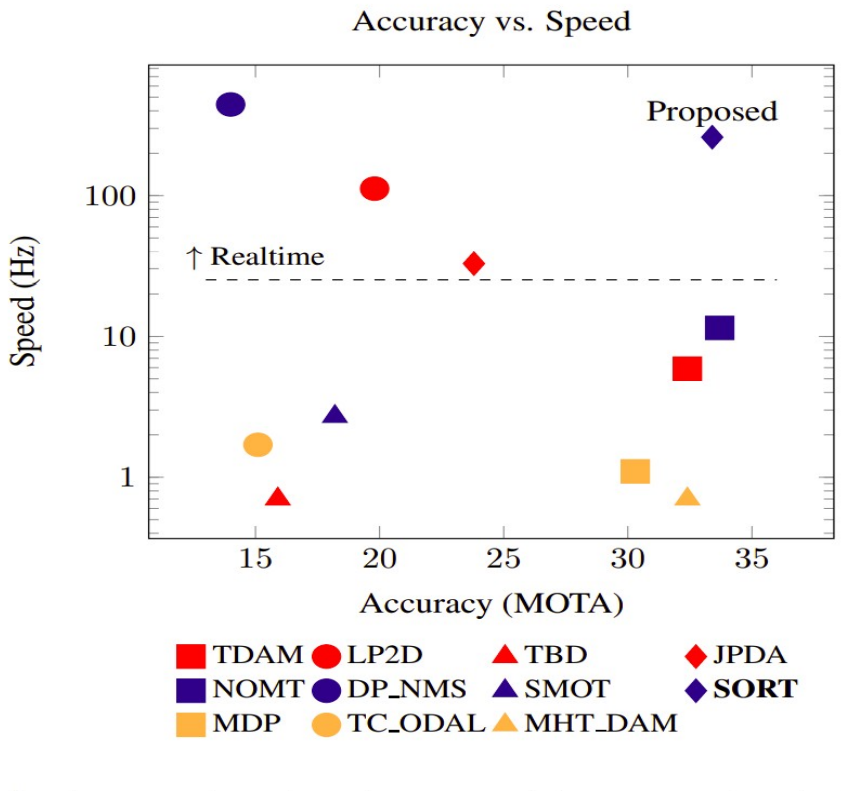
Smart Real Time Ordering

Abd al-Rahman al-Ktefane
Adel Kabboul
Ammar Abo Azan
Oday Mourad
Faculty of Information Technology,Damascus University

Introduction:

The MOT (multiple object tracking) problem can be viewed as a data association problem where the aim is to associate detection's across frames in a video sequence . Furthermore, the trade-off between accuracy and speed appears quite pronounced, since the speed of most accurate trackers is considered too slow for real time applications, this work is based on SORT [10], DeepSORT paper [11].



2. METHODOLOGY

2.1. Detection

As we are only interested in person tracking we ignore all other classes and only pass person detection results with output probabilities greater than ***IOU_threshold*** to the tracking framework. In our experiments, we found that the detection quality has a significant impact on tracking performance when comparing the ***FrRCNN*** [1] detection's with ***YOLO_v3*** [2] and ***YOLO_v4*** [3] and ***EfficientDet*** [4],which we found that the best detector (***EfficientDet0***) leads to the best tracking accuracy .

2.2. Estimation Model

Here we describe the object model,we approximate each object in frame with a linear constant velocity model which is independent of other objects and camera motion The state of each target is modeled as: $x=[u,v,s,r,\dot{u},\dot{v},\dot{s},\dot{r}]^T$

where ***u*** and ***v*** represent the horizontal and vertical pixel location of the center of the target, while the ***scale s*** and ***r*** represent the ***scale (area)*** and the aspect ratio of the target's bounding box respectively .

when a detection is associated to a target, the detected bounding box is used to update the target state where the velocity components are solved optimally via a ***Kalman filter framework*** [5]

If no detection is associated to the target, its state is simply predicted without correction using the linear velocity model.

2.3. Assignment Problem

In assigning detection's to existing targets, each target's bounding box geometry is estimated by predicting its new location in the current frame**.**

We propose two methodology to solve this problem the first one based on intersection-over-union ***(IOU) distance*** .The second approach is use ***Match Cascade*** algorithm and we'll talk more about this approaches later.

