Object Permanence in Object Detection Leveraging Temporal Priors at Inference Time

Michael Fürst*†, Priyash Bhugra†, René Schuster* and Didier Stricker*†
*DFKI - German Research Center for Artificial Intelligence
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
firsname.lastname@dfki.de
†Technische Universität Kaiserslautern
67663 Kaiserslautern, Germany

***Abstract*—Object permanence is the concept that objects do not suddenly disappear in the physical world. Humans understand this concept at young ages and know that another person is still there, even though it is temporarily occluded. Neural networks currently often struggle with this challenge. Thus, we introduce explicit object permanence into two stage detection approaches drawing inspiration from particle filters. At the core, our detector uses the predictions of previous frames as additional proposals for the current one at inference time. Experiments confirm the feedback loop improving detection performance by a up to 10.3 mAP with little computational overhead.**

Our approach is suited to extend two-stage detectors for stabilized and reliable detections even under heavy occlusion. Additionally, the ability to apply our method without retraining an existing model promises wide application in real-world tasks.

I. Introduction

Object detection has made significant advancements in the last decade and is applied to numerous new domains and live systems with increasing risks for humans and the environment. However, operating on single frames detectors lack understanding of object permanence. Object permanence [24] is the concept that objects in a physical world continue to exist despite the observers inability to sense them. This can result in temporally unstable detections and poor occlusion robustness.

These approaches without an understanding of object permanence are applied in domains where safety is critical such as robotics and autonomous vehicles. However, especially in these domains object permanence can greatly increase safety. For example, when a pedestrian disappears behind a pole, in the physical world the object is still there, but a common perception algorithm without object permanence does not detect the person anymore. Based on the most recent detections, the vehicle might optimize its trajectory to collide with the occluded pedestrians trajectory. Without prior knowledge of preceding time steps the model is unable to recover a detection precisely if the features computed from a single frame are ambiguous.

Object permanence is the understanding that an object still exists despite the inability to sense the object directly. This knowledge can help with disambiguation of features in the case of occlusions. In Figure 1 a person behind the pole is imperceptible to a Faster-RCNN [28]. When applying our approach giving the Faster-RCNN prior knowledge of the existence of the object, it is able to detect the person again without changing the weights of the model.

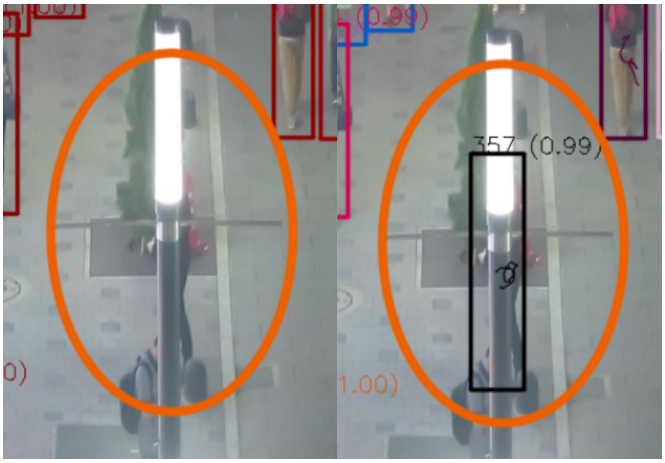

Fig. 1. A single frame object detector (left) cannot detect the pedestrian behind the street light, our approach leveraging object permanence (right) detects the person despite the major occlusion lasting 30 frames avoiding the overhead of tracking approaches.

In tracking, object permanence is used in integrated trackers to generate tracklets where an object has been. However, these trackers are computationally more expensive than detectors and require an altered training schema and special sequential data, limiting their availability to much rarer tracking data sets. Non integrated approaches like Kalman Filter [14] can be applied on detectors without special training requirements. However, temporal information is used only after the network. Thus the network lacks temporal information, leading to missing detections and lower precision.

