Bayesian Multiobject Tracking With Neural-Enhanced Motion and Measurement Models

Shaoxiu Wei, Mingchao Liang, and Florian Meyer

This manuscript provides derivations and further simulation results for the letter, "Bayesian Multiobject Tracking with Neural-Enhanced Motion and Measurement Models" by the same authors [1].

1 Loss Function and Training Procedure

The computation of the loss ℓ_k at time k used for the training of the proposed neural architecture can split up in a component related to the motion model, i.e., ℓ_k^{motion}, and a component related to the measurement model, ℓ_k^{meas}.

The motion-model related component of the loss is given by

$$\ell_k^{\text{motion}} = \frac{1}{I'_k} \sum_{i=1}^{I'_k} \left|\left|\hat{\mathbf{x}}_k^i - \mathbf{x}_{\text{gt},k}^i\right|\right|_1 \tag{1}$$

where we recall that $\hat{\mathbf{x}}_k^i$ is the MMSE estimate of the kinematic state of object i at time k and $\mathbf{x}_{\text{gt},k}^i$ is the corresponding groundtruth. For the evaluation of (1), we need to decide which estimated object state can be associated with a groundtruth state and only compute the loss for them. This decision is performed by applying the Hungarian algorithm [2]. Since only some kinematic object states are associated with ground truth states, we have that $I'_k \leq I_k$.

For the measurement model-related component of the loss, we use the "weighted" binary cross-entropy loss [3, Chapter 4.3]. In particular, the loss ℓ_k^{af} for the learning of affinity factors is given by

$$\ell_{1,k}^{\text{af}} = -\frac{\sum_{i=1}^{I_k}\sum_{j=1}^{J_k} f_{\text{gt},k}^{i,j} \ln\left(\sigma(\ln(f_{\text{af},k}^{i,j}))\right)}{\sum_{i=1}^{I_k}\sum_{j=1}^{J_k} f_{\text{gt},k}^{i,j}} \tag{2}$$

$$\ell_{2,k}^{\text{af}} = -\frac{\sum_{i=1}^{I_k}\sum_{j=1}^{J_k} (1-f_{\text{gt},k}^{i,j}) \ln\left(1-\sigma(\ln(f_{\text{af},k}^{i,j}))\right)}{\sum_{i=1}^{I_k}\sum_{j=1}^{J_k} (1-f_{\text{gt},k}^{i,j})} \tag{3}$$

$$\ell_k^{\text{af}} = \ell_{1,k}^{\text{af}} + \ell_{2,k}^{\text{af}} \tag{4}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $f_{\text{gt},k}^{i,j} \in \{0,1\}$ denotes the ground truth association of measurement j to PO i, i.e., if PO i is indeed matched with measurement j, we have $f_{\text{gt},k}^{i,j} = 1$, otherwise $f_{\text{gt},k}^{i,j} = 0$. The loss $\ell_{1,k}^{\text{af}}$ characterizes accuracy of existing associations, while $\ell_{2,k}^{\text{af}}$ characterizes accuracy of missing associations. The normalization terms in (2) and (3) address the data imbalance problem, i.e., the normalization makes sure that whether a PO i is associated with a measurement j equally contributes to the joint loss ℓ_k^{af}, regardless of the specific number of POs and measurements.

For false positive rejection coefficients $f_{\text{fpr},k}^{i,j}$, the corresponding loss function is given by

$$\ell_{1,k}^{\mathrm{fpr}} = \frac{\sum_{j=1}^{J_k} f_{\mathrm{gt},k}^{j} \ln(f_{\mathrm{fpr},k}^{j})}{\sum_{j=1}^{J_k} f_{\mathrm{gt},k}^{j}} \tag{5}$$

$$\ell_{2,k}^{\mathrm{fpr}} = -w_{\mathrm{fpr}} \frac{\sum_{j=1}^{J_k} (1 - f_{\mathrm{gt},k}^{j}) \ln(1 - f_{\mathrm{fpr},k}^{j})}{\sum_{j=1}^{J_k} (1 - f_{\mathrm{gt},k}^{j})} \tag{6}$$

$$\ell_{k}^{\mathrm{fpr}} = \ell_{1,k}^{\mathrm{fpr}} + \ell_{2,k}^{\mathrm{fpr}} \tag{7}$$

