Multiple Object Tracking with Mixture Density Networks for Trajectory Estimation

Andreu Girbau[1,2], Xavier Giró-i-Nieto[1], Ignasi Rius[2] and Ferran Marqués[1]

[1]Universitat Politècnica de Catalunya [2]AutomaticTV

{andreu.girbau, xavier.giro, ferran.marques}@upc.edu, irius@mediapro.tv

Abstract

In this work, we show that trajectory estimation can become a key factor for multiple object tracking, and present TrajE, a trajectory estimator based on recurrent mixture density networks, as a generic module that can be added to existing object trackers. To provide several trajectory hypotheses, our method uses beam search. Also, relying on the same estimated trajectory, we propose to reconstruct a track after an occlusion occurs. We integrate TrajE into two state of the art tracking algorithms, CenterTrack [7] and Tracktor [1], boosting their respective performances in the MOTChallenge 2017 test set by 6.3 and 0.3 points in MOTA score, and 1.8 and 3.1 in IDF1.

1. Introduction

In this work, we propose to use trajectory estimation for the purpose of multiple object tracking. We introduce *TrajE*, a ***Traj***ectory ***E***stimator based on a recurrent mixture density network, that learns to estimate the underlying distribution of an object trajectory. By sampling from such a distribution, multiple hypotheses for the most likely position of the object can be forecasted, giving the tracker a prior on the subsequent object position. We combine this model with a beam search technique in order to explore multiple trajectory hypotheses per object. We provide a lightweight implementation of the model, designing TrajE as a single layer recurrent neural network, capable of estimating the objects trajectories.

We have trained *TrajE* with pedestrian trajectories in the MOT challenge dataset [5], and included it as the module of motion estimation for two existing state of the art multiple object trackers, *CenterTrack* [7] and *Tracktor* [1].

The contributions of this work are (i) we show empirically that trajectory estimation can be a key factor for a better tracking performance, (ii) we build TrajE, a lightweight model for trajectory estimation based on mixture density networks that can be used as a generic motion model for many trackers, (iii) we add it to two state of the art multiple object trackers, boosting their performance by a considerable margin, setting a new state of the art for CenterTrack + TrajE in the MOT17 dataset.

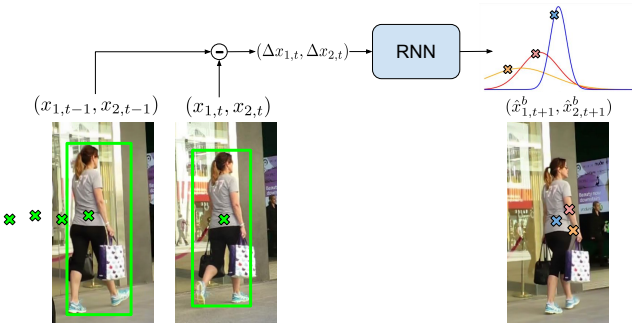

Figure 1. Concept of TrajE. We train a Recurrent Neural Network to estimate a mixture of Gaussians that model the object trajectory. From it, several hypotheses (B) of the object position in the next time step are sampled.

2. Related Work

Multiple object tracking. Advances in object detection have allowed multiple object trackers to rely on frame-by-frame detections. In consequence, most current algorithms follow the tracking-by-detection paradigm, addressing the tracking problem in two steps: (i) object detection and (ii) association of detections through time to form trajectories.

Trajectory estimation. [6] classifies the different human trajectory prediction methods, based on the model: (i) Physics-based methods, where motion is predicted by dynamics equations based on physical models; (ii) Pattern-based methods, where the dynamics are learned from data; and (iii) Planning-based methods, where there is a reasoning on the agent actions. These models can use (or not) available contextual cues, such as (i) Target agent cues, which are available target agent information; (ii) Dynamic environmental cues, where the target agent is aware of other agents; and (iii) Static environmental cues, where the target agent is aware of the environment information (e.g. static obstacles, such as trees or buildings).