We present an approach to integrate object permanence in two-stage object detectors using dynamic proposal priors. In contrast to full tracking approaches our goal is to improve detection performance with as little overhead and modification to the original task as possible. Thus, our approach does not produce full tracklets, but has little computational overhead. Further distinguishing our approach from current tracking approaches is that we integrate it into the model at test time, without re-training an existing two-stage single frame detector.

II. Related work

Object Detection is the task of predicting a bounding box for an object in an image or scene. The most common setup is 2D object detection, where the 2D axis aligned bounding box with a position and a size has to be predicted from an image. To achieve this goal various solutions have been proposed which can be generally categorized into single stage approaches and two stage approaches.

Based on the work of LeCun et al. [19], [20], OverFeat [30] was one of the first single stage approaches for object detection, followed by SSD [23], [7], YOLO [25], [26], [27], RetinaNet [22] and others like the CenterNet [40]. These approaches consist of a single CNN as a feature extractor and then directly predict the positions and classes of the objects.

In contrast to these approaches, two-stage detectors are descendants of traditional object detection, where algorithms find regions of interest (ROIs) in the image in a first step and then classify these in a subsequent step into foreground and background. Region-CNN [10] used a classical proposal algorithm and after cropping the ROIs used a CNN to extract the features and a support vector machine (SVM) for the classification to great success. However, as the inference speed was very slow, Fast-RCNN [9] and Faster-RCNN [28] improved this by computing proposals using a CNN and sharing the feature encoder between the proposal stage named Region Proposal Network (RPN) and the classification and refinement stage often referred to as the second stage (Figure 2). This greatly improved inference speed and accuracy of the models. Later adaptations of this methodology include Mask-RCNN [11] and others [12], [21]. The two stage-approaches can be further found in 3D object detectors [17] and 3D pose estimation [8] where fusion of multiple sensor streams is done.

Tracking-by-detection are algorithms which build on above object detectors and extend them by a tracking module. The core idea of these approaches is to use the detections, i.e. the bounding boxes, and track objects based on them. One of the simplest solutions is to use a Kalman Filter [14], [33] or a Particle Filter [3] on the detections.

A particle filter is a three step process, using a large number of particles representing a multimodal distribution. The particles are scored in a measurement step and then resampled based on the scores to better represent the distribution. Finally, it predicts the motion of particles over time. In object tracking, a particle usually is a tracklet with a history of past positions, a current position and a size for the bounding box.

Over time more complex approaches like [2], [6], [18], [29], [35], [41] have evolved. A key difficulty remains the association of predicted boxes to the boxes in the tracking filter. Thus, methods using appearance features [34], re-identification [32] and 3D shape information [31] have been applied.

Known limitations of tracking-by-detection are discarding image information in the data association step or using expensive feature extractors as pointed out by CenterTrack [39]. Beyond that, the tracking and detection are separated and prior knowledge of previous frames in the tracker cannot be used for better detection.

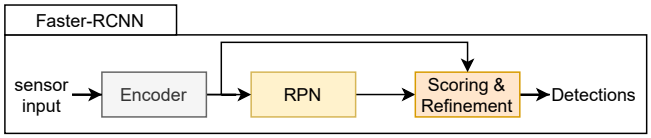

Fig. 2. Faster-RCNN [28] consists of three main building blocks. First is a shared encoder which produces feature maps used by both stages. Next is the Region Proposal Network (RPN) generating the proposals of the first stage in Faster-RCNN. Last is the scoring and refinement of the proposals based on the features extracted from the feature map.

Integrated Detection and Tracking: Due to the above limitations of tracking-by-detection effort has been spend on integrated detection and tracking. Integrated Detection [38] conditions the RPN and second stage on the tracklets. Kang et al. [15], [16] and Zhu et al. [42] use detection for a whole video segment and flow-warped intermediate features with a Faster-RCNN. TransCenter [36] uses transformers for tracking, while FairMOT [37] focuses on providing fair features for detection and re-identification to boost the performance of integrated tracking.

