SIRA: Scalable Inter-frame Relation and Association for Radar Perception

Supplementary Material

6. Related Work for Visual Tracking

In recent years, KF-based approaches have gained popularity in the context of visual tracking, and various extensions have been proposed [7, 8, 12, 33, 41, 52, 54, 61, 62], exemplified by SORT [4]. SORT can achieve high tracking performance, but it relies on the assumption that objects have consistent linear motion in a short time, which requires continuous observations. Therefore, it can face challenges when objects exhibit occlusion or nonlinear motion, requiring a high frame rate. To overcome occlusion problems, ByteTrack [62] uses the similarity between tracklets and low-scoring detection boxes to recover the true objects and filter out background detections. OC-SORT [8] introduces object motion computed from pre- and post-occlusion time pairs to address occlusion and non-linear motion. Our proposed framework extends recent KF-based methods and learning-based approaches by assuming high-density radar detection points. It explicitly considers strong object-level consistency by using multiple frames to capture the nonlinear motion of objects.

7. Related Work for Radar Datasets

If we categorize open radar datasets into the ones with sparse detection points, dense points and low-level heatmap, RADIATE[47] is the largest dense-point dataset with bounding box and tracking ID labels for both detection and tracking as shown in Table 4. We will use these datasets for further evaluation in the future.

Table 4. A list of open radar datasets in the format of dense points.

Dataset	# of data	Radar format	BBox	Tracking ID
RADIATE [47]	44K	dense points	2D	✓
Zendar [36]	4.8K	dense points	2D	✓
TJ4DRadSet [63]	7.7K	dense points	2D	✓
RADIal [45]	25K	heatmap+points	2D	

8. TempoRadar [27]

Our ETR generalizes TempoRadar [27] into a long time horizon and shares several key building blocks such as the top-K feature selector $\mathcal{S}_K$ and the design of the (temporal) masking matrix $\mathbf{M}$ in the masked multi-head attention (MCA).

Top-K Feature Selector $\mathcal{S}_K$: To exploit the feature-level temporal relation, TempoRadar introduces a temporal relation layer (TRL). Given the extracted features $\mathbf{Z}_t := \mathcal{F}_\theta\left(I_{t,t-1}\right)$ and $\mathbf{Z}_{t-1} := \mathcal{F}_\theta\left(I_{t-1,t}\right)$ from the encoder, where $I_{t-1,t}$ concatenates two consecutive radar frames I_{t-1} and I_t along the channel dimension in the order of $(t-1,t)$, the feature selector $\mathcal{S}_K$ of (3) is defined as:

$$\mathbf{H}_t = \mathcal{S}_K\left(\mathbf{Z}_t\right) := \mathbf{Z}_t\left[P_t^{\text{pre-hm}}\right],$$
$$\mathbf{H}_{t-1} = \mathcal{S}_K\left(\mathbf{Z}_{t-1}\right) := \mathbf{Z}_{t-1}\left[P_{t-1}^{\text{pre-hm}}\right],$$

where $\mathbf{H}_{t/t-1} \in \mathbb{R}^{C\times K}$ and $P_t^{\text{pre-hm}}$ is defined as the set of (x,y) coordinates corresponding to the K selected features,

$$P_t^{\text{pre-hm}} := \left\{(x,y) \;\middle|\; \{\mathbf{C}_t\}_{xy} \geq \{\mathbf{C}_t\}_K\right\}, \tag{18}$$

where $\mathbf{C}_t = \mathcal{G}_\theta^{\text{pre-hm}}\left(\mathbf{Z}_t\right)$ maps the channel dimension of the feature map via a learnable feedforward neural network (FNN) module $\mathcal{G}_\theta^{\text{pre-hm}} : \mathbb{R}^{C\times\frac{H}{s}\times\frac{W}{s}} \rightarrow \mathbb{R}^{1\times\frac{H}{s}\times\frac{W}{s}}$ into a scalar feature map for feature ranking, $\{\mathbf{C}_t\}_K$ stands for the K-th largest value in $\mathbf{C}_t$ over the spatial space $\frac{H}{s}\times\frac{W}{s}$, and the subscript xy takes value at the coordinate (x,y).

Design of $\mathbf{M}$ in Masked MCA: Let us stack the top-K selected features from the two consecutive radar frames as $\mathbf{H}_{t,t-1} := \{\mathbf{H}_t, \mathbf{H}_{t-1}\}^\top \in \mathbb{R}^{2K\times C}$. The masked MCA takes $\mathbf{H}_{t,t-1}$ and applies cross-frame attention over the two sets of features, as shown in Fig. 7a.

Since the position is lost in $\mathbf{H}_{t,t-1}$, we generate the position information of the selected top-K features via a learnable positional encoding network $\mathcal{E}_\theta$ from the coordinate set $P_t^{\text{pre-hm}}$ of (18)

$$\mathbf{P}_t^{\text{enc}} = \mathcal{E}_\theta\left(P_t^{\text{pre-hm}}\right) \in \mathbb{R}^{K\times D_{\text{pos}}},$$

where D_{pos} is the dimension of positional encoding. We then supplement the positional encoding into feature vectors

$$\mathbf{H}_{t,t-1}^{\text{pos}} = \left\{\mathbf{H}_{t,t-1}, \mathbf{P}_{t,t-1}^{\text{enc}}\right\} \in \mathbb{R}^{2K\times(C+D_{\text{pos}})},$$

where $\mathbf{P}_{t,t-1}^{\text{enc}} = \left\{\mathbf{P}_t^{\text{enc}}, \mathbf{P}_{t-1}^{\text{enc}}\right\}^\top \in \mathbb{R}^{2K\times D_{\text{pos}}}$, and pass it to the masked MCA for temporal attention.

In computing the temporal relation, the masked MCA follows [2, 20, 21, 43] and uses a temporal inductive bias with a masking matrix $\mathbf{M}$

$$\mathcal{A}\left(\mathbf{V},\mathbf{X}\right) := \text{softmax}\left(\frac{\mathbf{M} + q\left(\mathbf{X}\right)k\left(\mathbf{X}\right)^\top}{\sqrt{d}}\right)v\left(\mathbf{V}\right),$$

where $q\left(\cdot\right)$, $k\left(\cdot\right)$ and $v\left(\cdot\right)$ are linear transformation layers and are referred to as query, keys and values, respectively, and d is the dimension of the query and the

keys. For the temporal attention over $\{t, t-1\}$, we have $\mathcal{A}\{\mathbf{H}_{t,t-1}, \mathbf{H}^{\text{pos}}_{t,t-1}\}$ with $\mathbf{V} = \mathbf{H}_{t,t-1}$ for the value and $\mathbf{X} = \mathbf{H}^{\text{pos}}_{t,t-1}$ for the key and query. The masking matrix $\mathbf{M}$ is given as

$$\mathbf{M} := \begin{bmatrix} \mathbb{I}_K, \mathbf{1}_K \\ \mathbf{1}_K, \mathbb{I}_K \end{bmatrix} + \sigma \left(\begin{bmatrix} \mathbf{1}_K, \mathbf{0}_K \\ \mathbf{0}_K, \mathbf{1}_K \end{bmatrix} - \mathbb{I}_{2K} \right), \quad (19)$$

where $\mathbb{I}_K$ is the identity matrix of size K, $\mathbf{1}_K$ and $\mathbf{0}_K$ are the all-one and all-zero matrix with size $K \times K$, respectively, and σ is a large negative constant, e.g., -10^{10}, to guarantee a near-zero value in the output through the softmax function. It can be shown that diagonal blocks in $\mathbf{M}$ disable attention between features within the same frame, while off-diagonal blocks allow for cross-frame attention. The masked MCA may repeat multiple times.

9. Details of BBox Loss

We pick the object's center coordinates from the heatmap, and learn its attributes from feature representations through regression. Regression functions, which are heatmap loss $\mathcal{L}^{\text{h}}_t$, width & Length loss $\mathcal{L}^{\text{b}}_t$, orientation loss $\mathcal{L}^{\text{r}}_t$, and offset loss $\mathcal{L}^{\text{o}}_t$, compose the training objective by a linear combination as (15):

$$\mathcal{L}^{\text{BBox}}_t = \frac{1}{N_{\text{gt}}} \sum_{k=1}^{N_{\text{gt}}} \left(\mathcal{L}^{\text{b}}_{t,k} + \mathcal{L}^{\text{r}}_{t,k} + \mathcal{L}^{\text{o}}_{t,k} \right) - \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}^{\text{h}}_{t,i},$$

where N denotes the total number of pixels in the heatmap and N_{gt} is the total number of ground truth bounding boxes. Each loss is as follows:

$$\begin{aligned} \mathcal{L}^{\text{h}}_{t,i} =& \mathbb{1}_{\{c_{t,i}=1\}} \left(1 - \widehat{c}_{t,i}\right)^{\alpha} \log\left(\widehat{c}_{t,i}\right) \\ &+ \mathbb{1}_{\{c_{t,i}\neq 1\}} \left(1 - c_{t,i}\right)^{\beta} \widehat{c}^{\alpha}_{t,i} \log\left(1 - \widehat{c}_{t,i}\right), \end{aligned} \quad (20)$$

where $c_{t,i}$ and $\widehat{c}_{t,i}$ denote the ground-truth and predicted value at i-th coordinate in $\mathcal{G}^{\text{hm}}_{\theta}\left(\mathbf{Z}^{\text{hm}}_t\right)$, and α and β are hyper-parameters and are chosen empirically with 2 and 4, respectively.

