A Fast Unified System for 3D Object Detection and Tracking Supplementary Material

Thomas Heitzinger and Martin Kampel
Computer Vision Lab, TU Wien
Vienna, Austria
{thomas.heitzinger, martin.kampel}@tuwien.ac.at

1. Implementation Details

The following sections go into the details of our implementation of FUS3D and discuss the nuances involved in training our system.

Loss function and matching cost A bounding box predicted by the FUS3D system consists of the attributes: box-center location $loc \in \mathbb{R}^3$, box dimensions $dim \in \mathbb{R}^3$, facing direction as a normalized orientation vector in the ground plane $dir \in \mathbb{R}^2$, a confidence score $conf \in \mathbb{R}$, a centerness attribute $center \in \mathbb{R}$ and a class probability distribution $cls \in \mathbb{R}^{n_{cls}}$, where n_{cls} is the number of object classes considered in the given setting.

For a matched pair of a ground truth bounding box $\boldsymbol{b}$ and a predicted bounding box $\hat{\boldsymbol{b}}$, generated either by the FUS3D detection stage or the FUS3D tracking stage, the training loss is a weighted sum of losses corresponding to the individual attributes:

$$\begin{aligned}\mathcal{L}_{box}(\boldsymbol{b}, \hat{\boldsymbol{b}}) := & \; c_{loc}\,\mathrm{MSE}(\boldsymbol{b}_{loc}, \hat{\boldsymbol{b}}_{loc}) \\ & + c_{dim}\,\mathrm{MSE}(\boldsymbol{b}_{dim}, \hat{\boldsymbol{b}}_{dim}) \\ & + c_{dir}\,(1 - \boldsymbol{b}_{dir} \cdot \hat{\boldsymbol{b}}_{dir}) \\ & + c_{center}\,\mathrm{MSE}(\boldsymbol{b}_{center}, \hat{\boldsymbol{b}}_{center}) \\ & + c_{cls}\,\mathrm{CE}(\boldsymbol{b}_{cls}, \hat{\boldsymbol{b}}_{cls}) \\ & + c_{conf}\,\mathrm{BCE}(1, \hat{\boldsymbol{b}}_{conf}),\end{aligned} \tag{1}$$

where MSE is the mean squared error, CE denotes the cross-entropy between two class distributions, BCE is the binary cross-entropy loss, and all constants c_* are non-negative scalars. For unmatched predictions, the loss simplifies so that only the term corresponding to the box confidence remains, i.e. $\mathcal{L}_{box}(\hat{\boldsymbol{b}}) := c_{conf}\,\mathrm{BCE}(0, \hat{\boldsymbol{b}}_{conf})$.

Apart from scaling constants, the box loss function $\mathcal{L}_{box}$ given in *Equation* (1) is nearly identical between the FUS3D detection and the tracking stages. The only notable difference are 1. the absence of the centerness loss in the tracking stage (i.e. $c_{center} = 0$), and 2. the location target for the tracking stage is given as a position in world space, whereas for the detection stage we instead predict the offset between the associated cell center and the ground truth location. Furthermore, we experiment with the additional use of a holistic differentiable 3D $\mathcal{L}_{GIoU}$ or $\mathcal{L}_{DIoU}$ loss [6]. Our experiments show that they can provide a small gain in detection performance, but are coupled with a notable increase in training time per batch due to their inherent complexity. This is despite the availability of efficient implementations [1].

The process of matching predictions to ground truth bounding boxes requires further distinction between the detection and tracking stage:

- **Detection** In the detection stage, each prediction is associated with a cell, which in turn represents a region in 3D space. It is matched with a ground truth bounding box if and only if the corresponding cell center lies within the box interior.
- **Tracking** In the tracking stage, matching of newly acquired tracks is done using the Hungarian Algorithm [2]. The matching cost between a ground truth box $\boldsymbol{b}$ and an anchor box $\hat{\boldsymbol{b}}$ is given by:

  $$\mathcal{C}_{match}(\boldsymbol{b}, \hat{\boldsymbol{b}}) := \|\boldsymbol{b}_{loc} - \hat{\boldsymbol{b}}_{loc}\|_2 + \mathcal{L}_{DIoU}(\boldsymbol{b}, \hat{\boldsymbol{b}}), \tag{2}$$

  where $\mathcal{L}_{DIoU}$ is a 3D version of the Distance-IoU loss [5]. Matching with existing tracks is achieved implicitly by enforcing that the track query appears at the same latent token index as on the previous time step.

For a set of ground truth objects $\mathcal{B}$ and a set of predicted objects $\hat{\mathcal{B}}$ at time step t, let $\sigma^t_{match} \subseteq \mathcal{B} \times \hat{\mathcal{B}}$ be the set of matched bounding box pairs and let $\sigma^t_{neg} \subseteq \hat{\mathcal{B}}$ be the set of unmatched predictions. The combined loss for a single time

[1]https://github.com/lilanxiao/Rotated_IoU

step t is thus given as

$$\mathcal{L}_{stage}^{t} := \sum_{(\boldsymbol{b},\hat{\boldsymbol{b}})\in\sigma_{match}^{t}} \mathcal{L}_{box}(\boldsymbol{b},\hat{\boldsymbol{b}}) + \sum_{\hat{\boldsymbol{b}}\in\sigma_{neg}^{t}} \mathcal{L}_{box}(\hat{\boldsymbol{b}}), \tag{3}$$

where $stage \in \{det, track\}$ indicates if the loss is applied to the FUS3D detection or tracking stage.