Tracktor [1] uses the second stage of Faster-RCNN to re-align the boxes for the next time step and uses proposals form the RPN which have no substantial IoU with the existing tracks to generate new tracks. This approach is most similar to ours, but using only proposals with a low IoU with existing tracks, they limit the potential for exploration and multi hypothesis modeling in case of pedestrian to pedestrian occlusions, which cause high IoUs while still representing different objects.

The box and IoU centric nature of Faster-RCNN based approaches is, according to CenterTrack [39], the cause for association difficulties. Thus, they follow the idea of CenterNet [40] to predict the center of the bounding box as a heatmap and extends this by using the heatmap of the previous frame as an input to the current frame. Further they predict the offset of the center from the current to the last frame to solve the association problem. We agree, that IoU based suppression of detections is an issue for multi-modal bounding box distributions, specifically in the case of occlusions, but we emphasize that Faster-RCNN does not rely on the IoU based suppression to generate new proposals.

Other methods than IoU based suppression and selection have been successfully applied. A particle filter has various particles which not directly suppress each other, but by scoring and re-sampling a selection process is taking place. This makes particle filters optimal filters for complex multi-modal distributions, given sufficient particles.

In contrast to other approaches like Tracktor which used manual proposal selection via IoU thresholding, we propose an approach inspired by the implicit scoring of particle filters and integrate it into a pre-trained Faster-RCNN at inference time. To our knowledge there have been no experiments on full integration of particle filter concepts and a Faster-RCNN.

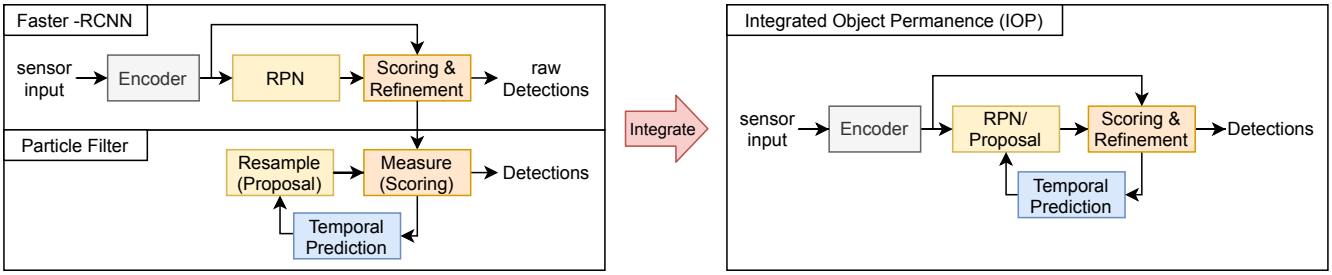

Fig. 3. The core of our approach is integrating the particle filter into the Faster-RCNN as their modules share similar functionality. We combine the RPN and proposal (re-sample) step from the Faster-RCNN into one and integrate the scoring of both into a single step, resulting in the approach on the right, a two-stage object detector with integrated object permanence. As the integration does not change the weights of the model, there is no re-training of an existing detector required. Further, leveraging the synergies reduces the computational complexity and is favorable in terms of accuracy as shown by our experiments.

III. Approach

In this work we present an approach for object permanence in detection which is heavily inspired by particle filters. Particle filters propagate box candidates between frames, re-sample and score them. By propagating the box candidates between frames the approach has an explicit model for object permanence.

As our approach requires some understanding of a traditional baseline using Faster-RCNN and a separate particle filter, we will present that first and then describe how to integrate the particle filter into the model to derive our Faster-RCNN detector with object permanence. All changes to the model will not require any re-training and have minimal computational overhead, when the particle filter is fully integrated.

A. Baseline: Extending Faster-RCNN with Particle Filter

First we establish a baseline by extending Faster-RCNN with particle filter as a pre-cursor to our approach. We choose to use a particle filter, as it shares the conceptual foundation for our approach. Sharing concepts it is a straightforward stepping stone and allows to clearly attribute all observed performance changes to our contributions.