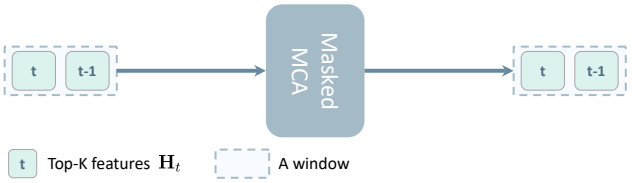
$$\mathcal{L}^{\text{b}}_{t,k} = S_{L_1}\left(\left\| \mathcal{G}^{b}_{\theta}\left(\mathbf{Z}_t\left[P^{\text{gt}}_{t,k} \right] \right) - \left(w_{t,k}, h_{t,k}\right)^{\top} \right\| \right), \quad (21)$$

$$\mathcal{L}^{\text{r}}_{t,k} = S_{L_1}\left(\left\| \mathcal{G}^{r}_{\theta}\left(\mathbf{Z}_t\left[P^{\text{gt}}_{t,k} \right] \right) - \left(\cos\vartheta_{t,k}, \sin\vartheta_{t,k}\right)^{\top} \right\| \right), \quad (22)$$

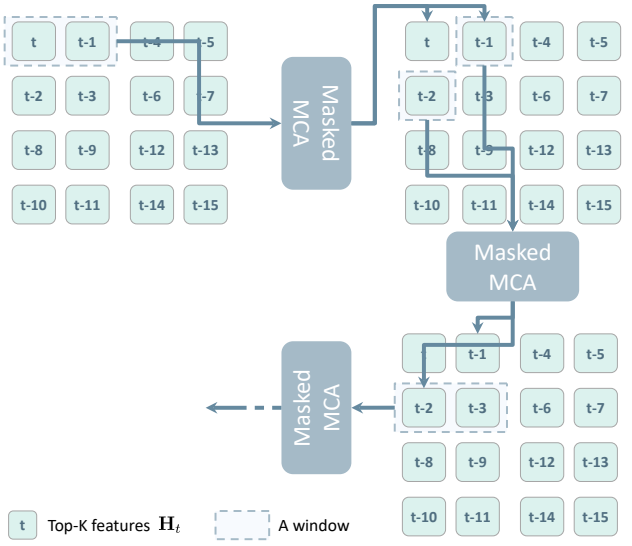
$$\mathcal{L}^{\text{o}}_{t,k} = S_{L_1}\left(\left\| \mathcal{G}^{o}_{\theta}\left(\mathbf{Z}_t\left[P^{\text{gt}}_{t,k} \right] \right) - \left(o_{x,t,k}, o_{y,t,k}\right)^{\top} \right\| \right), \quad (23)$$

where $P^{\text{gt}}_{t,k}$ denotes the coordinate $(x_{t,k}, y_{t,k})$ of the center of k-th ground truth object, $(w_{t,k}, h_{t,k})$ is the width & length, and $(o_{x,t,k}, o_{y,t,k})$ is the offset as follows:

$$\left(o_{x,t,k}, o_{y,t,k}\right) = \left(\frac{x_{t,k}}{s} - \left\lfloor \frac{x_{t,k}}{s} \right\rfloor, \frac{y_{t,k}}{s} - \left\lfloor \frac{y_{t,k}}{s} \right\rfloor \right). \quad (24)$$



(a) Masked MCA of TempoRadar.



(b) Masked MCA of SeTR.

Figure 7. Masked MCA. (a) TempoRadar [27] computes masked multi-head cross-attention (MCA) over the top-K selected features from a time horizon of only $T = 2$ consecutive radar frames. (b) SeTR computes masked MCA for two consecutive radar frames at a time, the same as the TempoRadar in (a), but slides the window of two frames after each MCA sequentially to cover a longer time horizon of $T > 2$ frames.

10. Sequential TempoRadar (SeTR)

One might postulate: "What are the implications of extending *TempoRadar* to cover more consecutive radar frames?" The answer might be two-fold. On the one hand, one should expect improved performance under the assumption that most radar features are present over more than just 2 frames, considering a typical radar frame rate of > 4 fps (*Radiate* dataset has 4 fps [47]). On the other hand, directly applying temporal attention to a longer time horizon incurs a quadratic computation complexity (refer to (8)) over the number of features from each frame K and the number of frames T.

One straightforward way for a scalable TempoRadar is to stack temporal feature attention for two consecutive frames and sequentially connect them, which we refer to as *sequential TempoRadar* (SeTR). As illustrated in Fig. 7b, SeTR computes masked MCA for two consecutive radar frames at a time, the same as the TempoRadar in Fig. 7a, but slides

the window of two frames after each MCA sequentially to cover a longer time horizon of $T > 2$ frames.

11. Training and Inference Pipelines for SIRA

Training Pipeline for SIRA: To train SIRA, we takes T consecutive radar frames, pass them into the training pipeline in the Top diagram of Fig. 8, and compute the loss function $\mathcal{L}^{\text{BBox}}$ of (15) at the decoder output for detection loss and the pseudo-direction loss $\mathcal{L}^{\text{DEst}}$ of (16) at the output of the DEst module (detailed in **Motion Consistency for Training** of Section 3.3). Through backpropagation, the learnable modules, hatched in light green, are updated using the derived loss value.

Inference Pipeline for SIRA: In the bottom of Fig. 8, we show the inference pipeline for SIRA. Noticeably, a tracker is attached to the DEst module to further enforce the motion consistency via the concept of pseudo-tracklet, detailed in **Motion Consistency for Inference** of Section 3.3. All learnable parameters during training are frozen in the inference. We further include the pseudo-code of the inference pipeline in Algorithm 1. A typical tracker consists of five steps: Prediction, Association, Update, Deletion, and Initialization. We can integrate our MCTrack with standard trackers (e.g., OC-SORT) by incorporating the key components (highlighted in **green** in Algorithm 1), e.g., the use of motion similarity of (11), to the Association step.

Extension with higher-order KF: As shown in Fig. 9, we expect SIRA can deal with nonlinear motion to an extent as the average predicted angle $\widehat{\phi}_{\text{ave}}$ can correct the KF predicted state $\widehat{\mathbf{x}}_{T|T-1}$ closer to the right observation $\mathbf{z}_T$. SIRA can be extended with higher-order KF (e.g., extended/unscented KF) to further improve the predicted state $\widehat{\mathbf{x}}_{T|T-1}$ with a proper nonlinear model and the average predicted angle $\widehat{\phi}_{\text{ave}}$, yielding improved trajectory predictions.

The choice of patch size: Since the patch is a subset of Top-K features in each frame, we have the patch size $M \in [1, K]$. Given the number of frames U in each window, the smaller the patch size M, the smaller the window size UM in the re-grouping stage of the TRWA block (see the top right of Fig. 3 for an example of $U = 4$ frames and $M = K/2$), and the lower the computational complexity of the window-based attention which is quadratic with respect to UM. On the other hand, a small M may limit the number of features to be correlated across windows and reduces the connectivity of temporal attention.

12. Details of Experimental Settings

Algorithm 1: Pseudo-code of SIRA for Inference.

Input: A radar frame sequence `V`; encoder `Enc`; decoder `Dec`; object detector `ETR`; direction estimator `DEst`; detection score threshold γ; birth threshold β
Output: Tracks $\mathcal{T}$ of the video

Initialization: $\mathcal{T} \leftarrow \emptyset$
for *frame* f_k *in* `V` **do**
  `/* Fig.2, and Fig.8 */`
  `/* **predict bboxes with ETR** */`
  $\mathcal{F}_k \leftarrow$ `Enc`(f_k)
  $\mathcal{F}_k \leftarrow$ `ETR`$(\mathcal{F}_k)$
  $\mathcal{D}_k \leftarrow$ `Dec`$(\mathcal{F}_k)$
  `/* **tracking with MCTrack** */`
  $\mathcal{J}_k \leftarrow$ `DEst`$(\mathcal{F}_k)$
  $\mathcal{D}_{high} \leftarrow \emptyset$
  $\mathcal{J}_{high} \leftarrow \emptyset$
  for d, j *in* $\mathcal{D}_k, \mathcal{J}_k$ **do**
    if $d.score > \gamma$ **then**
      $\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}$
      $\mathcal{J}_{high} \leftarrow \mathcal{J}_{high} \cup \{j\}$
    end
  end
  `/* predict new locations of tracks */`
  for t *in* $\mathcal{T}$ **do**
    $t \leftarrow$ `KalmanFilter.predict`(t)
  end
  `/* Fig.5 */`
  `/* association */`
  Associate $\mathcal{T}$ and $\mathcal{D}_{high}$&$\mathcal{J}_{high}$ with `Similarity` Eq.11
  $\mathcal{D}_{remain} \leftarrow$ remaining unmatched object from $\mathcal{D}_{high}$
  $\mathcal{T}_{remain} \leftarrow$ remaining matched tracks from $\mathcal{T}$
  `/* update status of matched tracks */`
  for t *in* $\mathcal{T}_{remain}$ **do**
    $t \leftarrow$ `KalmanFilter.update`(t)
  end
  `/* delete unmatched tracks */`
  $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_{remain}$
  `/* initialize new tracks */`
  for d *in* $\mathcal{D}_{remain}$ **do**
    if $d.score > \beta$ **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$
    end
  end
end
Return: $\mathcal{T}$

In **green** is the key of our method.