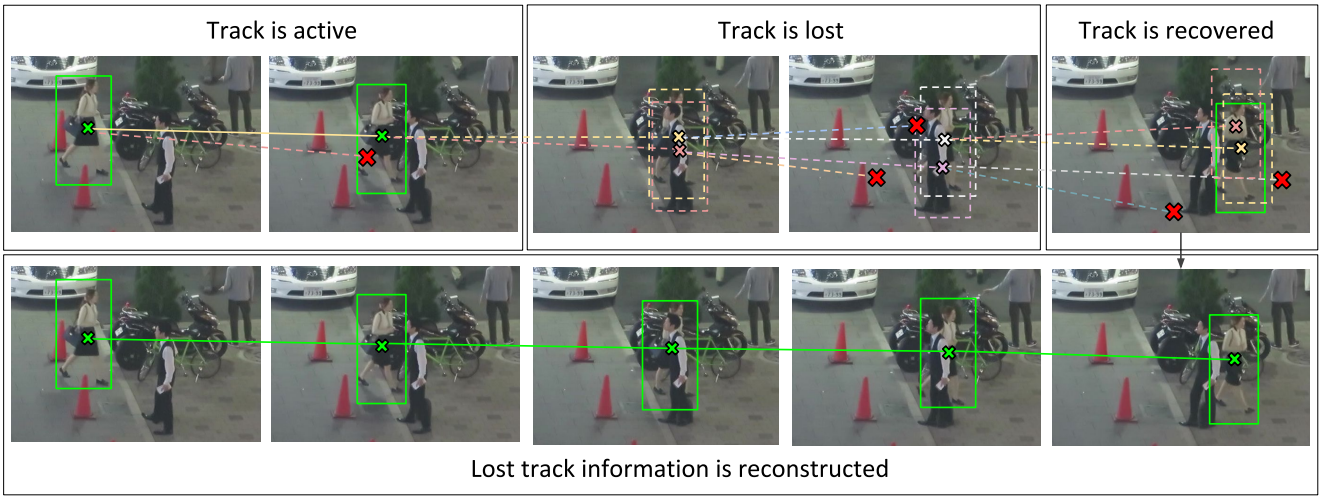

Figure 2. Visualization of the beam search and occlusion handling. First, B (here, $B = 2$) hypotheses of where the object could be in the next time step are sampled from the generated trajectory distribution. If a detection is associated to a track, the beam search corresponding to that track is reset. If no detection is associated to an active track, several hypotheses are sampled from the trajectory distribution and pruned (red crosses) in the next time step to keep B hypotheses. If the track is recovered (a detection is assigned to the track again), and the estimated trajectory is coherent, the computed trajectory during the lost state of the track is added to the object track. The occlusion-filling bounding boxes are generated using the centroids of the estimated trajectory, and the width and height of the new detection.

Lately, the trend in the field is to use pattern-based methods. They follow the *Sense - Learn - Predict* paradigm, and learn motion behaviors by fitting different function approximators (i.e. neural networks, hidden Markov models, or Gaussian processes) to data.

In this work, we propose to use the trajectory information for the purpose of multiple object tracking, and implemented TrajE for this purpose.

3. Multiple Object Tracking using MDNs

In this section we introduce TrajE, the main techniques that form it, the baseline trackers where we have integrated TrajE, and experiments.

Mixture Density Networks. TrajE estimates independently each object trajectory by using a recurrent (GRU) MDN (Mixture Density Network) [3]. Its input is a point, i.e. the centroid of the detection in t or the estimated trajectory from $t - \tau$ to t, and its output are the parameters of M Gaussians (we set $M = 5$), to form a GMM that models the evolution of the trajectory in $t + 1$. Figure 1 depicts this concept. From such a distribution, we sample a set of hypotheses for the object position in the next time step.

Biased sampling. Directly sampling from the generated distribution leads to a huge space of possible trajectories per object, which may not be the best solution. To palliate this, we bias (sharpen) the distribution as in [4] by increasing the weight of a Gaussian forming the probability distribution, and reducing the variance by a bias b factor.

Beam search. We use beam search for multiple trajectory hypotheses. Beam search is a well known exploration technique in Natural Language Processing (NLP), used in tasks like machine translation. Its core idea is to explore a finite set of possibilities, known as beam width B, in order to maximize the likelihood of a sentence being outputted.

We adapt beam search for trajectory estimation to keep several trajectory hypotheses alive. First, we sample B trajectory hypotheses from the generated distribution. Then, these B hypotheses are used to detect the object in the next frame. If the object is detected, the beam search is reset. If it is still missing, we input the B trajectory hypotheses to B instances of TrajE, and sample from each one B new hypotheses, ending up with B^2. To avoid the exponential growth, these B^2 samples are pruned to B samples by using a maximum likelihood criteria.