Faster-RCNN: In our implementation we use a regular Faster-RCNN [28] re-implementation without any bells and whistles, since it is widely used and a can be considered the gold-standard of two-stage detection. For example, the MOT Dataset [4] uses Faster-RCNN to provide baseline predictions.

RCNNs are a two stage approach meaning it has a region proposal and a refinement stage. The region proposal stage uses the entire image to find regions of interest (ROIs) where objects are likely. The refinement stage then uses ROI crops of the image to refine the proposals by scoring them for the object that is visible and predicting deltas between the proposal and the actual box. In Faster-RCNN [28] the first and second stage share an encoder which predicts a feature map and apply the ROI pooling on the feature map greatly reducing the inference time. Figure 2 visualizes the architecture.

Particle Filter: A *Particle Filter* predicts the behavior of objects over time using particles. A particle is a detection with velocity as an additional attribute, that is estimated by the filter. Our implementation is build from three core components: Resample, measure and predict.

The prediction step in our implementation uses a simple constant velocity linear motion model to predict the position of the particle in the next frame. We assume a fixed frame rate by using a constant Δt.

In the measurement step, errors made by the prediction are partially corrected by assigning the particles to detections from the detector and in the case of a successful assignment interpolating linearly. For this assignment and correction an IoU with the predictions is computed and used to update the scores of particles for subsequent re-sampling step.

The resampling step creates a distribution of particles around interesting regions, compensating uncertainties of the system, e.g. motion changes. To re-sample, we simply use existing particles, remove the particles with the lowest scores and re-sample new particles with a bias towards detections from the Faster-RCNN which have a low IoU with all particles. This bias helps to reduce the number of required particles for the filter and improves FPS.

By using a baseline that is consistent with the benchmark defaults and adding a particle filter we can focus on the *integration* of the particle filter into the model and attribute all improvements to the integration. An overview over our baseline architecture is given in Figure 3 on the left.

B. Two-Stage Detection with Integrated Object Permanence (IOP)

To integrate object permanence into two stage detection, we merge the particle filter and the Faster-RCNN. Their components have similar functionalities enabling this integration. As described in our baseline, a particle filter consists of three steps: Resample, predict and measure. The goal of the resample step can be described as generating proposals of ROIs and is similar to the RPN in Faster-RCNN. The measurement step then scores these proposals by their plausibility based on the measurements and can be compared to the scoring and refinement stage in Faster-RCNN. Finally, the prediction step is used to relate two timesteps and propagate the particles through time, there is no equivalent in two-stage object detectors.

Leveraging the similarities, we merge resampling (proposal generation) and the RPN, as well as the measurement step of the particle filter with the refinement of Faster-RCNN. The prediction step from particle filter is then used to connect the refinement with the proposal stage. We call the resulting architecture *Integrated Object Permanence (IOP)* and visualize it in Figure 3 on the right side. However, design decisions when implementing lead to *IOP with particles*, *IOP lite* and *IOP with history*, with varying precision and inference speed.

IOP with particles the most similar to the particle filter baseline. For resampling, we apply the traditional resample of the particle filter and then use the particle detections as additional proposals for the second stage by concatenating. Assigning the outputs of the second stage to the particles again is done via IoU based assignment. This extra step is required, since we use a standard detector without any knowledge of tracklets. After assignment, the particle filter can be used to predict for the next time frame. Figure 4 including the dashed boxes visualizes this architecture.

IOP lite applies the least changes to an existing two-stage detector, completely omitting tracklets. The resampling step concatenates the unchanged predictions from the previous frame to the proposals of Faster-RCNN from the current frame, as the RPN is sampling new meaningful proposals. The measurement step is just the second stage of Faster-RCNN and the prediction step is a simple time-delay. Figure 4 without the dashed boxes visualizes this architecture.