Dataset To facilitate the research on robust and reliable vehicle perception, *Radiate* dataset was collected in 7 scenarios under various weather and lighting conditions: Sunny (Parked), Sunny/Overcast (Urban), Overcast (Motorway), Night (Motorway), Rain (Suburban), Fog (Suburban) and Snow (Suburban). It includes multiple sensor modalities from radar and optical images to 3D LiDAR point clouds and GPS. 8 object classes, i.e., car, van, truck, bus, motorbike, bicycle, pedestrian and group of pedestrian, were annotated on the radar frames. The data for-

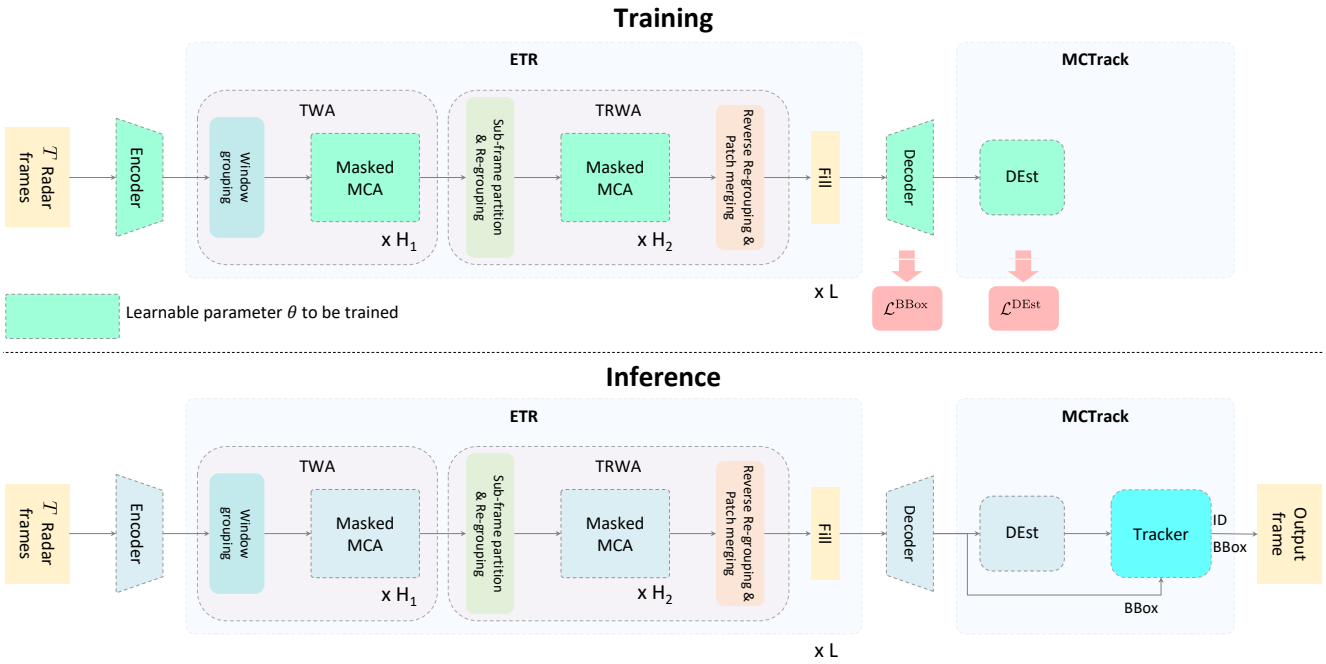


Figure 8. Training and Inference Pipelines for SIRA. (Top) SIRA takes T consecutive radar frames into the training pipeline, and computes $\mathcal{L}^{\text{BBox}}$ at the decoder and the pseudo-direction loss $\mathcal{L}^{\text{DEst}}$ at the DEst module in **Motion Consistency for Training** of Section 3.3. Learnable modules are hatched in light green. (Bottom) SIRA attaches a tracker at the DEst module output to further enforce **Motion Consistency for Inference** of Section 3.3. The tracker incorporates the motion similarity of (11) for association. Learnable parameters are frozen during inference.

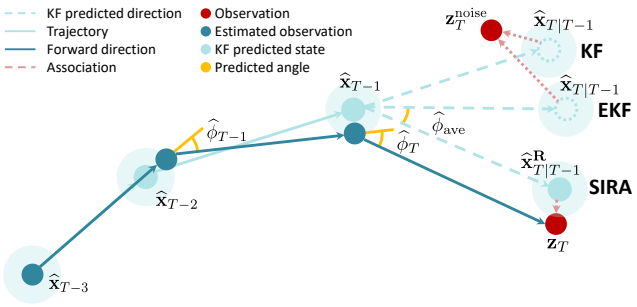


Figure 9. Trajectory prediction of SIRA with KF and EKF.

mat of radar frames generated from dense point clouds, where the pixel values indicate radar reflection magnitude. Radiate adopted the Navtech CTS350-X FMCW radar, a scanning radar that provides 360° high-resolution range-azimuth BEV images at 4 Hz. It was set to have 100-meter maximum operating range with a distance resolution of 0.175 m, an azimuth resolution of 1.8° and an elevation resolution of 1.8°. It does not provide Doppler information. Radar frames in Cartesian are provided as .png at 1152×1152 resolution. Nearest neighbour interpolation was used to convert the radar framess from the polar coordinate to the Cartesian one. Each pixel in the Cartesian coordinate represents a grid of $0.17361 \times 0.17361\text{m}^2$. In other words, the field of view is about $[-100\text{m}, 100\text{m}]$ in one axis and $[-100\text{m}, 100\text{m}]$ in the other axis in BEV. Radiate dataset has official 3 splits: "train in good weather" which consists of 31 sequences (22383 frames, only in good weather, sunny or overcast), "train good & bad weather" which consists of 12 sequences (9749 frames, both good & bad weather conditions), and "test" which consists of 18 sequences (11305 frames, all kinds of weather conditions). Fig. 10 shows sampled RGB and corresponding radar frames under adverse weather and low lighting conditions. We separately train models on the former two training sets and evaluate on the test set.

Hyper-parameters The hyper-parameters used in our experiments of Section 4 are shown in Table 5. The table is divided into three parts, Data, Architecture, and Training, each with parameter names, notations, and values.

13. Comparison of Complexity Analysis

Fig. 11 compares the computational complexity of TempoRadar in (8) and ETR in (9) as a function of the number of consecutive radar frames T, under two settings of the number of selected features $K = 8$ and $K = 16$ (grouped in

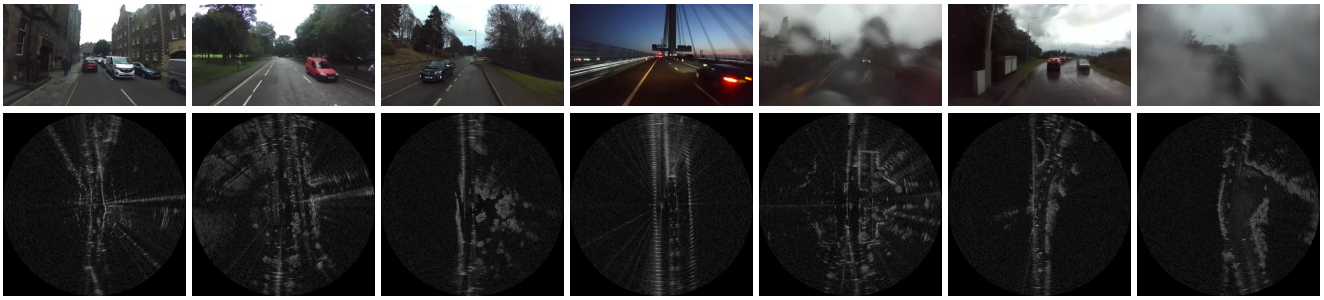


Figure 10. Visualization of RGB and corresponding radar frames. From left to right, the scenes are from City-3-7, City-7-0, Junction-1-10, Night-1-4, Fog-6-0, Rain-4-0 and Snow-1-0 in *Radiate*. Albeit of more coarse-grained and less semantic features, radar frames are much more resilient than RGB frames in adverse weather and low lighting conditions.

Table 5. Hyper-parameters used in our experiments.

