SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC

Nicolai Wojke[†], *Alex Bewley*[◇], *Dietrich Paulus*[†]

University of Koblenz-Landau[†], Queensland University of Technology[◇]

ABSTRACT

Simple Online and Realtime Tracking (SORT) is a pragmatic approach to multiple object tracking with a focus on simple, effective algorithms. In this paper, we integrate appearance information to improve the performance of SORT. Due to this extension we are able to track objects through longer periods of occlusions, effectively reducing the number of identity switches. In spirit of the original framework we place much of the computational complexity into an offline pre-training stage where we learn a deep association metric on a large-scale person re-identification dataset. During online application, we establish measurement-to-track associations using nearest neighbor queries in visual appearance space. Experimental evaluation shows that our extensions reduce the number of identity switches by 45%, achieving overall competitive performance at high frame rates.

Index Terms— Computer Vision, Multiple Object Tracking, Data Association

1. INTRODUCTION

Due to recent progress in object detection, tracking-by-detection has become the leading paradigm in multiple object tracking. Within this paradigm, object trajectories are usually found in a global optimization problem that processes entire video batches at once. For example, flow network formulations [1, 2, 3] and probabilistic graphical models [4, 5, 6, 7] have become popular frameworks of this type. However, due to batch processing, these methods are not applicable in online scenarios where a target identity must be available at each time step. More traditional methods are Multiple Hypothesis Tracking (MHT) [8] and the Joint Probabilistic Data Association Filter (JPDAF) [9]. These methods perform data association on a frame-by-frame basis. In the JPDAF, a single state hypothesis is generated by weighting individual measurements by their association likelihoods. In MHT, all possible hypotheses are tracked, but pruning schemes must be applied for computational tractability. Both methods have recently been revisited in a tracking-by-detection scenario [10, 11] and shown promising results. However, the performance of these methods comes at increased computational and implementation complexity.

Simple online and realtime tracking (SORT) [12] is a much simpler framework that performs Kalman filtering in image space and frame-by-frame data association using the Hungarian method with an association metric that measures bounding box overlap. This simple approach achieves favorable performance at high frame rates. On the MOT challenge dataset [13], SORT with a state-of-the-art people detector [14] ranks on average higher than MHT on standard detections. This not only underlines the influence of object detector performance on overall tracking results, but is also an important insight from a practitioners point of view.

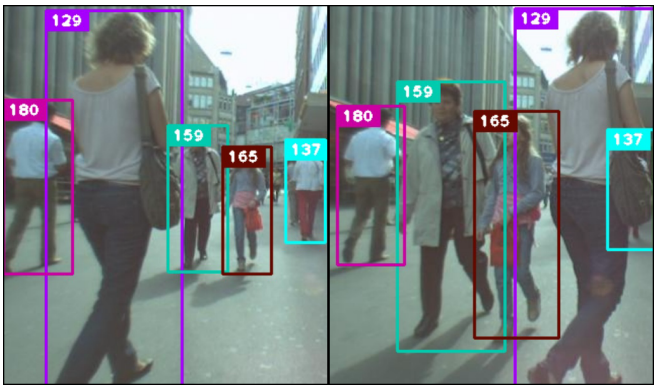

Fig. 1: Exemplary output of our method on the MOT challenge dataset [15] in a common tracking situation with frequent occlusion.

While achieving overall good performance in terms of tracking precision and accuracy, SORT returns a relatively high number of identity switches. This is, because the employed association metric is only accurate when state estimation uncertainty is low. Therefore, SORT has a deficiency in tracking through occlusions as they typically appear in frontal-view camera scenes. We overcome this issue by replacing the association metric with a more informed metric that combines motion and appearance information. In particular, we apply a convolutional neural network (CNN) that has been trained to discriminate pedestrians on a large-scale person re-identification dataset. Through integration of this network we increase robustness against misses and occlusions while keeping the system easy to implement, efficient, and applicable to online scenarios. Our code and a pre-trained CNN model are made publicly available to facilitate research experimentation and practical application development.