2.4 Intersection-Over-Union D*istance*

compute the cost matrix via intersection-over-union ***(IOU) distance*** between each detection and all predicted bounding boxes from the existing targets, then the assignment is solved optimally using the Hungarian algorithm.

2.5 Match Cascade

In this approach, we integrate motion and appearance information through combination of two appropriate metrics to improve the performance of ***(IOU) distance*** approach, Due to this extension we are able to track objects through longer periods of occlusions, effectively reducing the number of identity switches.

2.5.1. Motion Information

we use the (squared) Mahalanobis distance [6] between predicted Kalman states and newly arrived measurements : $d^{(1)}(i,j)=(d_j-y_i)^T S_i^{-1}(d_j-y_i)$ . . . (1)

where we denote the projection of the *i-th* track distribution into measurement space by (y_i,s_i) , and the *j-th* bounding box detection by 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.***

denote this decision with an indicator: $b_{i,j}^{(1)}=[d^{(1)}(i,j)\le t^{(1)}]$ . . . (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
$t^{(1)}=9.4877$.
PS: Mahalanobis distance is a measure of the distance between a point P and a distribution D, and it's a multi-dimensional generalization of the idea of measuring how many standard deviations away P is from the mean of D.
this distance is zero if P is at the mean of D and grows as P moves away from the mean.

2.5.2. appearance information
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 d_j we compute an appearance descriptor r_j with $\|r_j\|=1$ .Further, we keep a gallery $R_k=\{r_k^{(i)}\}_{k=1}^{L_k}$ of the last $L_k=100$ associated appearance descriptors for each track k. Then, our second metric measures the smallest cosine distance [7] between the i-th track and j-th detection in appearance space: $d^{(2)}(i,j)=min\{1-r_j^T r_k^{(i)} | r_k^{(i)} \in R_i\}$ . . . (3)
Again, we introduce a binary variable to indicate if an association is admissible according to this metric : $b_{i,j}^{(2)}=[d^{(2)}(i,j)\leq t^{(2)}]$ . . . (4)
In combination, both metrics complement each other by serving different aspects of the assignment problem. on the one hand, the ***Mahalanobis distance*** useful for ***short-term predictions*** and on the other hand, the ***cosine distance*** 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
$c_{i,j}=\lambda d_{(i,j)}^{(1)}+(1-\lambda) d_{(i,j)}^{(2)}$ . . . (5)
during our experiments we found that setting $\boldsymbol{\lambda = 0}$ is a reasonable choice when there is substantial ***camera motion***. In this setting, only appearance information are used in the association cost term.
where we call an association admissible if it is within the gating region of both metrics:
$b_{i,j}=\prod_{m=1}^{2} b_{i,j}^{(m)}$. . . . (6)
full algorithm:
Input: Track indices $\boldsymbol{T}$ = {1, . . . , N}, Detection indices $\boldsymbol{D}$ = {1, . . . , M}, Maximum age A_{max}
1: Compute cost matrix $C=[c_{i,j}]$ using Equation. 5
2: Compute gate matrix $B=[b_{i,j}]$ using Equation. 6
3: Initialize set of matches $\boldsymbol{M} \leftarrow \varnothing$
4: Initialize set of unmatched detection's $\boldsymbol{U} \leftarrow \boldsymbol{D}$
5: for n $\in$ {1, . . . , A_{max} } do
6: Select tracks by age $T_n \leftarrow$ {i $\in \boldsymbol{T}$ | a_i = n}
7: $[x_{i,j}] \leftarrow$ min cost matching (C, T_n, U)
8: $\boldsymbol{M} \leftarrow \boldsymbol{M} \cup$ {(i, j) | $b_{i,j}.x_{i,j}>0$ }
9: $\boldsymbol{U} \leftarrow \boldsymbol{U} \setminus$ {j | $\sum_i b_{i,j}.x_{i,j}>0$ }
10: ***end for***
11: ***return M, U***

Name	Patch Size/Stride	Output Size
Conv 1	3 × 3/1	32 × 128 × 64
Conv 2	3 × 3/1	32 × 128 × 64
Max Pool 3	3 × 3/2	32 × 64 × 32
Residual 4	3 × 3/1	32 × 64 × 32
Residual 5	3 × 3/1	32 × 64 × 32
Residual 6	3 × 3/2	64 × 32 × 16
Residual 7	3 × 3/1	64 × 32 × 16
Residual 8	3 × 3/2	128 × 16 × 8
Residual 9	3 × 3/1	128 × 16 × 8
Dense 10		128
Batch and L2 normalization		128

Table 1: Overview of the CNN architecture.