	Name	Notation	Value
Data	dataset	-	*Radiate*
	train good weather	-	22383
	train good & bad weather	-	9749
	test	-	11305
	cropped image size	$H \times W$	256×256
	full image size	$H \times W$	1152×1152
Architecture	position dimention	D_{pos}	64
	downsampling ratio	s	4
	# of top-Ks	K	8
	# of sets of top-K	U	2
	# of ETR stages	L	1
	# of masked MCAs: TWA	H_1	2
	# of masked MCAs: TRWA	H_2	2
	operation	$\mathcal{M}$	max
	coefficient	λ	0.5
	detection score threshold	γ	0.08
	birth threshold	β	0.20
Training	batch size	-	16
	epoch	-	10
	optimizer	-	Adam
	learning rate	-	5e-4
	schedule for train good weather $\times 0.1$	-	5
	schedule for train good & bad weather $\times 0.1$	-	2
	weight decay for detection	-	1e-2
	weight decay for tracking	-	1e-5
	# of GPUs	-	1

two different colors). For each setting of K, we further include four ETR variants (denoted by different markers) with different combinations of hyper-parameters of the number of consecutive radar frames within one temporal window U and the number of features within one patch M. Under both settings, ETR provides more affordable temporal attention over longer time horizons T than TempoRadar.

While (9) represents the complexity of the general ETR module, [56] presents a special case of ETR with $U = 2$, $M = K/2$ and a stride $K/4$ (a special sub-frame partition with 50% overlapping). For this special case, the computational complexity is shown to be $2K^2\,(3T-4)\,L$.

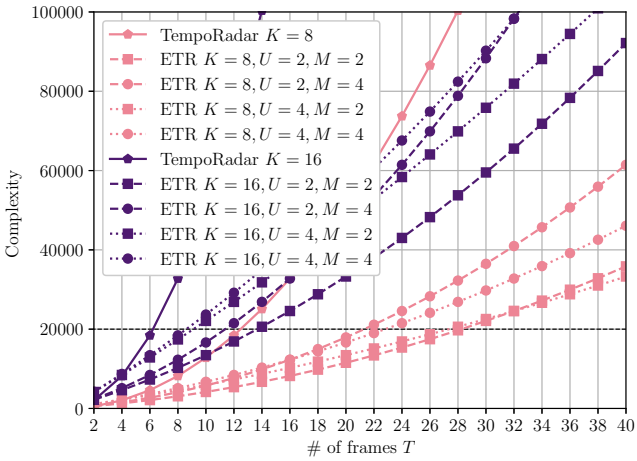


Figure 11. Comparison of computational complexities of TempoRadar (solid) and ETR (dashed) as a function of the number of frames T (along the x-axis) and the number of selected features K (grouped in different colors).

14. Definition of MOT Metrics

We adopt the series of MOT metrics [32, 35] for evaluation. We pick several key metrics in the experiments: MOTA (Multiple Object Tracking Accuracy), IDF1, ID switch (IDs), track fragmentations (Frag.), mostly tracked (MT), and partially tracked (PT). The MOTA score is calculated by

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t},$$

where t is the frame index, GT is the number of ground-truth objects, and FN and FP refer to false negative and false positive detection, respectively. The value of MOTA is in the range $(-\infty, 100]$. It can be deemed as the combination of detection and tracking performance, and is widely used as the main metric for accessing multiple object tracking quality.

Table 6. Additional results of object detection on *Radiate* for mAP@0.7. The number following the model name indicates the # of layers in the ResNet, and the number in parentheses indicates the # of frames T.

mAP@0.7	**Train good weather**	**Train good & bad weather**
RetinaNet-18 (1)	8.46±0.61	6.97±1.24
CenterPoint-18 (1)	19.02±1.80	14.43±2.56
BBAVectors-18 (1)	19.72±1.10	15.07±1.76
TR-18 (2)	20.57±1.47	15.59±2.31
TR-18 (4)	19.59±0.78	19.62±1.33
SeTR-18 (4)	21.90±1.12	19.65±0.84
SIRA-18 (4)	**21.95**±1.72	**19.66**±1.87
RetinaNet-34 (1)	7.67±1.71	6.93±1.60
CenterPoint-34 (1)	18.93±1.46	13.43±1.92
BBAVectors-34 (1)	19.86±1.36	14.67±1.45
TR-34 (2)	21.08±1.66	14.35±2.15
TR-34 (4)	22.46±1.76	19.03±1.10
SeTR-34 (4)	21.68±1.24	19.63±1.29
SIRA-34 (4)	**22.81**±0.86	**19.85**±0.95

Table 7. Comparison on object detection with full size images. Comparison on object detection with full size images.

Train good weather	mAP@0.3	mAP@0.5
FasterRCNN-50 (1) [47]	-	45.31
FasterRCNN-101 (1) [47]	-	45.84
TR-18 (2) [27]	-	48.02
TR-34 (2) [27]	-	48.66
ETR-34 (4)	65.10	49.19
SIRA-34 (4)	**65.67**	**51.49**
ETR-34 (6)	67.19	49.37
SIRA-34 (6)	**67.72**	**52.14**
ETR-34 (8)	65.53	50.59
SIRA-34 (8)	**67.82**	**52.55**
ETR-34 (10)	64.24	50.12
SIRA-34 (10)	**66.03**	**50.77**

IDF1 evaluates the identity preservation ability and focuses on the association performance. Specifically, IDF1 calculates a bijective (one-to-one) mapping between the sets of ground truth trajectories and predicted trajectories (unlike MOTA at the detection level) and is a function of

- IDTPs (identity true positives): the matches in the overlapping sections of trajectories that are correctly associated with the same identity;
- IDFNs (identity false negatives): instances where the ground truth has an identity that the prediction fails to identify. This often occurs in non-overlapping sections of matched trajectories or when the tracker loses track of an object;
- IDFPs (identity false positives): instances where the prediction assigns an identity that does not exist in the ground truth. This often happens in the case of oversegmentation or incorrect identity assignments;

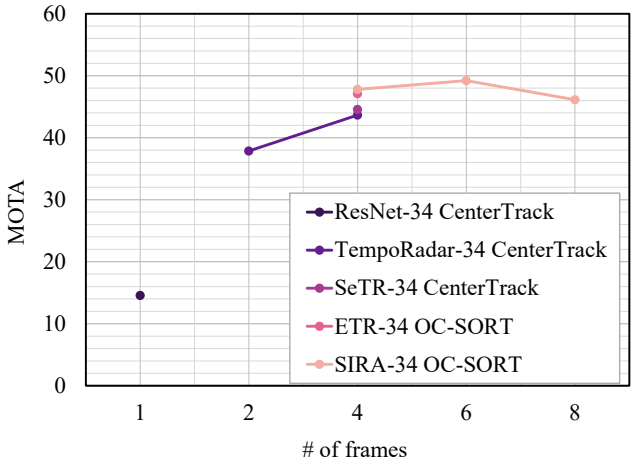


Figure 12. Tracking performance as a function of number of frames T. Compared with the single-frame baseline (ResNet-34 CenterTrack), SIRA with $T = 6$ consecutive frames results in a margin of $+34.67$ MOTA.

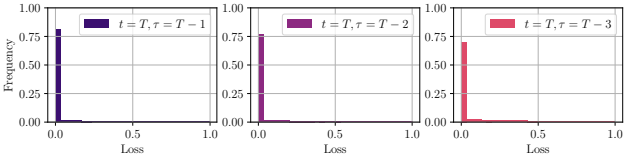


Figure 13. Pseudo-direction smooth L_1 loss in different time steps.

$$\text{IDF1} = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFP} + \text{IDFN}}. \tag{25}$$

The rest of these metrics all reflect the quality of predicted tracklets. For detailed definitions and calculations of MOT metrics, please refer to [35].

15. Additional Ablation Study

We present supplementary experimental results from our ablation studies. Each experimental setting aligns with the conditions detailed in Section 4.

Detection Results at mAP@0.7: For Table 1 in Section 4.2, additional results for mAP@0.7 are shown in Table 6. Compared to mAP@0.3 and mAP@0.5, all mAP@0.7 values are lower as expected when the IoU threshold increases. It is seen that SIRA provides consistently better detection performance than the baseline methods.

Detection Results With Full Size Radar Frames: We keep the original resolution with full size 1152×1152 to make a fair comparison to the results from [47]. Regarding variations in image size, a marginal decline in detection

Table 8. Experimental results of multiple object tracking on *Radiate*. The number following the model name indicates the # of layers in the Resnet backbone, and the number in parentheses indicates the # of frames T.

Train good weather	MOTA↑	MOTP↑	IDF1↑	IDs↓	FP↓	FN↓	Frag.↓	MT↑	ML↓	PT↑
ResNet-18 (1) CenterTrack	13.01	70.26	-	873	-	-	920	269	-	254
ResNet-34 (1) CenterTrack	14.55	70.05	-	802	-	-	831	282	-	279
TR-18 (2) CenterTrack	33.59	73.49	-	349	-	-	498	145	-	330
TR-34 (2) CenterTrack	37.85	71.85	39.90	457	970	6114	511	108	422	246
TR-18 (4) CenterTrack	42.77	70.38	44.91	519	1061	5206	520	244	196	336
TR-34 (4) CenterTrack	43.64	71.58	44.17	503	854	5892	538	197	253	326
SeTR-18 (4) CenterTrack	42.11	68.71	50.33	658	1481	4672	561	261	198	317
SeTR-34 (4) CenterTrack	44.57	71.65	48.72	875	1511	4606	602	348	129	299
ETR-34 (4) CenterTrack	46.06	70.23	50.81	1832	1141	4904	613	345	126	305
ETR-34 (4) OC-SORT	47.11	70.08	50.04	540	1411	4523	481	343	120	313
SIRA-34 (4) CenterTrack*	47.30	70.19	50.16	1249	1218	4756	566	354	122	300
SIRA-34 (4) OC-SORT	47.79	70.09	51.13	523	1408	4513	488	342	120	314

* For CenterTrack, C^{tracklet} is only used for association since this tracker is not based on SORT.