2. SORT WITH DEEP ASSOCIATION METRIC

We adopt a conventional single hypothesis tracking methodology with recursive Kalman filtering and frame-by-frame data association. In the following section we describe the core components of this system in greater detail.

2.1. Track Handling and State Estimation

The track handling and Kalman filtering framework is mostly identical to the original formulation in [12]. We assume a very general tracking scenario where the camera is uncalibrated and where we have no ego-motion information available. While these circumstances pose a challenge to the filtering framework, it is the most common setup considered in recent multiple object tracking benchmarks [15]. Therefore, our tracking scenario is defined on the eight dimensional state space $(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ that contains the bounding box center position (u, v), aspect ratio γ, height h, and their respective velocities in image coordinates. We use a standard Kalman filter with constant velocity motion and linear observation model, where we take the bounding coordinates (u, v, γ, h) as direct observations of the object state.

For each track k we count the number of frames since the last successful measurement association a_k. This counter is incremented during Kalman filter prediction and reset to 0 when the track has been associated with a measurement. Tracks that exceed a predefined maximum age $A_{\max}$ are considered to have left the scene and are deleted from the track set. New track hypotheses are initiated for each detection that cannot be associated to an existing track. These new tracks are classified as tentative during their first three frames. During this time, we expect a successful measurement association at each time step. Tracks that are not successfully associated to a measurement within their first three frames are deleted.

2.2. Assignment Problem

A conventional way to solve the association between the predicted Kalman states and newly arrived measurements is to build an assignment problem that can be solved using the Hungarian algorithm. Into this problem formulation we integrate motion and appearance information through combination of two appropriate metrics.

To incorporate motion information we use the (squared) Mahalanobis distance between predicted Kalman states and newly arrived measurements:

$$d^{(1)}(i,j) = (\boldsymbol{d}_j - \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{S}_i^{-1} (\boldsymbol{d}_j - \boldsymbol{y}_i), \tag{1}$$

where we denote the projection of the i-th track distribution into measurement space by $(\boldsymbol{y}_i, \boldsymbol{S}_i)$ and the j-th bounding box detection by $\boldsymbol{d}_j$. The Mahalanobis distance takes state estimation uncertainty into account by measuring how many standard deviations the detection is away from the mean track location. Further, using this metric it is possible to exclude unlikely associations by thresholding the Mahalanobis distance at a 95% confidence interval computed from the inverse χ^2 distribution. We denote this decision with an indicator

$$b_{i,j}^{(1)} = \mathbb{1}[d^{(1)}(i,j) \leq t^{(1)}] \tag{2}$$

that evaluates to 1 if the association between the i-th track and j-th detection is admissible. For our four dimensional measurement space the corresponding Mahalanobis threshold is $t^{(1)} = 9.4877$.

While the Mahalanobis distance is a suitable association metric when motion uncertainty is low, in our image-space problem formulation the predicted state distribution obtained from the Kalman filtering framework provides only a rough estimate of the object location. In particular, unaccounted camera motion can introduce rapid displacements in the image plane, making the Mahalanobis distance a rather uninformed metric for tracking through occlusions. Therefore, we integrate a second metric into the assignment problem. For each bounding box detection $\boldsymbol{d}_j$ we compute an appearance descriptor $\boldsymbol{r}_j$ with $\|\boldsymbol{r}_j\| = 1$. Further, we keep a gallery $\mathcal{R}_k = \{\boldsymbol{r}_k^{(i)}\}_{k=1}^{L_k}$ of the last $L_k = 100$ associated appearance descriptors for each track k. Then, our second metric measures the smallest cosine distance between the i-th track and j-th detection in appearance space:

$$d^{(2)}(i,j) = \min\{1 - \boldsymbol{r}_j^{\mathrm{T}} \boldsymbol{r}_k^{(i)} \mid \boldsymbol{r}_k^{(i)} \in \mathcal{R}_i\}. \tag{3}$$