IOP with history is identical to the above except for the resampling step. Here, predictions from the N previous frames are concatenated to the proposals from the RPN. One advantage is if an object is fully occluded for a frame and rejected by the second stage, the history allows for quicker recovery once the object is partially visible again.

We trained Faster-RCNN once and all presented variations use the same weights. There is no training of the individual variations required, as the integration does not change the model itself. This allows our approach to be integrated into any pre-trained two-stage approach. Independent of the training data this improves models for inference on sequential data, despite a training on sequential data.

In summary, the core idea of our approach is the feedback of previous predictions as proposals into the model. Inspired by particle filter, it allows varying degrees of complexity. Further, keeping weights of the model intact, this idea can be applied to a wide variety of approaches and use cases.

IV. Experiments

Our approach is an improvement for object detection. However, since object permanence is a temporal aspect, we need a tracking dataset for our evaluation. The evaluation on the tracking dataset will evaluate mainly detection performance gains and computational overhead during inference time. However, a brief comparison to other trackers on applicable metrics will be done.

For evaluation of our approach for integrated object permanence the challenging MOT17 and MOT20 [4] dataset is ideal. One of the baselines on the dataset is Faster-RCNN which is also part of our baselines: Faster-RCNN with Particle Filter and Faster-RCNN with Kalman Filter. Comparing against these baselines we will show the effectiveness of the components in our approach.

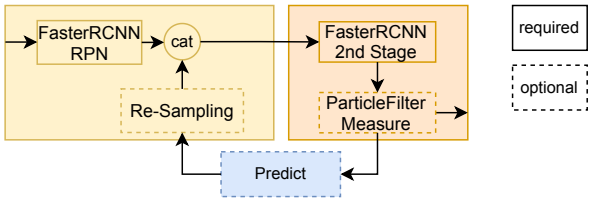

Fig. 4. At the core of our implementation for Integrated Object Permanence (IOP) is the depicted information flow. The concatenation of previous predictions and current proposals is a critical component for object permanence in two stage-detection. Other components denoted by dashed boxes are optional and can be removed for faster inference speed.

A. Experimental Setup

The MOT17 and MOT20 [4] dataset consists of 1.423 Frames with 80.274 annotated pedestrians in 11 sequences in total for validation. The dataset contains heavy occlusions induced by the environment and other pedestrians. Thus the dataset is very challenging and has multiple situations where object permanence is key for correct detection.

We train Faster-RCNN on this dataset from scratch without any additional datasets or augmentation apart from the pre-trained VGG16 encoder. SGD optimizer with momentum (0.9) and a learning rate of 0.001 is used to train the model for 30 epochs. As an encoder, the model uses VGG16 with batch norm pre-trained on ImageNet [5]. For training we follow the standard procedures of Faster-RCNN with sampling 128 samples from the ROIs for training attempting balanced sampling and filling with negative samples.

Kalman Filter still is the gold standard in application. Thus, we also evaluate a simple Kalman Filter implemented in C++. This baseline uses above trained Faster-RCNN and no special configuration.

For the particle filter baseline, the same pre-trained Faster-RCNN is used. The particle filter baseline is optimized for sample efficiency and 200 particles produce sufficiently good results at acceptable inference speed. Thus we use 50, 75, 100 and 200 particles respectively for our evaluations.

Our approaches, described in Section III-B, use again the same Faster-RCNN. The configuration named *IOP* uses the same number of particles as the particle filter for best comparability. *IOP lite* has no configuration options and for *IOP with history* we evaluated different history lengths from 1-19 frames, but limit reporting to the best history length of 5 and 9 for IOP and IOP lite respectively in the tracking section.

B. Detection Performance

As an improvement for detectors, measuring gains in the detection performance is the most important evaluation. As

TABLE I

COMPARISON OF MAP GAINS ON DIFFERENT DATA SEQUENCES. OUR INTEGRATED OBJECT PERMANENCE (IOP) WITH PARTICLES IS BEST OVERALL, WHEREAS OUR IOP LITE NEVER DEGRADES PERFORMANCE AT SECOND BEST OVERALL MAP GAINS.