Table 9. Ablation study of various number of frames for multiple object tracking on train good weather. We used SIRA-34 OC-SORT.

# of frames T	MOTA↑	MOTP↑	IDF1↑	IDs↓	FP↓	FN↓	Frag.↓	MT↑	ML↓	PT↑
4	47.79	70.09	51.13	523	1408	**4513**	488	**342**	**120**	314
6	**49.22**	**71.70**	**51.87**	**399**	**1032**	4692	**306**	255	172	**349**
8	46.12	69.55	50.21	487	1076	4746	449	312	139	325

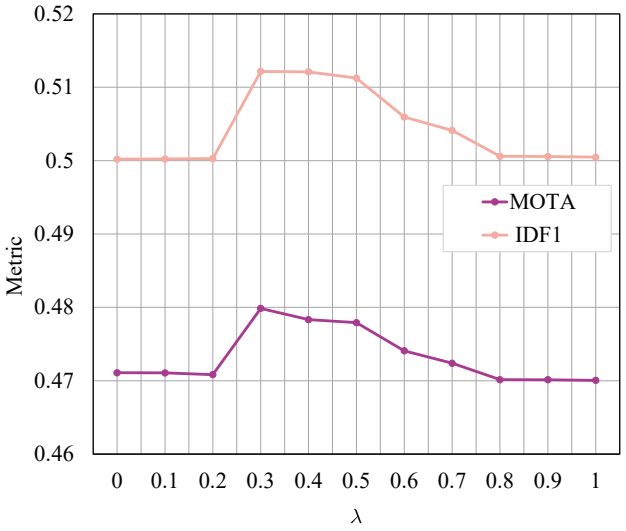


Figure 14. Performance variation due to different parameter λ.

performance is observed when dealing with larger scopes from Table 7. However, empirical evidence has shown that the utilization of SIRA consistently leads to superior performance compared to the TemporRadar (TR).

Tracking Results with Complete MOT Metrics: In Section 4.2, we showed tracking results in Table 2 with selected metrics. Here, we show the tracking results with complete metrics including MOTP, FP, FN and ML [35]. The tracking results are shown in Table 8 for comparison of the trackers. ID switches vary based on # of predicted BBox. With increased false negatives (FNs), both # of BBoxes and ID switches reduce. Table 8 suggests that TR generates more FNs and SIRA fewer FNs, thus higher ID switches. Nevertheless, we highlight the high IDF1 score in Table 2 and Table 8 of SIRA.

Number of Frames on Tracking: According to Table 9, considering longer time horizon contributes to the improvement in tracking performance in metrics such as MOTA and IDF1. These results clarify the significance of extending to longer time horizon while maintaining computational scalability. Fig. 12 illustrates the benefits of integrating more radar frames for the tracking performance over a range of methods. Compared with the single-frame baseline (ResNet-34 CenterTrack), SIRA with $T = 6$ consecutive frames results in a margin of $+34.67$ MOTA.

Effect of λ in (11): Fig. 14 illustrates the results obtained by varying λ in (11) in the main paper, which corresponds to C^{angle} when $\lambda = 1$ and C^{tracklet} when $\lambda = 0$. Fig. 14 appears to suggest that a combination of C^{angle} and C^{tracklet}, i.e., $\lambda \in [0.3, 0.7]$, consistently improves the tracking performance.

Performance of the Pseudo-Direction Estimation: We evaluated the pseudo-direction estimation performance in the terms of the smooth L_1 loss in (17) over the test dataset. Fig. 13 shows the loss histogram for three time steps $\tau = T - 1/T - 2/T - 3$ and it confirms that the majority of estimation errors are close to 0, indicating a high accuracy.

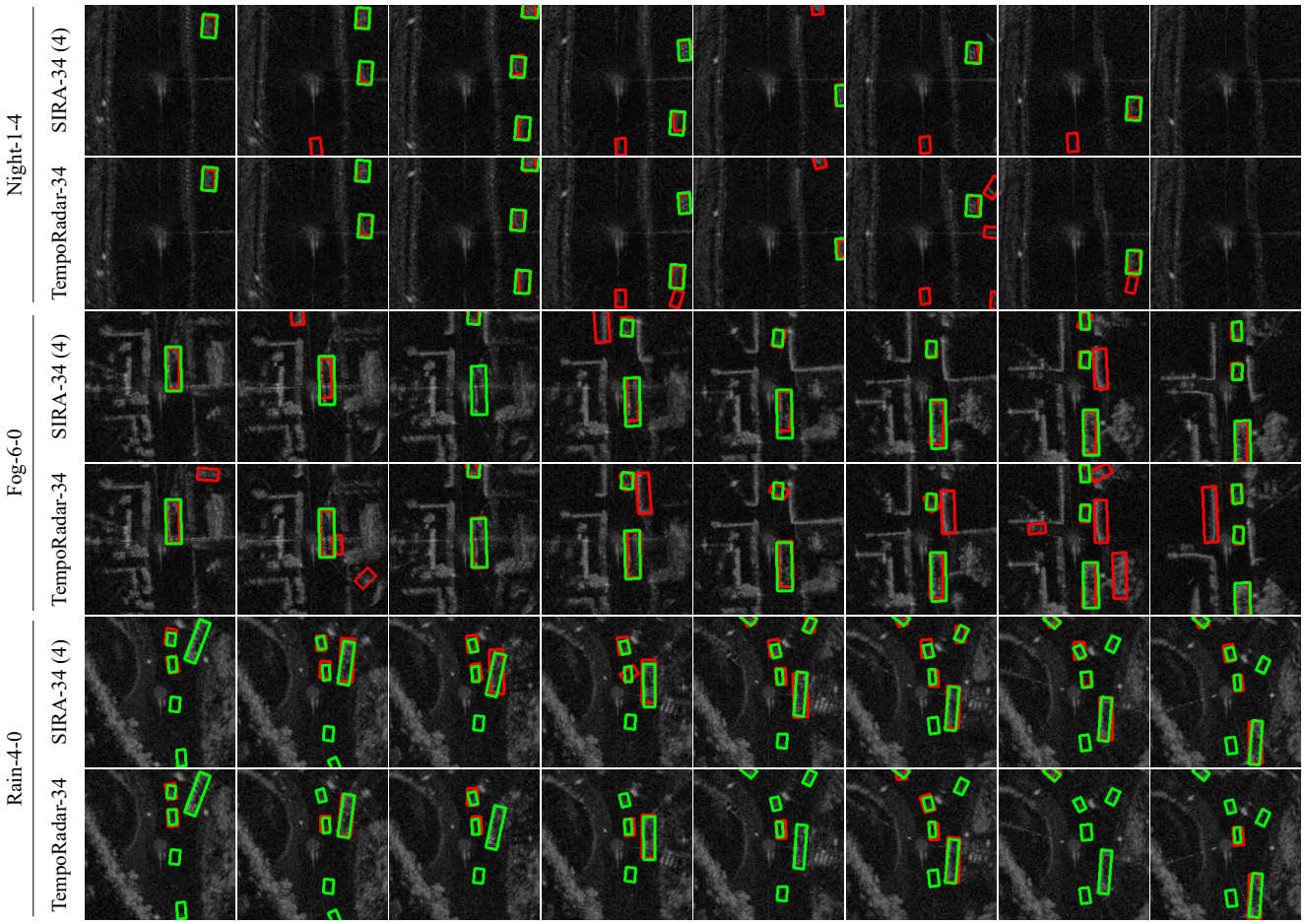


Figure 15. **Sampled detection results with cropped radar frames** on three scenarios: Night-1-4 (Top 2 Rows), Fog-6-0 (Middle 2 Rows) and Snow-4-0 (Bottom 2 Rows) on *Radiate*. For each scenario, we compare the SIRA and TempoRadar. Green boxes represent ground truth and red ones are predictions. The column represents consecutive radar frames. TempoRadar shows more false positives (FNs as unpaired red boxes) than SIRA, particularly in the first two scenarios.

16. Visualization Results

Detection with Cropped Radar Frames: Fig. 15 visualizes the detection results of Table 1 in Section 4.2 in adverse weather conditions: Night-1-4, Fog-6-0, and Rain-4-0, where green boxes represent the ground truth and red ones are the predictions. In this case, with $T = 4$ consecutive radar frames, SIRA allows for less FNs (unpaired green boxes) and less FPs (unpaired red boxes) in the BBox prediction than TempoRadar.

Detection with Full Size Radar Frames: Sampled detection results of Table 7 are visualized in Fig. 16. SIRA demonstrates robust performance even when applied to full size radar frames. However, compared to the cropped frames, there is a slight decline in performance with full size radar frames. Upon closer investigation of this phenomenon, it is observed that object shapes in regions distant from the radar appear blurred due to lower angular resolution, leading to a slight increase in both FPs and FNs. Furthermore, this blurring increases the difficulty of predicting angles, resulting in a lower IoU. FP predictions are also attributed to ghost objects present in the radar signal, as pointed out by Li et al. [27].