Again, we introduce a binary variable to indicate if an association is admissible according to this metric

$$b_{i,j}^{(2)} = \mathbb{1}[d^{(2)}(i,j) \leq t^{(2)}] \tag{4}$$

and we find a suitable threshold for this indicator on a separate training dataset. In practice, we apply a pre-trained CNN to compute bounding box appearance descriptors. The architecture of this network is described in Section 2.4.

In combination, both metrics complement each other by serving different aspects of the assignment problem. On the one hand, the Mahalanobis distance provides information about possible object locations based on motion that are particularly useful for short-term predictions. On the other hand, the cosine distance considers appearance information that are particularly useful to recover identities after long-term occlusions, when motion is less discriminative. To build the association problem we combine both metrics using a weighted sum

$$c_{i,j} = \lambda\, d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{5}$$

where we call an association admissible if it is within the gating region of both metrics:

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)}. \tag{6}$$

Listing 1 Matching Cascade

Input: Track indices $\mathcal{T} = \{1, \dots, N\}$, Detection indices $\mathcal{D} = \{1, \dots, M\}$, Maximum age $A_{\max}$

1: Compute cost matrix $\boldsymbol{C} = [c_{i,j}]$ using Eq. 5
2: Compute gate matrix $\boldsymbol{B} = [b_{i,j}]$ using Eq. 6
3: Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
4: Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
5: **for** $n \in \{1, \dots, A_{\max}\}$ **do**
6: Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
7: $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(\boldsymbol{C}, \mathcal{T}_n, \mathcal{U})$
8: $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
9: $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
10: **end for**
11: **return** $\mathcal{M}, \mathcal{U}$

The influence of each metric on the combined association cost can be controlled through hyperparameter λ. During our experiments we found that setting $\lambda = 0$ is a reasonable choice when there is substantial camera motion. In this setting, only appearance information are used in the association cost term. However, the Mahalanobis gate is still used to disregarded infeasible assignments based on possible object locations inferred by the Kalman filter.

2.3. Matching Cascade

Instead of solving for measurement-to-track associations in a global assignment problem, we introduce a cascade that solves a series of subproblems. To motivate this approach, consider the following situation: When an object is occluded for a longer period of time, subsequent Kalman filter predictions increase the uncertainty associated with the object location. Consequently, probability mass spreads out in state space and the observation likelihood becomes less peaked. Intuitively, the association metric should account for this spread of probability mass by increasing the measurement-to-track distance. Counterintuitively, when two tracks compete for the same detection, the Mahalanobis distance favors larger uncertainty, because it effectively reduces the distance in standard deviations of any detection towards the projected track mean. This is an undesired behavior as it can lead to increased track fragmentations and unstable tracks. Therefore, we introduce a matching cascade that gives priority to more frequently seen objects to encode our notion of probability spread in the association likelihood.

Listing 1 outlines our matching algorithm. As input we provide the set of track $\mathcal{T}$ and detection $\mathcal{D}$ indices as well as the maximum age $A_{\max}$. In lines 1 and 2 we compute the association cost matrix and the matrix of admissible associations. We then iterate over track age n to solve a linear assignment problem for tracks of increasing age. In line 6 we select the subset of tracks $\mathcal{T}_n$ that have not been associated with a detection in the last n frames. In line 7 we solve the linear assignment between tracks in $\mathcal{T}_n$ and unmatched detections $\mathcal{U}$.

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

Table 1: Overview of the CNN architecture. The final batch and ℓ_2 normalization projects features onto the unit hypersphere.

In lines 8 and 9 we update the set of matches and unmatched detections, which we return after completion in line 11. Note that this matching cascade gives priority to tracks of smaller age, i.e., tracks that have been seen more recently.