Exploration strategies. To generate trajectories, we propose three approaches based on the exploration strategy. The first one is the Best Mean (BM), which takes the mean value of the Gaussian with highest weight from the Gaussians forming the trajectory distribution outputted by the MDN. The second one is a Greedy Beam Search (GBS). It samples from the trajectory distribution, and it takes the sample with highest likelihood at every time step. The last strategy is the Pure Beam Search (PBS), which uses all B hypotheses to forward the track in B possible ways. If a detection is associated to the track, the best beam given the historic is chosen in order to keep a single detection per track at every time step. Note that, for $B = 1$, both GBS and PBS strategies become the same.

Occlusion reconstruction. If a track is recovered after an occlusion, we can use the estimated trajectory to reconstruct the track while lost. In Figure 2, we show an example of the system handling an occlusion.

Baselines. We integrated TrajE, into two existing multiple object trackers, *CenterTrack* [7] and *Tracktor* [1].
Both are defined as trackers-by-regression, meaning that, by "regressing" the detections in $t-1$ to detections in t, they are able to assign the newly detected object in t to the same track where the original object in $t-1$ belonged.
CenterTrack extends CenterNet object detector to associate detections over time by feeding the model two consecutive images. To further boost their performance, the model computes, as motion estimation, the offset between detections in two consecutive time steps.
Tracktor regresses the tracked objects using the ROI-pooling layer present in Faster-RCNN, as if they were object proposals computed by the network. Also, it uses re-identification with siamese networks, and a motion model based on the Constant Velocity (CV) assumption and Camera Motion Compensation (CMC).
We substitute the motion estimation of both trackers with TrajE, which forecasts the objects trajectories, and allows to reconstruct occluded tracks by using such information.

3.1. Experiments

Experiments are performed on the MOTChallenge 2017 (MOT17) dataset, which contains 14 video sequences recording pedestrians (7 for training, 7 for testing) with both static and moving camera, recorded in the wild (unconstrained environments) in different locations. For both CenterTrack and Tracktor, we use their provided object detection models, and the re-identification in the case of Tracktor.

Training. To train TrajE, we minimize the negative log-likelihood between an estimated distribution and a ground truth sample for every time step. We used MOT17 data by sampling random trajectories of 100 time steps from objects in different sequences. Also, we apply noise to the input sequences to make our model more robust to noisy detections.

Incremental study. Table 1 and Table 2 present a study on whether CenterTrack and Tracktor benefit from estimating the object trajectories.
The starting point are the trackers without and with their own motion estimation. We then swap this motion estimation (OFF or CV+CMC in tables 1, 2) for TrajE, and study the impact of the three exploration strategies, Best Mean (BM), Greedy Beam Search (GBS), and Pure Beam Search (PBS) with and without occlusion reconstruction using the estimated trajectories. To see whether the benefits of including trajectory information into object trackers generalize, we adapt the Kalman filter implemented in SORT [2] for trajectory estimation, by generating trajectories for a track whose state is set to lost, while including the new detections assigned to the track whenever its state is set to active.
The first observation is that any provided motion information is helpful to the trackers. In Table 1, by computing the motion with an offset head (OFF) for CenterTrack, and Table 2, by assuming a constant velocity (CV), and using Camera Motion Compensation (CMC) in Tracktor.

	MOTA ↑	IDF1 ↑	HOTA ↑	DetA ↑	AssA ↑	IDSW ↓
CenterTrack	67.4	62.7	63.0	69.3	57.2	1356
+OFF	67.7	63.8	63.8	69.2	58.8	1077
+Kalman	68.7	64.4	64.0	70.9	57.7	1642
+BM	68.4	65.6	64.9	69.9	60.2	833
+GBS	68.9 ± 0.3	65.6 ± 0.5	65.1 ± 0.5	70.3 ± 0.3	60.2 ± 1.0	857
+PBS	69.0 ± 0.2	66.0 ± 0.1	65.3 ± 0.2	70.4 ± 0.1	60.5 ± 0.4	789
+Kalman+occ	68.8	64.6	64.1	**71.2**	57.8	1382
+BM+occ	69.1	65.9	65.4	70.6	60.5	732
+GBS+occ	69.5 ± 0.2	66.0 ± 0.5	65.5 ± 0.5	70.9 ± 0.2	60.6 ± 1.1	751
+PBS+occ	**69.6** ± 0.1	**66.3** ± 0.1	**65.7** ± 0.2	71.0 ± 0.1	**60.9** ± 0.4	**706**

Table 1. Incremental study on CenterTrack+TrajE with the exploration strategies, GBS, PBS, and BM, and CenterTrack+Kalman filter for trajectory estimation. For PBS and GBS we depict the mean and variance over 5 runs. OFF stands for offset prediction present in CenterTrack, and occ for occlusion reconstruction from trajectories. We compare the performance on the full MOT17 training set using the Faster R-CNN public detections, with $bias = 1$, $B = 5$.