where $f_{\mathrm{gt},k}^{j} \in \{0, 1\}$ represents the ground truth label indicating for each measurement if it is a false positive or not, and $w_{\mathrm{fpr}} \geqslant 0$ is a parameter that makes it possible to prioritize the loss contribution $\ell_{1,k}^{\mathrm{fpr}}$ over $\ell_{2,k}^{\mathrm{fpr}}$ and vice versa. The loss contribution $\ell_{1,k}^{\mathrm{fpr}}$ represents the classification accuracy for true measurements and $\ell_{2,k}^{\mathrm{fpr}}$ represents the classification accuracy for false positives. The tuning parameter is typically $w_{\mathrm{fpr}} < 1$. This is motivated by the fact that missing an object is usually more harmful than generating a false positive. Finally, the measurement model-related component of the loss is obtained as $\ell_{k}^{\mathrm{meas.}} = \ell_{k}^{\mathrm{af}} + \ell_{k}^{\mathrm{fpr}}$.

Finally, the joint loss is given by

$$\ell = \ell_{k}^{\mathrm{motion}} + w_{\ell}^{\mathrm{meas.}} \ell_{k}^{\mathrm{meas.}} \tag{8}$$

where $w_{\mathrm{fpr}} \geqslant 0$ is another parameter that makes it possible to prioritize the loss contribution $\ell_{k}^{\mathrm{meas.}}$ over $\ell_{k}^{\mathrm{motion}}$ and vice versa. This joint loss function makes it possible to train the neural enhanced motion and measurement models jointly.

2 Feature Extraction

The object-oriented kinematic feature is given by $\underline{\mathbf{f}}_{\mathrm{ki},k}^{i} = [\underline{\mathbf{p}}_{k}^{i\mathrm{T}} \underline{\mathbf{v}}_{k}^{i\mathrm{T}} \underline{\mathbf{u}}_{k}^{i\mathrm{T}}]^{\mathrm{T}}$. Here, $\underline{\mathbf{p}}_{k}^{i}$ and $\underline{\mathbf{v}}_{k}^{i}$ are the 2-D position and velocity estimate extracted as the mean of the predicted posterior PDF $f(\mathbf{x}_{k}^{i} | \mathbf{z}_{1:k-1}, r_{k}^{i} = 1)$ and $\underline{\mathbf{u}}_{k}^{i}$ represents size and orientation information. In particular, we use bounding box information of the measurement used to initialize PO i as size information. In addition, orientation information is extracted from the last measurement associated with PO i. The measurement-oriented kinematic feature $\overline{\mathbf{f}}_{\mathrm{ki},k}^{j} = [\overline{\mathbf{p}}_{k}^{j\mathrm{T}} \overline{\mathbf{v}}_{k}^{j\mathrm{T}} \overline{\mathbf{u}}_{k}^{j\mathrm{T}}]^{\mathrm{T}}$ consists of a 2-D position and velocity measurement as well as size and feature information provided by the detector.

For object-oriented shape feature extraction, position, orientation, and size information determine the ROI $\underline{\mathcal{R}}_{1,k-1}^{i}$. The ROI is identified on the BEV feature map [4–6] in $\mathbb{R}^{180 \times 180 \times 128}$ corresponding to raw measurement from time $k-1$. Next, sample points from the ROI are determined using bilinear interpolation. In particular, we use edges or corners of the ROI. We apply two convolutional layers and a two-layer MLP to obtain the feature vector, i.e., $\mathrm{MLP}(\mathrm{Conv}(\underline{\mathcal{R}}_{1,k-1}^{i})) \in \mathbb{R}^{64}$. In addition, we perform an additional convolution over the entire BEV map and identify ROI $\underline{\mathcal{R}}_{2,k-1}^{i}$ on that resulting new BEV feature map in $\mathbb{R}^{180 \times 180 \times 64}$. By performing the same steps as above, we obtain a complementary feature vector $\mathrm{MLP}(\mathrm{Conv}(\underline{\mathcal{R}}_{2,k-1}^{i})) \in \mathbb{R}^{32}$. Finally, we concatenate these two feature vectors to obtain the final object-oriented shape feature, i.e.,

$$\underline{\mathbf{f}}_{\mathrm{sa},k}^{i} = \left[\mathrm{MLP}(\mathrm{Conv}(\underline{\mathcal{R}}_{1,k-1}^{i}))^{\mathrm{T}} \, \mathrm{MLP}(\mathrm{Conv}(\underline{\mathcal{R}}_{2,k-1}^{i}))^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{9}$$