Sequence	FRCNN	FRCNN + KF	Baseline (FRCNN+ PF)				IOP with Particles [ours]				IOP Lite [ours]
			50	75	100	200	50	75	100	200	
MOT17-02	39.6	+5.6	+7.3	+7.3	+7.3	+7.3	**+11.6**	**+11.6**	**+11.6**	**+11.6**	+5.9
MOT17-04	78.5	+4.0	+3.5	+4.4	+4.4	+4.4	+6.1	**+6.2**	**+6.2**	**+6.2**	+2.5
MOT17-05	73.3	-13.3	**+0.3**	**+0.3**	**+0.3**	**+0.3**	-3.1	-3.1	-3.1	-3.1	+0.0
MOT17-09	73.2	+2.5	+7.4	+7.4	+7.4	+7.4	**+8.1**	**+8.1**	**+8.1**	**+8.1**	+4.5
MOT17-10	26.1	+3.6	+3.2	+3.2	+3.2	+3.2	**+8.8**	**+8.8**	**+8.8**	**+8.8**	+7.4
MOT17-11	51.7	-0.4	**+1.5**	**+1.5**	**+1.5**	**+1.5**	+0.0	+0.0	+0.0	+0.0	+0.0
MOT17-13	12.7	+2.5	-2.4	-2.4	-2.4	-2.4	+7.3	+7.3	+7.3	+7.3	**+7.9**
MOT20-01	66.6	+7.4	+12.6	+12.6	+12.6	+12.6	**+16.1**	**+16.1**	**+16.1**	**+16.1**	+12.1
MOT20-02	72.3	+5.3	+6.9	+8.3	+8.3	+8.3	**+11.0**	**+11.0**	**+11.0**	**+11.0**	+5.5
MOT20-03	41.9	+19.9	-10.7	+3.2	+12.6	+20.2	-8.1	+5.6	+18.4	**+28.3**	+22.2
MOT20-05	55.4	+14.3	-25.2	-11.2	-0.4	+14.5	-24.3	-7.7	-0.7	**+18.7**	+12.6
MINIMUM	12.7	-13.3	-25.2	-11.2	-2.4	-2.4	-24.3	-7.7	-3.1	-3.1	**+0.0**
MAXIMUM	78.5	+19.9	+12.6	+12.6	+12.6	+20.2	+16.1	+16.1	+18.4	**+28.3**	+22.2
AVERAGE	53.8	+4.7	+0.4	+3.2	+5.0	+7.0	+3.0	+5.8	+7.6	**+10.3**	+7.3

TABLE II

AVERAGE LATENCY OVERHEAD OF THE TESTED APPROACHES ON MOT.

Approach	#Particles	mAP	Latency
Faster-RCNN	-	53.8	550 ms
Kalman Filter	-	+4.7	**+3 ms**
Particle Filter	50	+0.4	+76 ms
	75	+3.2	+57 ms
	100	+5.0	+60 ms
	200	+7.0	+63 ms
IOP with Particles [ours]	50	+3.0	+91 ms
	75	+5.8	+74 ms
	100	+7.6	+79 ms
	200	**+10.3**	+79 ms
IOP Lite [ours]	-	+7.3	**+3 ms**

common in object detection we use the Pascal VOC [13] mean average precision (mAP) metric for evaluation. We evaluated on a per sequence basis and an average over all sequences (see Table I), allowing for further insights into the performance.

Averaged over all sequences, our integrated object permanence approach with 200 particles with a gain of +10.3 mAP is best. IOP lite which is faster and requires no particles achieves an average improvement of +7.32 mAP. Kalman Filter and Particle Filter can also increase the performance by +4.7 mAP and +7.0 mAP but are outperformed by our IOP or IOP lite.