Tracking: Fig. 17 is in good weather, and Fig. 18 and Fig. 19 are in bad weather. From Fig. 17, TempoRadar faces by numerous FNs and frequent ID switches. In contrast, SIRA, leveraging longer temporal information for consideration of spatio-temporal consistency, exhibits fewer FNs and a reduced ID switches. As a result, SIRA consistently achieves stable tracking. Moreover, Fig. 18 illustrates that SIRA can detect and track objects even in adverse weather conditions. Particularly in the Rain-4-0 environment, where

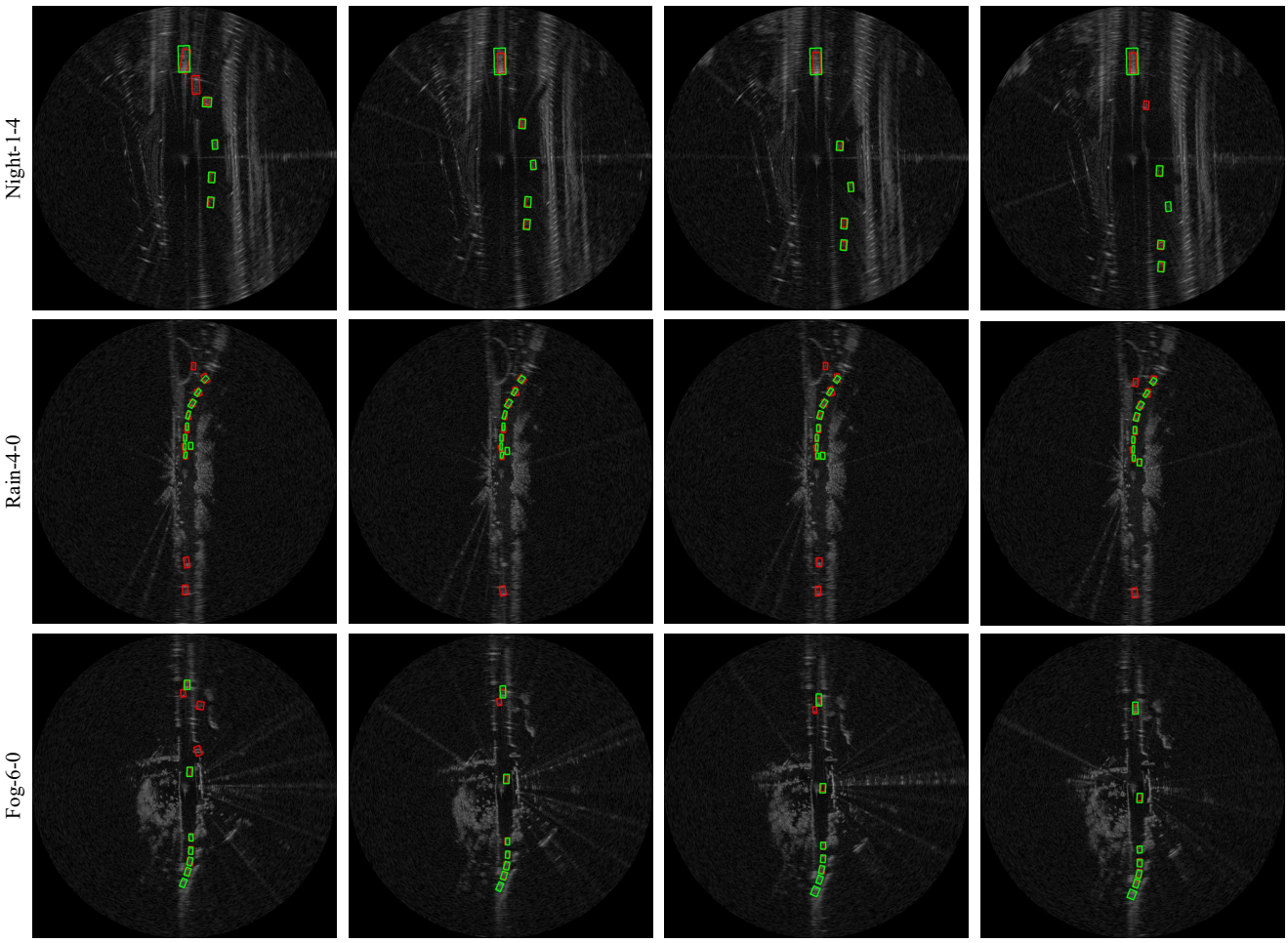


Figure 16. **Sampled detection results of SIRA-34 (6) with full size radar frames** on three scenarios: Night-1-4 (Top), Rain-4-0 (Middle) and Fog-6-0 (Bottom) on *Radiate*. Green boxes represent ground truth and red ones are predictions. The column represents consecutive radar frames.

vehicles exhibit nonlinear movement, the continuous tracking without interruptions underscores the effectiveness of MCTrack. However, in Fig. 19, SIRA does exhibit a slight presence of FPs, likely influenced by reflections from multipath or ghost objects, due to tracking across consecutive frames. Addressing such false information poses an intriguing challenge.

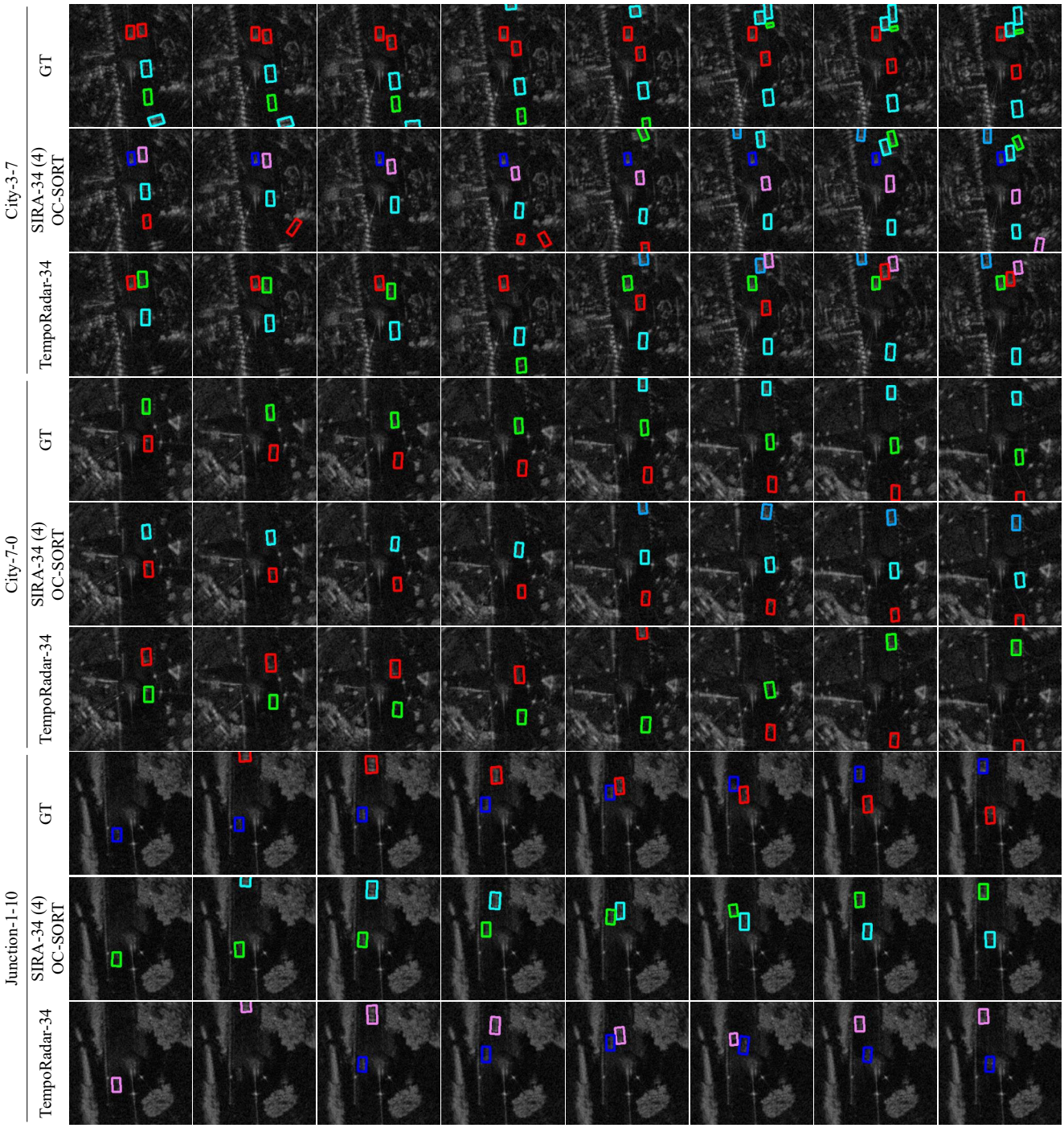


Figure 17. **Sampled tracking results** on three scenarios: City-3-7 (Top 3 Rows), City-7-0 (Middle 3 Rows) and Junction-1-10 (Bottom 3 Rows) on *Radiate*. For each scenario, we include ground truth (GT), SIRA (SIRA-34 (4)) and TempoRadar (TempoRadar-34). The color of bounding boxes represents the object ID. The column represents consecutive radar frames.

