CV3DST MultiObject Tracking Challenge

Christopher Heumann
Technical University of Munich
ga63xir@mytum.de

Abstract

My plan for this work was to improve the baseline tracker by small adjustments and an extensive hyperparameter search to make it more robust to detection inaccuracies. The resulting tracker is still an online, non neural network tracker where the association step is based on the IOU score between tracks and detections. Some improvements are based on the IOU tracker by Bochinski et al. [1], whose basic principle is illustrated in figure 1. The code is available at `https://github.com/heumchri/cv3dst_exercise`.

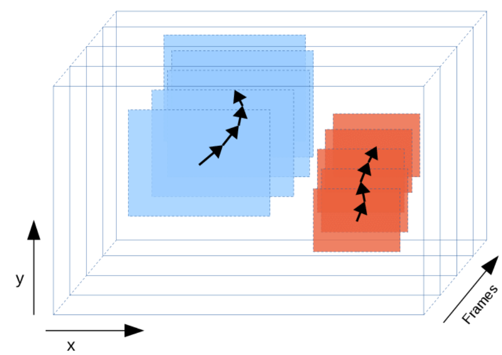

Figure 1. Basic principle of the IOU tracker [1]. Tracks are continued by the detection with the closest IOU distance.

1. Improvements to the Tracker

1.1. detections can be assigned to only one track

The baseline tracker can assign a single detection to multiple tracks if it is the closest detection to all of them. I adjusted the algorithm such that the detection is greedily assigned to the first track to which it is the closest. The MOTA score improved from the 26.3% of the original baseline tracker to 52.6%.

1.2. unassigned detection initialize new tracks

The baseline tracker initializes new tracks from detections that are far away (IOU distance > 0.5) of any tracks. This creates a virtual inter-frame non maximum supression. Because the detector already has NMS, I tried initializing new tracks with all unmatched detections which increased the MOTA to 66.3%.

1.3. tracks are extended when detections are close enough

The baseline tracker only associates detections to tracks if their IOU distance is ≤ 0.5. I made this value the hyperparameter σ_{IOU} and used it in the grid search. Because it is the IOU distance and not the IOU, it is the reverse hyperparameter of σ_{IOU} in [1].

1.4. remove short tracks

Like in [1], I only considered tracks with a length of at least t_{min}, which is a hyperparameter.

1.5. remove tracks with uncertain detections

Like in [1], I only considered tracks with at least one detection with a score of at least σ_h, which is a hyperparameter. The hyperparameters t_{min} and σ_h can be included in the online tracker, but I decided to filter the tracks in postprocessing to speed up the grid search.

1.6. extensive hyperparameter search

I did an extensive grid search of the following hyperparameters by evaluating the performance of the tracker on the entire training set. The NMS threshold for the object detector $nms \in [0.2, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6, 0.7, 0.8]$. $\sigma_{IOU} \in [0.2, 0.3, 0.4, 0.5, 0.6]$. $\sigma_h \in [0, 0.5, 0.9, 0.95, 0.99]$. $t_{min} \in [1, 2, 3, 4, 5]$. I also included a hyperparameter to switch between the two ways how new tracks are initialized described in subsection 1.2 and the number of frames a track is allowed to skip from the method in subsection 2.2. Because tracking and evaluating the metrics on the entire training set takes a long time, these two and other hyperparameters that showed no promising result were excluded before finishing the complete grid search.

2. Failed improvements

2.1. merge tracks in postprocessing

The tracker terminates a track when it cannot be matched to a detection in a single frame and cannot reidentify tracks in the future. To counteract missing frames, I tried to merge tracks that probably belong to the same object in postprocessing. If the end of one track and the start of another track are less than a certain amount of frames apart and their boxes overlap by a certain amount, the tracks are merged. This method did not result in any improvements. For example, for the hyperparameters $nms = 0.4, \sigma_{IOU} = 0.2, \sigma_h = 0.95, t_{min} = 2$ and allowing frameskips of $[0, 1, 2, 3, 4, 5]$ the resulting MOTA scores on the training set are $[68.7\%, 66.2\%, 63.1\%, 62.8\%, 61.3\%, 60.6\%]$ and the IDF scores are $[59.4\%, 59.2\%, 58.4\%, 58.3\%, 57.8\%, 57.8\%]$. The method was not used for generating the final tracks.

2.2. allow frame skips

This method allows to reidentify tracks while the tracker is running. If no detection is found for a track, the track gets deactivated instead of terminated. The tracker tries to associate detections to the active as well as the inactive tracks. If a track has been inactive for a certain amount of frames, it finally is terminated. This method did improve the IDF1 score while decreasing the MOTA score. For example, for the hyperparameters $nms = 0.4, \sigma_{IOU} = 0.2, \sigma_h = 0.95, t_{min} = 2$ and allowing frameskips of $[0, 1, 2, 3]$ the resulting MOTA scores on the training set are $[68.7\%, 68.5\%, 68.3\%, 68.2\%]$ and the IDF scores are $[59.4\%, 59.8\%, 60.0\%, 60.1\%]$. The method was not used for generating the final tracks.

3. Results

The best MOTA score on the training set was achieved with $nms = 0.4, \sigma_{IOU} = 0.2, \sigma_h = 0.95, t_{min} = 2$, which resulted in an MOTA of 68.7% and an IDF1 score of 59.4% on the training set and 64.71% and 52.96% on the test set. The best IDF1 score on the training set was achieved with $nms = 0.45, \sigma_{IOU} = 0.2, \sigma_h = 0.99, t_{min} = 5$, which resulted in an MOTA of 68.1% and an IDF1 score of 60.3% on the training set and 65.61% and 54.65% on the test set, which fulfill the bonus thresholds.

References

[1] E. Bochinski, V. Eiselein, and T. Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017.