2.6. Deep Appearance Descriptor

we employ a CNN that has been trained on a large-scale person re-identification data set [8] 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 [9] with two convolutional layers followed by six residual blocks. The global feature map of dimensionality 128 is computed in dense layer 10. A final batch and L2 normalization projects features onto the unit hyper-sphere 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.

2.7. Creation and Deletion of Track Identities

When objects enter and leave the image, unique identities need to be created or destroyed accordingly. For creating trackers, we consider any detection with an overlap less than ***IOU_min*** to signify the existence of an untracked object.
The tracker is initialized using the geometry of the bounding box with the velocity set to zero. Since the velocity is unobserved at this point the covariance of the velocity component is initialized with large values, reflecting this uncertainty. Additionally, the new tracker then undergoes a probationary period (***n_init***) where the target needs to be associated with detection's to accumulate enough evidence in order to prevent tracking of false positives.
For each track ***k*** we count the number of frames since the last successful measurement association ***'time_since_update'***. This counter is incremented during Kalman filter prediction and reset to 0 when the track has been associated with a measurement. when the track has been associated with a measurement. Tracks that exceed a predefined maximum age ***Max_Age*** are considered to have left the scene and this prevents an unbounded growth in the number of trackers and localization errors caused by predictions over long duration's without corrections from the detector.
In our experiments ***Max_Age*** is set to 15 and When track deleted we're re-sort tracks id's.

3. Future Plan

1-Add the idea of importance and priority to arrange the trackers.
2-Add facial recognition feature and link it with a database
3-Add the ability to track more than one queue for more than one service
4-Add a talking assistant to organize the proces

4. CONCLUSION

In this work, a simple online tracking framework is presented that focuses on frame-to-frame prediction and association. We showed that the tracking quality is highly dependent on detection performance and by capitalizing on recent developments in detection, state-of-the-art tracking quality can be achieved with only classical tracking methods. The presented framework achieves best in class performance with respect to both speed and accuracy, while other methods typically sacrifice one for the other.

5. REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Advances in Neural Information Processing Systems, 2015. https://arxiv.org/pdf/1506.01497.pdf

[2] Joseph Redmon, Ali Farhadi University of Washington,YOLOv3: An Incremental Improvement. https://pjreddie.com/media/files/papers/YOLOv3.pdf

[3] YOLOv4: Optimal Speed and Accuracy of Object Detection. https://arxiv.org/pdf/2004.10934.pdf

[4] Mingxing Tan Ruoming Pang Quoc V. Le Google Research, Brain Team, EfficientDet: Scalable and Efficient Object Detection. https://arxiv.org/pdf/1911.09070.pdf

[5] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, vol. 82, no. Series D, pp. 35–45, 1960 https://www.researchgate.net/publication/236897001_The_Kalman_Filter_and_Related_Algorithms_A_Literature_Review

[6] Mahalanobis distance https://en.wikipedia.org/wiki/Mahalanobis_distance. http://insa.nic.in/writereaddata/UpLoadedFiles/PINSA/Vol02_1936_1_Art05.pdf.

[7] cosine similarity https://en.wikipedia.org/wiki/Cosine_similarity

[8] 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. https://www.researchgate.net/publication/308277502_MARS_A_Video_Benchmark_for_Large-Scale_Person_Re-Identification

[9] S. Zagoruyko and N. Komodakis, "Wide residual networks," in BMVC, 2016, pp. 1–12. https://arxiv.org/pdf/1605.07146.pdf

[10] Alex Bewley, Zongyuan Ge , Lionel Ott, Fabio Ramos , Ben Upcroft, SIMPLE ONLINE AND REALTIME TRACKING https://arxiv.org/pdf/1602.00763.pdf.

[11] Nicolai Wojke, Alex Bewley , Dietrich Paulus, SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC https://arxiv.org/pdf/1703.07402.pdf.