In a final matching stage, we run intersection over union association as proposed in the original SORT algorithm [12] on the set of unconfirmed and unmatched tracks of age $n = 1$. This helps to to account for sudden appearance changes, e.g., due to partial occlusion with static scene geometry, and to increase robustness against erroneous initialization.

2.4. Deep Appearance Descriptor

By using simple nearest neighbor queries without additional metric learning, successful application of our method requires a well-discriminating feature embedding to be trained offline, before the actual online tracking application. To this end, we employ a CNN that has been trained on a large-scale person re-identification dataset [21] that contains over 1,100,000 images of 1,261 pedestrians, making it well suited for deep metric learning in a people tracking context.

The CNN architecture of our network is shown in Table 1. In summary, we employ a wide residual network [22] with two convolutional layers followed by six residual blocks. The global feauture map of dimensionality 128 is computed in dense layer 10. A final batch and ℓ_2 normalization projects features onto the unit hypersphere to be compatible with our cosine appearance metric. In total, the network has 2,800,864 parameters and one forward pass of 32 bounding boxes takes approximately 30 ms on an Nvidia GeForce GTX 1050 mobile GPU. Thus, this network is well suited for online tracking, provided that a modern GPU is available. While the details of our training procedure are out of the scope of this paper, we provide a pre-trained model in our GitHub reposi-

		MOTA ↑	MOTP ↑	MT ↑	ML ↓	ID ↓	FM ↓	FP ↓	FN ↓	Runtime ↑
KDNT [16]⋆	BATCH	68.2	79.4	41.0%	19.0%	933	1093	11479	45605	0.7 Hz
LMP_p [17]⋆	BATCH	**71.0**	**80.2**	**46.9%**	21.9%	434	**587**	7880	**44564**	0.5 Hz
MCMOT_HDM [18]	BATCH	62.4	78.3	31.5%	24.2%	1394	1318	9855	57257	35 Hz
NOMTwSDP16 [19]	BATCH	62.2	79.6	32.5%	31.1%	**406**	642	**5119**	63352	3 Hz
EAMTT [20]	**ONLINE**	52.5	78.8	19.0%	34.9%	910	**1321**	**4407**	81223	12 Hz
POI [16]⋆	**ONLINE**	**66.1**	79.5	**34.0%**	20.8%	805	3093	5061	**55914**	10 Hz
SORT [12]⋆	**ONLINE**	59.8	**79.6**	25.4%	22.7%	1423	1835	8698	63245	**60 Hz**
Deep SORT (Ours)⋆	**ONLINE**	61.4	79.1	32.8%	**18.2%**	**781**	2008	12852	56668	40 Hz

Table 2: Tracking results on the MOT16 [15] challenge. We compare to other published methods with non-standard detections. The full table of results can be found on the challenge website. Methods marked with ⋆ use detections provided by [16].

tory [1] along with a script that can be used to generate features.

3. EXPERIMENTS

We assess the performance of our tracker on the MOT16 benchmark [15]. This benchmark evaluates tracking performance on seven challenging test sequences, including frontal-view scenes with moving camera as well as top-down surveillance setups. As input to our tracker we rely on detections provided by Yu et al. [16]. They have trained a Faster RCNN on a collection of public and private datasets to provide excellent performance. For a fair comparison, we have re-run SORT on the same detections.

Evaluation on test sequences were carried out using $\lambda = 0$ and $A_{\max} = 30$ frames. As in [16], detections have been thresholded at a confidence score of 0.3. The remaining parameters of our method have been found on separate training sequences which are provided by the benchmark. Evaluation is carried out according to the following metrics:

- Multi-object tracking accuracy (MOTA): Summary of overall tracking accuracy in terms of false positives, false negatives and identity switches [23].
- Multi-object tracking precision (MOTP): Summary of overall tracking precision in terms of bounding box overlap between ground-truth and reported location [23].
- Mostly tracked (MT): Percentage of ground-truth tracks that have the same label for at least 80% of their life span.
- Mostly lost(ML): Percentage of ground-truth tracks that are tracked for at most 20% of their life span.
- Identity switches (ID): Number of times the reported identity of a ground-truth track changes.
- Fragmentation (FM): Number of times a track is interrupted by a missing detection.