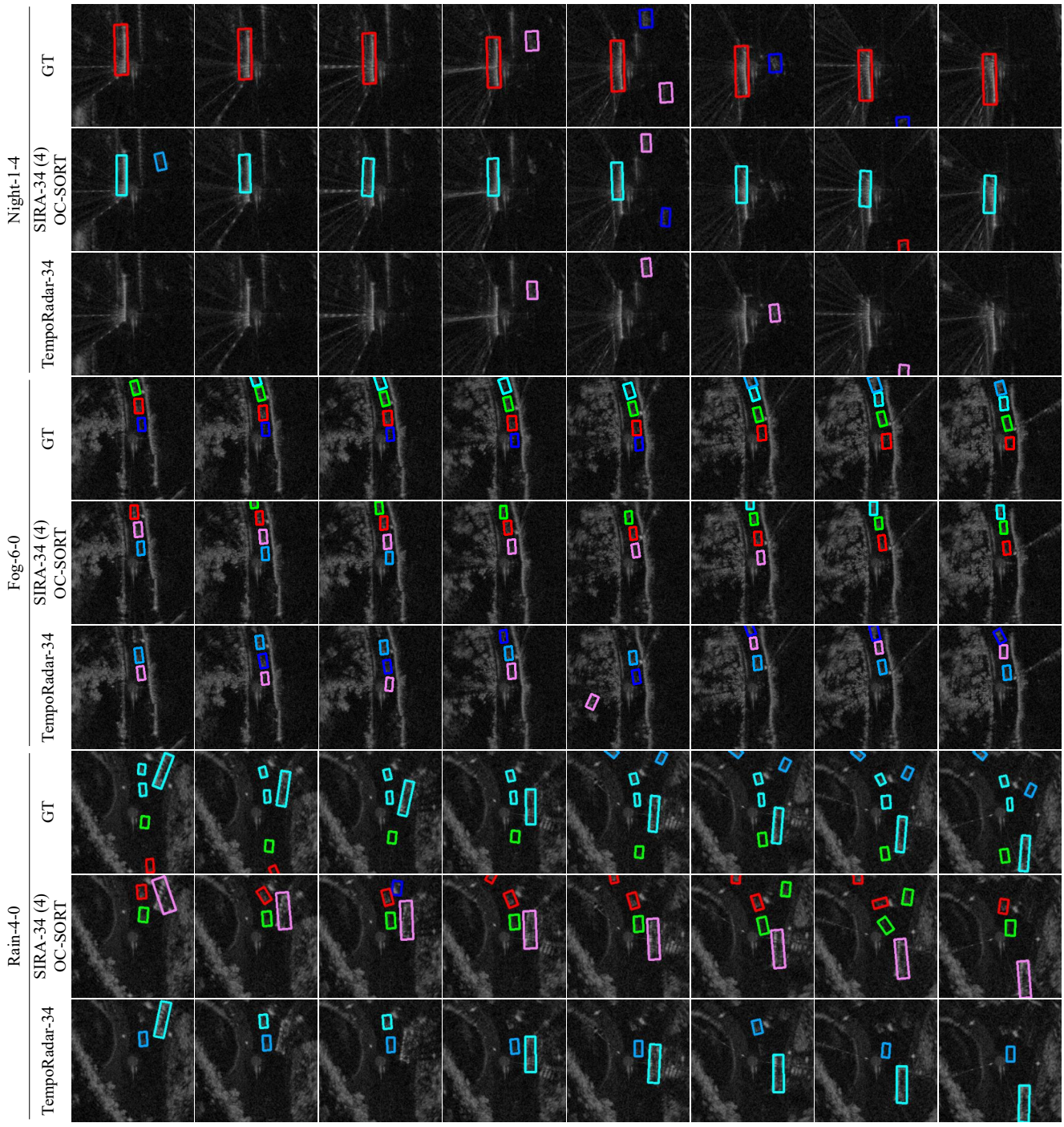


Figure 18. **Sampled tracking results** on three scenarios: Night-1-4 (Top 3 Rows), Fog-6-0 (Middle 3 Rows) and Rain-4-0 (Bottom 3 Rows) on *Radiate*. For each scenario, we include ground truth (GT), SIRA (SIRA-34 (4)) and TempoRadar (TempoRadar-34). The color of bounding boxes represents the object ID. The column represents consecutive radar frames.

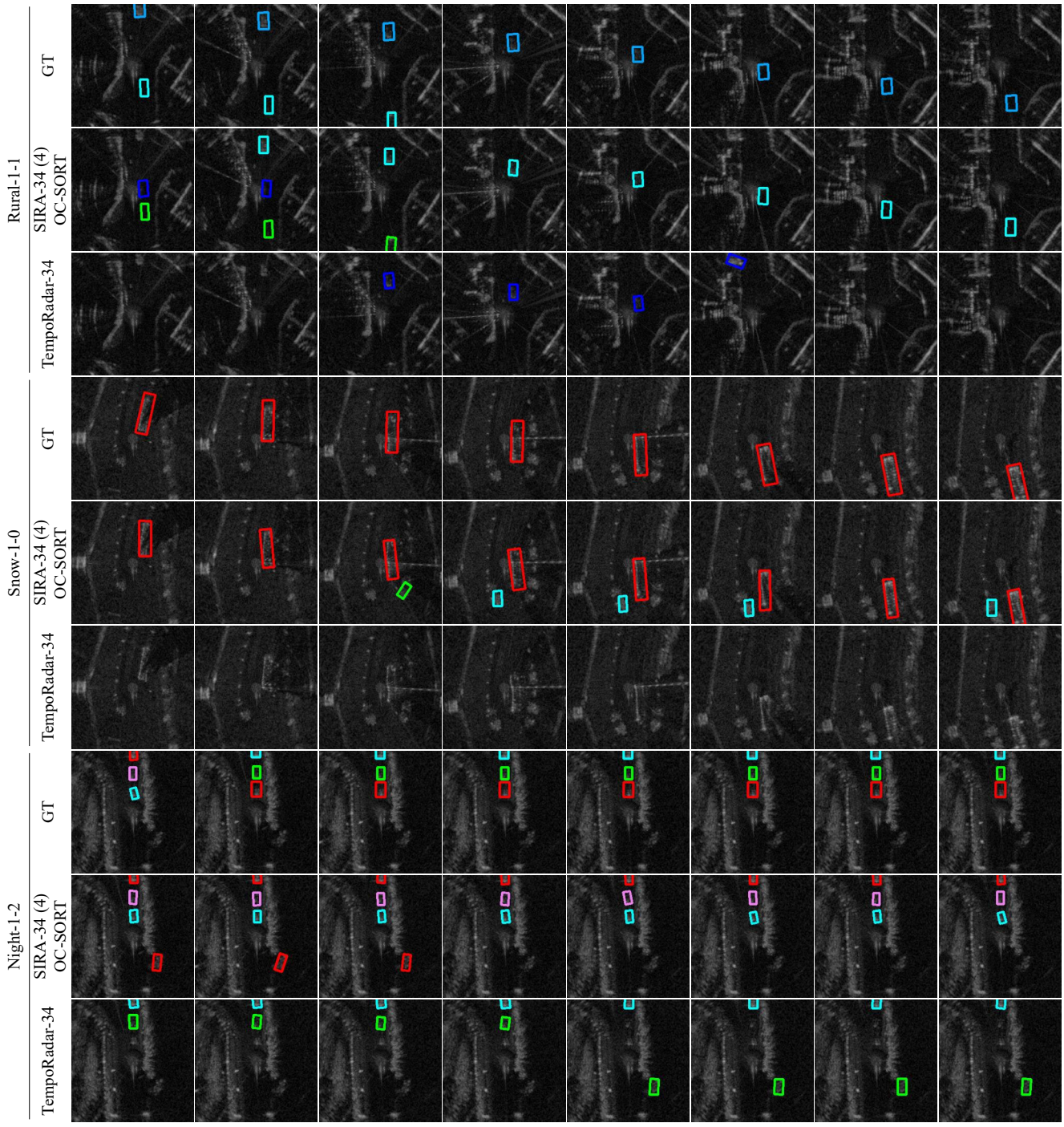


Figure 19. **Sampled tracking results** on three scenarios: Rural-1-1 (Top 3 Rows), Snow-1-0 (Middle 3 Rows) and Night-1-2 (Bottom 3 Rows) on *Radiate*. For each scenario, we include ground truth (GT), SIRA (SIRA-34 (4)) and TempoRadar (TempoRadar-34). The color of bounding boxes represents the object ID. The column represents consecutive radar frames.

17. KF-based Multiple Object Tracking for Radar Perception

Kalman Filter KF is a linear estimator for discretized dynamical systems in the time domain. KF operates by utilizing state estimations from the previous time step and current measurements to predict the target state at the next time step. The filter maintains two key variables: the posterior state estimate represented as $\mathbf{x}$, and the posterior estimate covariance matrix denoted as $\mathbf{P}$.