	MOTA ↑	IDF1 ↑	HOTA ↑	DetA ↑	AssA ↑	IDSW ↓
Tracktor v2	60.6	62.0	60.7	61.6	59.7	913
+CV	61.2	63.9	62.0	61.9	62.1	557
+CV+CMC	61.7	64.9	62.8	62.1	63.5	**269**
+Kalman	61.4	65.2	63.1	62.0	64.2	462
+BM	61.6	66.4	63.7	62.1	65.3	399
+GBS	61.4 ± 0.1	66.6 ± 0.5	63.6 ± 0.3	62.0 ± 0.1	65.4 ± 0.6	407
+PBS	61.5 ± 0.1	66.7 ± 0.5	63.7 ± 0.3	62.1 ± 0.1	65.5 ± 0.6	369
+Kalman+occ	61.2	66.9	**64.4**	**62.8**	**66.2**	506
+BM+occ	**62.2**	66.8	64.2	**62.8**	65.7	399
+GBS+occ	61.9 ± 0.1	66.9 ± 0.5	64.1 ± 0.3	62.5 ± 0.1	65.7 ± 0.6	409
+PBS+occ	62.0 ± 0.1	**67.0** ± 0.4	64.2 ± 0.3	62.6 ± 0.1	65.8 ± 0.6	370

Table 2. Incremental study on Tracktor+TrajE with the exploration strategies, GBS, PBS, and BM, and Tracktor+Kalman filter for trajectory estimation. For PBS and GBS we depict the mean and variance over 5 runs. CV stands for Constant Velocity assumption, CMC for Camera Motion Compensation, and occ for occlusion reconstruction from trajectories. We compare the performance on the full MOT17 training set using the Faster R-CNN public detections, with $bias = 1$, $B = 5$.

Trajectory estimation. Following the incremental study, we observe that, by adding TrajE, both trackers consistently improve in all metrics. As TrajE has a sampling step, the mean and variance for five runs is depicted.

TrajE parameters. In Figure 3 and Figure 4, we study the impact of the *bias* and beam width (B) values for the different trajectory estimation strategies.
The results show how, by using beam search, the trackers benefit from the several hypotheses, and how biasing the outputs of the MDN can become crucial in the trajectory estimation. Regarding the exploration strategies, PBS has better overall results than GBS. Interestingly, by using the BM strategy, which is equivalent to use a beam width of $B = 1$ with $bias \to \infty$, TrajE also boosts the performance of the trackers by a considerable margin, and can be a good

alternative when a faster computation is mandatory.

Comparison with state of the art. In Table 3 we compare the two trackers using TrajE with the PBS setting that lead to the best results on the training set ($bias = 1, B = 5$), with and without occlusion reconstruction, with respect to the state of the art of multiple object trackers in the test set of the MOTChallenge dataset. By using TrajE to predict trajectories, their performance in both MOTA and IDF1 scores is boosted by a considerable margin, and set a new state of the art results in the case of CenterTrack + TrajE, with an increase of 5.9, and 6.3 points in the MOTA score without and with occlusion reconstruction respectively.

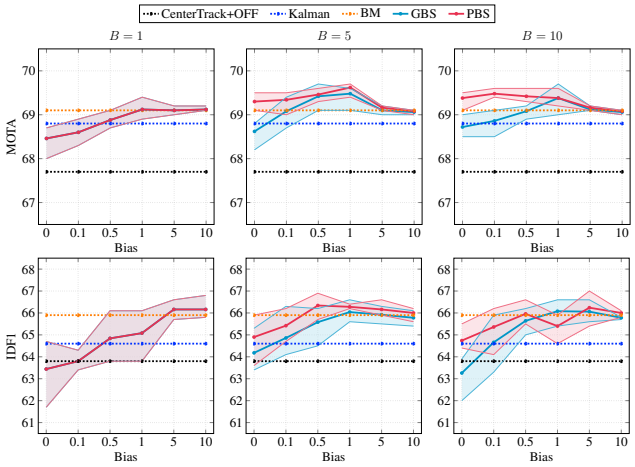

Figure 3. Experiments over TrajE parameters $bias$ and beam width B for CenterTrack using TrajE or Kalman with occlusion reconstruction. The y-axis corresponds to the metric score, the x-axis to the $bias$, and the columns to the different B values. Solid lines correspond to the average value over five runs, and the transparent region limits correspond to the maximum and minimum values.

