations, where there is non-constant, often large

Video	Recall	Precision	F-measure	mAP	MOTA	Std. dev. (MOTA)
HUN_V01	0.8339	0.6658	0.7404	0.854	0.6728	0.0921
HUN_V02	0.6950	0.8487	0.7642	0.714	0.6595	0.1345
HUN_V03	0.7120	0.6758	0.6935	0.705	0.6754	0.0905
HUN_V04	0.8683	0.7694	0.8158	0.885	0.6133	0.1421
HUN_V05	0.7460	0.6396	0.6887	0.772	0.5956	0.0920
HUN_V06	0.9075	0.7762	0.8367	0.906	0.7223	0.0555
HUN_V07	0.7465	0.8419	0.7913	0.833	0.5440	0.0571
HUN_V08	0.7615	0.8022	0.7813	0.812	0.5922	0.1261
HUN_V09	0.8403	0.7368	0.7852	0.817	0.7248	0.1103
HUN_V10	0.9845	0.7737	0.8665	0.992	0.6155	0.0975
HUN_V11	0.4876	0.7269	0.5837	0.547	0.4201	0.0494
HUN_V12	0.7500	0.6970	0.7225	0.792	0.6624	0.1082
HUN_V13	0.5902	0.7451	0.6586	0.704	0.5561	0.0979
HUN_V14	0.7456	0.8200	0.7810	0.826	0.5481	0.1234
HUN_V15	0.7594	0.7988	0.7786	0.818	0.5465	0.0823
HUN_V16	0.6550	0.7042	0.6787	0.690	0.4671	0.1587
HUN_V17	0.9502	0.7744	0.8534	0.952	0.7778	0.1041
HUN_V18	0.8441	0.8169	0.8303	0.885	0.6032	0.1928
HUN_V19	0.4483	0.6274	0.5229	0.651	0.4143	0.1472
HUN_V20	0.4553	0.7078	0.5541	0.617	0.4404	0.0616

Table 3: The results of detection (at threshold 0.5) and tracking for 18 training and 2 test videos ($\alpha = 1$, $\beta = 0.6$, $\gamma = 0.6$, $\delta = 0.2$).

velocity and the camera has strong ego-motion (e.g. on-board car cameras).

We tested the proposed approach on a data-set of large number of object classes, sometimes with strong similarity. RetinaNet is a good candidate detector for such tasks. The complexity of tracking is very low, since the proposed cost function contains simple functions of pre-computed variables.

As future work we plan to compare it to other methods for speed and accuracy.

7 ACKNOWLEDGMENTS

Our special thanks to Dániel Zajzon for the object detection algorithms and to Nándor Szollát for data processing. We are grateful to the NVIDIA corporation for supporting our research with GPUs obtained by the NVIDIA GPU Grant Program. We acknowledge the financial support of the project EFOP-3.6.1-16-2016-00015 under the Széchenyi 2020 program, 2020-4.1.1-TKP2020 project under the Thematic Excellence Program, and grant GINOP-2.2.1-15-2017-00058.

8 REFERENCES

[Lim2017] Lim, K., Hong, Y., Choi, Y., Byun, H. (2017). Real-time traffic sign recognition based on a general purpose GPU and deep-learning. PLoS one, 12(3), e0173317.

[Temel2019] Temel, D., Alshawi, T., Chen, M. H., AlRegib, G. (2019). Challenging environments for traffic sign detection: Reliability assessment under inclement conditions. arXiv preprint arXiv:1902.06857.

[Lin2017] Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[VIA] VGG Image Annotator (VIA): https://www.robots.ox.ac.uk/ vgg/software/via/

[DarkLabel] Dark Label Image Annotation Tool: https://darkpgmr.tistory.com/16

[RouteShoot] RouteShoot georeferenced video recording application: https://www.wilsonpymmay.co.uk/

[Czúni2012] Czúni, L., Gál, M. (2012, September). Directional votes of optical flow projections for independent motion detection. In International Conference on Computer Vision and Graphics (pp. 329-336). Springer, Berlin, Heidelberg.

[Farneback2003] Farneback, Gunnar. "Two-frame motion estimation based on polynomial expansion." Scandinavian conference on Image analysis. Springer, Berlin, Heidelberg, (2003).

[Fiaz2019] Fiaz, M., Mahmood, A., Javed, S., Jung, S. K. (2019). Handcrafted and deep trackers: Recent visual object tracking approaches and trends.

ACM Computing Surveys (CSUR), 52(2), 1-44.

[Milan2016] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.

[Cannons2008] Cannons, K. (2008). A review of visual tracking. Dept. Comput. Sci. Eng., York Univ., Toronto, Canada, Tech. Rep. CSE-2008-07, 242.

[Smeulders2013] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M. (2013). Visual tracking: An experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(7), 1442-1468.

[Ning2017] Ning, G., Zhang, Z., Huang, C., Ren, X., Wang, H., Cai, C., He, Z. (2017, May). Spatially supervised recurrent convolutional neural networks for visual object tracking. In 2017 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-4). IEEE.

[Wang2019] Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P. H. (2019). Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1328-1338).

[Huang2008] Huang, C., Wu, B., Nevatia, R. (2008, October). Robust object tracking by hierarchical association of detection responses. In European Conference on Computer Vision (pp. 788-801). Springer, Berlin, Heidelberg.

[Bewley2016] Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B. (2016, September). Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464-3468). IEEE.

[Karunasekera2019] Karunasekera, H., Wang, H., Zhang, H. (2019). Multiple object tracking with attention to appearance, structure, motion and size. IEEE Access, 7, 104423-104434.

[Henriques2011] Henriques, J. F., Caseiro, R., Batista, J. (2011, November). Globally optimal solution to multi-object tracking with merged measurements. In 2011 International Conference on Computer Vision (pp. 2470-2477). IEEE.

[Weng2020] Weng, X., Wang, J., Held, D., Kitani, K. (2020). 3d multi-object tracking: A baseline and new evaluation metrics. arXiv preprint arXiv:1907.03961.

[Stiefelhagen2006] R. Stiefelhagen, K. Bernardin, R. Bowers, J. S. Garofolo, D. Mostefa, and P. Soundararajan. The CLEAR 2006 evaluation. In CLEAR, 2006.

[Kuhn1955] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2), 83-97.

[Ma2019] Ma, Y. (2019). An object tracking algorithm based on optical flow and temporal-spatial context. Cluster Computing, 22(3), 5739-5747.

[Jiang2018] Jiang, M. X., Deng, C., Pan, Z. G., Wang, L. F., Sun, X. (2018). Multiobject tracking in videos based on LSTM and deep reinforcement learning. Complexity, 2018.

[WangZ2019] Wang, Z., Zheng, L., Liu, Y., Wang, S. (2019). Towards real-time multi-object tracking. arXiv preprint arXiv:1909.12605, 2(3), 4.