The same feature-extraction procedure is performed to obtain measurement-oriented shape features $\overline{\mathbf{f}}_{\mathrm{sa},k}^{i}$ using the BEV feature map corresponding to raw measurements from time k.

Based on kinematic and shape features related to a PO i and measurement j pair, the affinity factor $f_{\mathrm{af},k}^{i,j}$ is obtained as

$$f_{\mathrm{af},k}^{i,j} = \exp(\mathrm{MLP}(\Delta \mathbf{f}^{i,j})) \tag{10}$$

where we introduced

$$\Delta\mathbf{f}^{i,j} = \left[\left(\underline{\mathbf{p}}_k^i - \overline{\mathbf{p}}_k^j \right)^{\mathrm{T}} \ \underline{\mathbf{v}}_k^{i\mathrm{T}} \ \underline{\mathbf{u}}_k^{i\mathrm{T}} \ \underline{\mathbf{f}}_{\mathrm{sa},k}^{i\mathrm{T}} \ \overline{\mathbf{v}}_k^{j\mathrm{T}} \ \overline{\mathbf{u}}_k^{j\mathrm{T}} \ \overline{\mathbf{f}}_{\mathrm{sa},k}^{j\mathrm{T}} \right]^{\mathrm{T}}.$$

Taking the difference of the position entries of the kinematic states removes the influence of the absolute values of position entries, which can vary strongly across scenes. The false positive rejection factor only relies on measurement-oriented features. In particular, for each measurement j, the false alarm rejection factor $f_{\mathrm{fpr},k}^j$ is obtained as

$$f_{\mathrm{fpr},k}^j = \sigma\Big[\mathrm{MLP}(\overline{\mathbf{v}}_k^j, \overline{\mathbf{u}}_k^j, \overline{\mathbf{f}}_{\mathrm{sa},k}^j)\Big]. \tag{11}$$

3 Positive Impact of Affinity Factors and False Positive Rejection Factors

The positive impact of affinity factors is visualized in Fig. 2. Here, false positive rejection factors are set to one to make sure their effects are omitted. In particular, predicted particles weighted by the functions $l_{\mathrm{ne}}(\mathbf{z}_k|\underline{\mathbf{y}}_k^i)$ and $l(\mathbf{z}_k|\underline{\mathbf{y}}_k^i)$ used for model-based and neural-enhanced measurement update are shown. Fig. 2(a) depicts a scenario with three moving cars. Particles and their corresponding value of $l_{\mathrm{ne}}(\mathbf{z}_k|\underline{\mathbf{y}}_k^i)$ and $l(\mathbf{z}_k|\underline{\mathbf{y}}_k^i)$ for the three moving cars are shown. It can be observed that, contrary to the model-based measurement update, the neural-enhanced measurement update assigns lower particle weights to particles near false-positive measurements. Fig. 2(b) illustrates another potential advantage of the neural-enhanced update step in a scenario with two pedestrians. In particular, it can be seen that the neural-enhanced update step makes it possible to more clearly distinguish between the two pedestrians by assigning smaller weights to particles located in the area between the two pedestrians. Fig. 1, shows the positive effect of false positive rejection factors. In particular, it can be seen that the neural-enhanced update step with false positive rejection factors can clearly reduce the number of false positive object state estimates.

References

[1] S. Wei, M. Liang, and F. Meyer, "Bayesian multiobject tracking with neural-enhanced motion and measurement models," 2025, submitted.