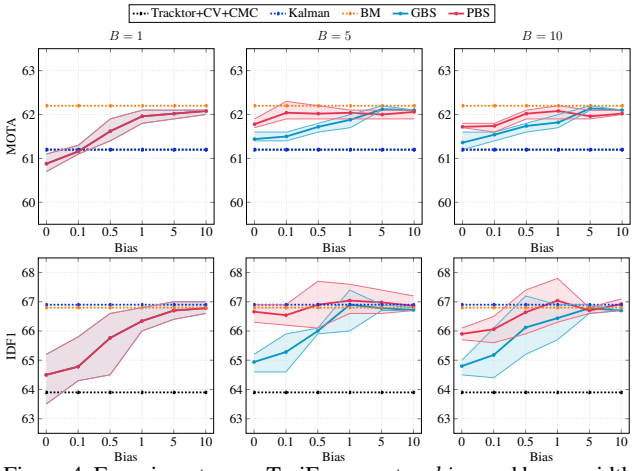

Figure 4. Experiments over TrajE parameters $bias$ and beam width B for Tracktor using TrajE or Kalman with occlusion reconstruction. The y-axis correspond to the metric score, the x-axis to the $bias$, and the columns to the different B values. Solid lines correspond to the average value over five runs, and the transparent region limits correspond to the maximum and minimum values.

MOT17 [5]						
Method	MOTA ↑	IDF1 ↑	MT % ↑	ML % ↓	FP ↓	FN ↓
Online, TrajE without occlusion reconstruction						
CenterTrack+TrajE	**67.4**(+5.9)	**61.2**(+1.6)	**34.8**	**24.9**	18652	**161347**
CenterTrack	61.5	59.6	26.4	31.9	14076	200672
GSM	56.4	57.8	22.2	34.5	14379	230174
Tracktor(v2)+TrajE	56.3(+0.0)	57.8(+2.7)	21.4	35.8	10068	233885
Tracktor(v2)	56.3	55.1	21.1	35.3	**8866**	235449
FAMNet	52.0	48.7	19.1	33.4	14138	253616
Offline, TrajE with occlusion reconstruction						
CenterTrack+TrajE	**67.8**(+6.3)	61.4(+1.8)	**36.0**	**24.5**	20982	**157468**
Lif_T	60.5	**65.6**	27.0	33.6	14966	206619
MPNTrack	58.8	61.7	28.8	33.5	17413	213594
Tracktor(v2)+TrajE	56.6(+0.3)	58.2(+3.1)	21.9	35.7	**10119**	231091
TT17	54.9	63.1	24.4	38.1	20236	233295
TPM	54.2	52.6	22.8	37.5	13739	242730
JBNOT	52.6	50.8	19.7	35.8	31572	232659

Table 3. Comparison integrating *TrajE* to CenterTrack and Tracktor against the state of the art methods on the test set of MOTChallenge 17 dataset using public detections. In bold tracker + our method (TrajE). In red the best result, in blue the second best. The tracking baselines are CenterTrack+OFF, Tracktor+CV+CMC.

4. Conclusions

We have introduced *TrajE*, a lightweight trajectory estimator based on mixture density networks and beam search, capable of significantly increasing the performance of existing multiple object trackers. Also, with the same estimated trajectory, we propose to do a track reconstruction when the object is lost due occlusions. Our experiments adding our trajectory estimator to *CenterTrack*, and *Tracktor* provide very interesting insights on how the trajectory estimation can help in the tracking, while establishing a new state of the art in the MOTChallenge 2017 dataset.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 941–951, 2019. 1, 3

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468. IEEE, 2016. 3

[3] Christopher M Bishop. Mixture density networks. 1994. 2

[4] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 2

[5] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831. 1, 4

[6] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *arXiv preprint arXiv:1905.06113*, 2019. 1

[7] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *arXiv preprint arXiv:2004.01177*, 2020. 1, 3