Finally, since the full FUS3D system must perform propagation and association of tracks over time, we have to define a loss that is applied over a continuous time interval. This total loss $\mathcal{L}_{total}$, applied over a period of n time steps, is given as a weighted sum over the individual single-frame losses:

$$\mathcal{L}_{total}^{t} := \sum_{i=0}^{n-1} \lambda_{t-i} \left(\mathcal{L}_{det}^{t-i} + \mathcal{L}_{track}^{t-i} \right), \tag{4}$$

where $(\lambda_1, \ldots, \lambda_n)$ is a sequence of positive scalars weighting the losses of the individual time steps according to their importance. We obtain best results with a monotonically decreasing sequence, thus giving higher importance to initial track acquisition and reduced weight on later propagation. The length of the training period n is chosen in the $6-10$ step range. We apply a loss to both stages of the FUS3D system, even if the detection stage is already pretrained, since the loss on the detection stage is the only mechanism that enforces an association between cells / tokens and a specific region in world space. Removal of $\mathcal{L}_{det}$ would therefore lead to instability during the training phase.

Training strategy We find that training of the FUS3D system requires a multi step approach to achieve full performance. Best results on the MIPT dataset [1] were obtained using a four-step approach. Adaptations may be required if the system is applied to other datasets:

1. Initially, the focus lies on object detection alone. We disregard the tracking stage and train the first stage as a standalone object detector on individual time steps using the loss function $\mathcal{L}_{det}^{t}$ given in Equation (3). Furthermore, we restrict training to a single "Human" class, no differentiation between the different pose classes "Standing", "Sitting" and "Lying" is made. The reasoning for this approach is a significant imbalance in the class distribution of the MIPT dataset with approximate fractions of $0.66, 0.24$ and 0.1 respectively.
2. Second, the pretrained object detection stage obtained from the first step is finetuned on the multi-class task. Using this approach, as opposed to directly training the multi-class problem from scratch, leads to an improvement in the mAP score of about 5 percentage points.
3. The third step unlocks the second stage. We freeze the weights of the now pretrained object detection stage and train the tracking stage until convergence. This step requires training over multiple time steps and thus the use of the full loss function given in Equation (4). Due to the frozen state of the first stage, the $\mathcal{L}_{det}^{t}$ term has no effect at this point. Training the two stages individually first has the further advantage that the learning rate schedule can be adapted to each stage. Furthermore, the transformer-based tracking stage takes about twice the number of epochs to reach full convergence.
4. Finally, we unfreeze the first stage and finetuning the entire FUS3D system end-to-end. This step gives another small improvement in all performance metrics.

Track augmentation There are three main tasks that any tracking algorithm must perform: Acquisition of new tracks, propagation of existing tracks, and removal of old tracks that are no longer valid. In terms of frequency of occurrence, propagation is the predominant task, while acquisition and removal are comparatively rare events. Our approach to dealing with this imbalance is a form of data augmentation applied to tracks, as proposed by TrackFormer [4]. Artificial false negative tracks are generated by dropping track queries passed from one time step to the next. We find this step to be critical to achieve higher recall on the acquisition on new tracks. In a similar manner, we can generate artificial false positive tracks. This is achieved by either falsely promoting object queries to track queries, or by keeping track queries that should have been dropped. We also experimented with randomly dropping entire frames in a sequence in order to simulate sensors with an unreliable frame rate or, equivalently, objects with large variation in movement speed. However, the latter was found to have no positive effect.

Hyperparameters and optimization Our model operates on images of size 320×320 pixels as input. Within the CNN backbone, these images undergo downsampling to form a tensor of dimensions $h = w = 10$ with $cd = 1056$ channels. In preparation for the subsequent tracking stage, this tensor is reshaped into a 4D representation, featuring a depth of $d = 24$ and $c = 44$ channels per individual cell. Through the application of a two-layer Multi-Layer Perceptron (MLP), each cell is expanded into tokens with dimensions $c' = 128$. These dimensions were determined through a comprehensive grid search, aiming to balance the enhancement of both detection and tracking metrics while ensuring an acceptable frame rate on the Jetson Nano target device.

Furthermore, an additional grid search was executed to find optimal thresholds for track addition and removal, which were determined as $\tau_{add} = 0.12$ and $\tau_{remove} = 0.58$, respectively.

For training of the FUS3D detection stage, we use the AdaBelief optimizer [7]. For our use-case this optimizer consistently demonstrates superior performance compared to alternatives like AdamW [3] or SGD. Our chosen configuration uses weight decay $wd = 10^{-4}$ and an initial learning rate of $lr = 10^{-3}$. Throughout the training process, the learning rate for the detection stage undergoes multiplicative decay in $\gamma = 0.1$ steps, gradually reaching a minimum value of 10^{-6} through a series of steps triggered upon encountering plateaus in the learning curve.

In the context of the tracking stage, we opt for the AdamW optimizer with a weight decay of $3 \cdot 10^{-3}$. The initial learning rate for this stage is also set at 10^{-3}, with the learning rate schedule following a cosine annealing pattern.

References

[1] Thomas Heitzinger and Martin Kampel. A foundation for 3d human behavior detection in privacy-sensitive domains. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, page 305. BMVA Press, 2021. 2

[2] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 1

[3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[4] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8844–8854, 2022. 2

[5] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020. 1

[6] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, pages 85–94. IEEE, 2019. 1

[7] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in neural information processing systems*, 33:18795–18806, 2020. 3