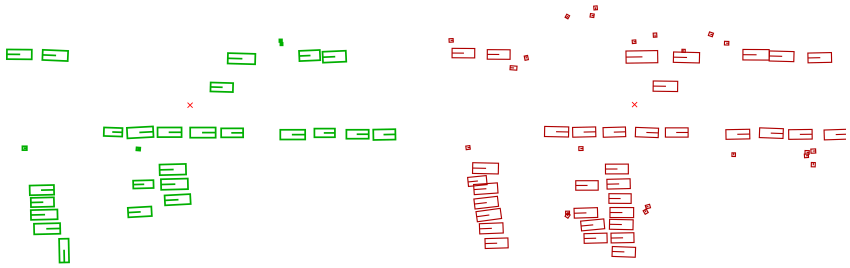
[2] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

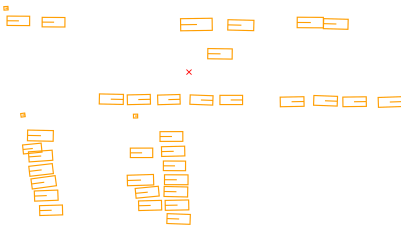
[4] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, Oct. 2018.

[5] Y. Chen, Z. Yu, Y. Chen, S. Lan, A. Anandkumar, J. Jia, and J. Alvarez, "FocalFormer3D : Focusing on hard instance for 3D object detection," in *Proc. ICCV-23*, Oct. 2023, pp. 8360–8371.

[6] T. Yin, X. Zhou, and P. KrÃ¤henbÃ¼hl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF CVPR-2021*, Nov. 2021, pp. 11 779–11 788.

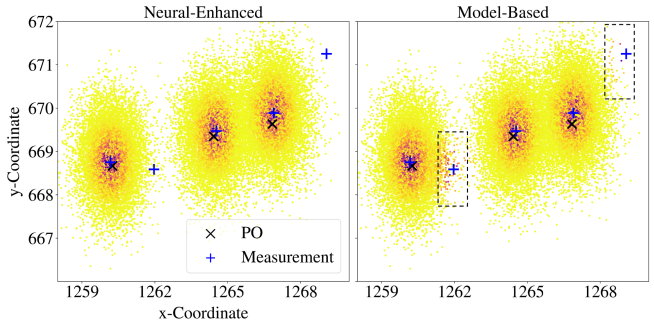


(a) Groundtruth (b) Estimation result with $f_{\text{af}}(\checkmark)$ and $f_{\text{fpr}}(\times)$

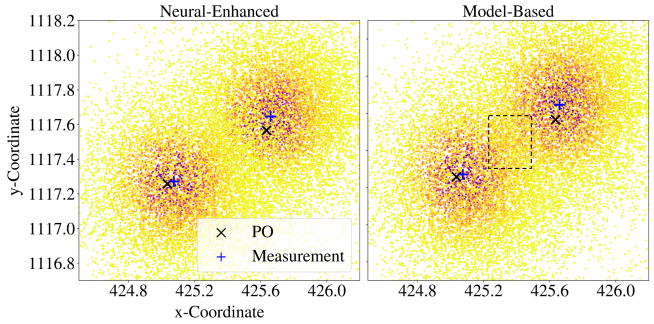


(c) Estimation result with $f_{\text{af}}(\checkmark)$ and $f_{\text{fpr}}(\checkmark)$

Figure 1: Tracking results for cars and pedestrians classes at one time step. The ground truth in (a) is compared with the proposed neural-enhanced update step. The neural enhanced update step is performed without false positive rejection (b) and with false positive rejection factors (c). Object declaration is performed by setting $T_{\text{dec}} = 0.7$. It can be seen that the result in (c) is closer to the result in (a) by having fewer false positives compared to (b).



(a) Likelihood for three cars with clutter



(b) Likelihood for two close pedestrians

Figure 2: Visualization of two examples that indicate how the affinity factors used by the neural-enhanced measurements update step can lead to improvements compared to a model-based measurement update step. Predicted particles and their corresponding weights provided by $l_{\mathrm{ne}}(\mathbf{z}_k|\underline{\mathbf{y}}_k^i)$ and $l(\mathbf{z}_k|\underline{\mathbf{y}}_k^i)$ are shown. A redder color indicates a higher particle weight. The black $\times$ denotes predicted position and the blue $+$ denotes measurements. For this visualization, the false positive rejection factor is set to one. The neural-enhanced measurements update step assigns smaller weights to false positive measurements in (a) and more clearly distinguish between the two pedestrians in (b).