On MOT20 gains are larger than on MOT17. It can be observed, that gains are larger on sequences with elevated camera position and lower ego-motion. In scenes with higher perceived camera motion the gains are smaller and sometimes performance degrades. This can be explained by the large motion objects have in the image plane and the lack of an ego-motion model to compensate this in all presented approaches.

Overall, our integrated object permanence approach with 200 particle is best in 8 out of 11 sequences. Our IOP lite is second best in the average category and never degrades the performance of Faster-RCNN, which no other approach was able to achieve. We explain this by the fact, that IOP lite injects additional proposals and does not remove any proposals or predictions from Faster-RCNN.

C. Latency Overhead

As latency is an important factor for predictors, we measured the overhead produced by each of the presented approaches compared to the original Faster-RCNN. We averaged the latency over all samples, to minimize effects caused by the underlying operating systems scheduler.

Our IOP lite (implemented in Python) and Kalman filter (implemented in C++), share the same overhead of 3 ms. Using particles in a particle filter and our IOP with particles, the overhead is 60-91 ms (see Table II). The number of particles has little impact on the overhead.

When speed is more important than the best possible mAP, it is recommended to use IOP lite over IOP with particles, as the overhead is significant.

D. Tracking Metrics

Evaluation of tracking metrics was done on the MOT17 validation dataset to evaluate our four best configurations against the other state-of-the-art approaches. Since the numbers for the other approaches were extracted from the papers, we can only compare against reported numbers. For CenterTrack [39] FP, FN and IDS are not comparable, since they are reported in percent.

In Table III it can be seen that our IOP outperforms Tracktor [1], CenterTrack [39] and FairMOT [37] in all captured metrics except MT. Our approach is designed to improve detection performance and not primarily as a tracker, however, the integration makes the outputs of the model easily usable by simple IoU based association.

E. Qualitative Analysis

When qualitatively analyzing the results, we immediately find situations in which the integration of object permanence can change the quality of the output of Faster-RCNN. In Figure 5 example outputs of Faster-RCNN and our approach are visualized. It can be seen, that on the left some pedestrians cannot be recovered by Faster-RCNN, but with the feedback loop the pedestrian can be successfully detected with high

TABLE III

EVALUATING TRACKING METRICS ON MOT 17 VALIDATION SET, THE OVERALL BEST APPROACH IS IOP LITE WITH A HISTORY LENGTH OF 9 FRAMES. *NUMBERS IN PERCENT ARE EXCLUDED FROM COMPARISON.

Approach	Hist	MOTA	MOTP	IDF1	MT	ML	FP	FN	IDS
Tracktor [1]	-	61.9	-	64.7	**35.3**	21.4	323	42454	326
CenterTrack [39]	-	66.1	-	64.2	41.3	21.2	4.5%*	28.4%*	1.0%*
FairMOT [37]	-	69.1	-	72.8	-	-	-	-	299
IOP w/ Particles [ours]	1	52.3	34.1	67.8	123	**12**	1591	**678**	231
IOP w/ Particles [ours]	5	52.5	31.4	69.1	114	**12**	1554	750	200
IOP lite [ours]	1	69.4	61.7	75.3	89	37	131	1817	**46**
IOP lite [ours]	9	**72.5**	**63.5**	**77.1**	93	34	**83**	1685	48

Fig. 5. Qualitative comparison of Faster-RCNN (left), Integrated Object Permanence (IOP) with particles (center) and IOP lite (right). The highlighted persons are hard to detect but can be recovered by using spatio-temporal information at inference time without sequential training data.

accuracy. In the lower image the person is behind the pole. During 28 frames Faster-RCNN detects the person only in 5 frames, with a maximum confidence of 0.5. In contrast, our proposed IOP detects the person in all frames with a minimum confidence of 0.91. IOP can stabilize and improve the predictions of a pre-trained Faster-RCNN, but it is dependent on the Faster-RCNN to detect the object in at least a few frames with low confidence.