The results of our evaluation are shown in Table 2. Our adaptions successfully reduce the number of identity switches. In comparison to SORT, ID switches reduce from 1423 to 781. This is a decrease of approximately 45%. At the same time, track fragmentation increase slightly due to maintaining object identities through occlusions and misses. We also see a significant increase in number of mostly tracked objects and a decrease of mostly lost objects. Overall, due to integration of appearance information we successfully maintain identities through longer occlusions. This can also be seen by qualitative analysis of the tracking output that we provide in the supplementary material. An exemplary output of our tracker is shown in Figure 1.

Our method is also a strong competitor to other online tracking frameworks. In particular, our approach returns the fewest number of identity switches of all online methods while maintaining competitive MOTA scores, track fragmentations, and false negatives. The reported tracking accuracy is mostly impaired by a larger number of false positives. Given their overall impact on the MOTA score, applying a larger confidence threshold to the detections can potentially increase the reported performance of our algorithm by a large margin. However, visual inspection of the tracking output shows that these false positives are mostly generated from sporadic detector responses at static scene geometry. Due to our relatively large maximum allowed track age, these are more commonly joined to object trajectories. At the same time, we did not observe tracks jumping between false alarms frequently. Instead, the tracker commonly generated relatively stable, stationary tracks at the reported object location.

Our implementation runs at approximately 20 Hz with roughly half of the time spent on feature generation. Therefore, given a modern GPU, the system remains computationally efficient and operates at real time.

4. CONCLUSION

We have presented an extension to SORT that incorporates appearance information through a pre-trained association metric. Due to this extension, we are able to track through longer periods of occlusion, making SORT a strong competitor to state-of-the-art online tracking algorithms. Yet, the algorithm remains simple to implement and runs in real time.

[1] https://github.com/nwojke/deep_sort

5. REFERENCES

[1] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *CVPR*, 2008, pp. 1–8.

[2] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *CVPR*, 2011, pp. 1201–1208.

[3] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, 2011.

[4] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking," in *CVPR*, 2012, pp. 2034–2041.

[5] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *CVPR*, 2012, pp. 1918–1925.

[6] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *CVPR*, 2012, pp. 1926–1933.

[7] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *CVPR*, 2013, pp. 3682–3689.

[8] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, 1979.

[9] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–184, 1983.

[10] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *ICCV*, 2015, pp. 4696–4704.

[11] S.H. Rezatofighi, A. Milan, Z. Zhang, Qi. Shi, An. Dick, and I. Reid, "Joint probabilistic data association revisited," in *ICCV*, 2015, pp. 3047–3055.

[12] A. Bewley, G. Zongyuan, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *ICIP*, 2016, pp. 3464–3468.

[13] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[15] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[16] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *ECCV*. Springer, 2016, pp. 36–42.

[17] M. Keuper, S. Tang, Y. Zhongjie, B. Andres, T. Brox, and B. Schiele, "A multi-cut formulation for joint segmentation and tracking of multiple objects," *arXiv preprint arXiv:1607.06317*, 2016.

[18] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee, "Multi-class multi-object tracking using changing point detection," in *ECCV*. Springer, 2016, pp. 68–83.

[19] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *ICCV*, 2015, pp. 3029–3037.

[20] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *European Conference on Computer Vision*. Springer, 2016, pp. 84–99.

[21] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "MARS: A video benchmark for large-scale person re-identification," in *ECCV*, 2016.

[22] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016, pp. 1–12.

[23] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process*, vol. 2008, 2008.