V. CONCLUSIONS

Object detection is applied to safety-critical domains, e.g. robotics and autonomous vehicles. However, current detectors lack the concept of object permanence, leading to temporally unstable predictions, e.g. with temporary occlusion. Trackers solve this but are complex or need special training data.

Our Integrated Object Permanence (IOP) fills this gap by introducing object permanence into two-stage approaches without the need for re-training. Inspired by a particle filter, a feedback loop is integrated into a Faster-RCNN. At the core, predictions of previous frames are used as proposals for the next frame.

In multiple experimental setups, the effects of each design decision are evaluated and we conclude, that for most use-cases IOP lite is the best option, as it has best or second to best performance with least computational overhead. IOP with particles is best in object detection, but IOP lite is best in inference speed and tracking on most metrics.

Our approach is an ideal solution to improve object detection performance without any need for sequential training data. As no retraining is needed, we can apply this approach to already existing two-stage detectors, like Faster-RCNN. The concept is general and only requires a proposal and refinement step in the model and has little computational overhead as exhibited by IOP lite. Thus, we see a wide range of applications and use-cases where IOP can improve object detection. For example in instance segmentation or even human pose estimation.

ACKNOWLEDGMENT

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project ”KI-Absicherung” (grant: 19A19005U).

REFERENCES

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *CVPR*, 2019. 2, 5, 6

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 2

[3] Yvo Boers and JN Driessen. Particle filter based detection for tracking. In *IEEE American Control Conference.(Cat. No. 01CH37148)*, 2001. 2

[4] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. In *1906.04567*, 2018. 3, 4

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[6] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. 2018. 2

[7] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander Berg. DSSD: Deconvolutional Single Shot Detector. *CoRR*, abs/1701.06659, 2017. 2

[8] Michael Fürst, Shriya TP Gupta, René Schuster, Oliver Wasenmüller, and Didier Stricker. Hperl: 3d human pose estimation from rgb and lidar. In *ICPR*, 2021. 2

[9] Ross Girshick. Fast r-cnn. In *CVPR*, 2015. 2

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. 37(9):1904–1916, 2015. 2

[13] Derek Hoiem, Santosh K Divvala, and James H Hays. Pascal voc 2008 challenge. *World Literature Today*, 2009. 5

[14] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 1, 2

[15] Kai Kang, Hongsheng Li, and Tong Xiao et al. Object detection in videos with tubelet proposal networks. In *CVPR*, 2017. 2

[16] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE TCSVT*, 28(10), 2017. 2

[17] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. 2018. 2

[18] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016. 2

[19] Yann LeCun, Bernhard Boser, and John S Denker et al. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural computation*, 1(4):541–551, 1989. 2

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 2

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *CVPR*, 2017. 2

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*. Springer, 2016. 2

[24] Jean Piaget. *The construction of reality in the child*, volume 82. Routledge, 2013. 1

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[26] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 2

[27] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *1804.02767*, 2018. 2

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2, 3

[29] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. In *CVPR*, 2017. 2

[30] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *CoRR*, abs/1312.6229, 2013. 2

[31] Sarthak Sharma, Junaid Ahmed Ansari, J Krishna Murthy, and K Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. IEEE, 2018. 2

[32] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, 2017. 2

[33] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 2

[34] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 2

[35] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *ICCV*, 2019. 2

[36] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *2103.15145*, 2021. 2

[37] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *2004.01888*, 2020. 2, 5, 6

[38] Zheng Zhang, Dazhi Cheng, Xizhou Zhu, Stephen Lin, and Jifeng Dai. Integrated object detection and tracking with tracklet-conditioned detection. *1811.11167*, 2018. 2

[39] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*. Springer, 2020. 2, 5, 6

[40] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *1904.07850*, 2019. 2

[41] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *ECCV*, 2018. 2

[42] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *CVPR*, 2017. 2