In the context of object tracking, the KF process is defined by several components, including the state transition model $\mathbf{F}$, the observation model $\mathbf{H}$, the process noise covariance $\mathbf{Q}$, and the measurement noise covariance $\mathbf{R}$. In each time step t, when presented with observations $\mathbf{z}_t$, the KF operates through a sequence of predict and update stages.

$$\text{predict} \begin{cases} \widehat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \widehat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t, \end{cases} \tag{26}$$

$$\text{update} \begin{cases} \mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{G}_t^\top \left(\mathbf{G}_t \mathbf{P}_{t|t-1} \mathbf{G}_t^\top + \mathbf{R}_t \right)^{-1} \\ \widehat{\mathbf{x}}_{t|t} = \widehat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{G}_t \widehat{\mathbf{x}}_{t|t-1} \right) \\ \mathbf{P}_{t|t} = \left(\mathbf{I} - \mathbf{K}_t \mathbf{G}_t \right) \mathbf{P}_{t|t-1} \end{cases} . \tag{27}$$

The prediction stage involves calculating the state estimations for the subsequent time step t. In contrast, the update stage is focused on refining the posterior parameters within the KF when presented with measurfor thents of target states for time step t. In many scenarios, this measurement is derived from the observation model $\mathbf{H}$ and is commonly referred to as an observation.

KF parameters In MOT, KF-based typically consists of five steps: Prediction, Association, Update, Deletion, and Initialization. The prediction and update phases are handled by KF. In our setting for radar perception, the KF's state $\mathbf{x}_t$ and observation $\mathbf{z}_t$ is defined as follows:

$$\mathbf{x}_t := \left(x_t, y_t, s_t, r_t, \vartheta_t, \dot{x}_t, \dot{y}_t, \dot{s}_t, \dot{\vartheta}_t \right)^\top, \tag{28}$$

$$\mathbf{z}_t := \left(x_t, y_t, \widehat{w}_t, \widehat{h}_t, \widehat{\vartheta}_t, \widehat{c}_t \;\middle|\; \widehat{c}_t > \gamma \right)^\top, \tag{29}$$

where (x_t, y_t) is the two-dimensional coordinates of the object center in the image. $s = w \times h$ is the bounding box scale (area), r is the bounding box aspect ratio and ϑ is object orientation, where w and h are the width and height of the object. The aspect ratio $r = \frac{w}{\text{float } (h+1\text{e}-6)}$ is assumed to be constant. The other four variables, $\dot{x}$, $\dot{y}$, $\dot{s}$ and $\dot{\vartheta}$ are the corresponding time derivatives. The detection confidence is c. The observation model is

$$\mathbf{G}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{30}$$

We note the process noise as in practice: $\mathbf{Q}_t = \text{diag}\left(\sigma_x^2, \sigma_y^2, \sigma_s^2, \sigma_r^2, \sigma_{\dot{\vartheta}}^2, \sigma_{\dot{u}}^2, \sigma_{\dot{v}}^2, \sigma_{\dot{s}}^2, \sigma_{\dot{\vartheta}}^2 \right)$. In the practice of SORT, we have to suppress the noise from velocity terms because it is too sensitive. We achieve it by setting a proper value for the process noise:

$$\mathbf{Q}_t = \text{diag}\left(0.1, 5, 1^{-4}, 1^{-4}, 10, 0.01, 0.01, 1^{-4}, 0.1 \right). \tag{31}$$

We note the linear transition model as:

$$\mathbf{F}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{32}$$

We set the measurement noise covariance as:

$$\mathbf{R}_t = 10\mathbf{I}_5. \tag{33}$$

We need to choose an initial value for $\mathbf{P}_{t-1|t-1}$, call it $\mathbf{P}_{0|0}$. If we were absolutely certain that our initial state estimate $\mathbf{x}_0 = \mathbf{0}$ was correct, we would let $\mathbf{P}_{0|0} = \mathbf{0}$. However, given the uncertainty in our initial estimate $\mathbf{x}_0$, choosing $\mathbf{P}_{0|0} = \mathbf{0}$ would cause the filter to initially and always believe $\mathbf{x}_t = \mathbf{0}$. Assuming some uncertainty in the initial state, we set as follows:

$$\mathbf{P}_{0|0} = \text{diag}\left(10, 10, 10, 10, 10, 10, 10, 10000, 10000 \right), \tag{34}$$

where $\dot{\vartheta}$ and $\dot{s}$ are set to a large value as the uncertainty is particularly high. On the other hand, we use the estimated pseudo-direction $\widehat{\mathbf{d}}_{T|T-1}$ as the initial value for $\dot{u}, \dot{v}$. Therefore, we set small uncertainties for these.

18. Fundamentals of FMCW for Automotive Radar

Radar technology offers a sensing solution that exhibits increased resilience to adverse weather conditions such as fog, rain, and snow. Typically, it generates low-resolution imagery, presenting significant challenges for tasks like object recognition and semantic segmentation. Contemporary

automotive radar systems are primarily based on the Multiple Input Multiple Output (MIMO) technique, which employs multiple transmitters and receivers to determine the direction of arrival (DOA) [22]. Although this approach is cost-effective, existing configurations often suffer from limited azimuth resolution. For example, a commercial radar system with a 15° angular resolution produces a cross-range image with an approximate span of 10 meters at a distance of 20 meters. Consequently, radar imagery does not provide the level of detail necessary for effective object recognition and detailed scene mapping. On the contrary, the scanning radar employs a mobile antenna to measure azimuth at each point, leading to significantly improved azimuth resolution [47].

Transmitter From [50], automotive radar predominantly employs a frequency-modulated continuous waveform (FMCW) for object detection, generating point clouds across multiple physical domains. As shown in Fig. 20, this is achieved by transmitting a series of K coded FMCW pulses from one of its M Tx transmitting antennas, given by the expression of the radio frequency (RF) wave form on Tx antenna m:

$$s_m(t) = \sum_{k=0}^{K-1} c_m(k)\, s_p(t - nT_{\text{PRI}})\, e^{j2\pi f_c t}, \tag{35}$$

$$s_p(t) = \begin{cases} e^{j\pi\beta t^2} & 0 \le t \le T \\ 0 & \text{otherwise} \end{cases}, \tag{36}$$

where $s_p(t)$ is the baseband FMCW waveform (chirp pulse) with β denoting the chirp rate and T the pulse duration, and is repeated K times. k is the index for pulse, and $c_m(k)$ is the slow-time orthogonal code for the k-the pulse at the m-th Tx antenna, which satisfies the following:

$$\sum_{k=0}^{K-1} c_i(k)\, c_m(k) = \begin{cases} K & \text{if } i = m \\ 0 & \text{otherwise} \end{cases}. \tag{37}$$

T_{PRI} is pulse repetition interval and f_c is the carrier frequency, e.g., $f_c = 79$ GHz. The bandwidth of the FMCW waveform is $B = \beta T$. The baseband waveform is repeated at each antenna before being multiplied by orthogonal codes $c_m(k)$, for example, the Hadamard code.

Receiver An object at a range of R_0 with a radial velocity v and a far-field spatial angle (i.e. azimuth, elevation, or both) induces amplitude attenuation and phase modulation to the received FMCW signal at each of N Rx receiver RF chains (including the low noise amplifier (LNA), local oscillator (LO), and analog-to-digital converter (ADC)) of Fig. 20. The round-trip propagation delay from m-th Tx antenna to its n-th Rx antenna is

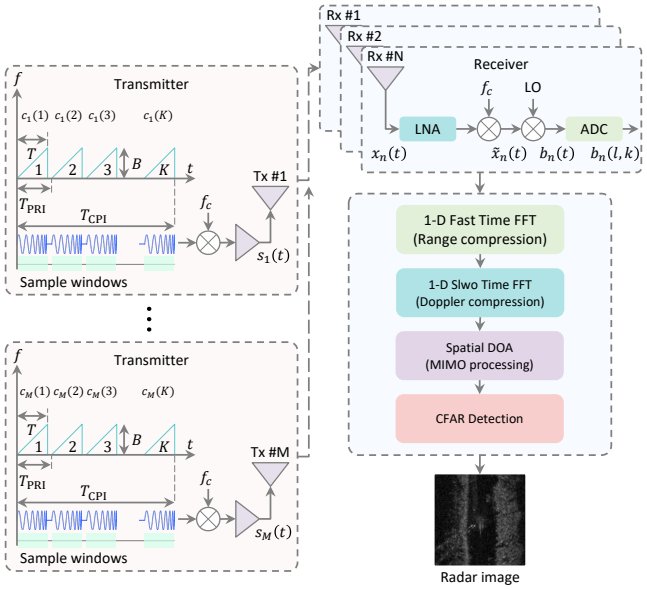


Figure 20. The slow-time FMCW automotive radar architecture from [50]. On the left, a sequence of FMCW pulses with orthogonal slow-time (pulse) codes are sent from M transmitting antennas while, on the right, each of N receivers uses the same source FMCW waveform to sample the beat signal followed by range-doppler processing and slow-time waveform separation for spatial detection.

$$\tau_{mn}(t) = 2\frac{R_0 + vt}{c} + m\frac{d_t \sin(\theta_t)}{c} + n\frac{d_r \sin(\theta_r)}{c}, \tag{38}$$

where d_t, d_r, θ_t and θ_r are the inter-element spacing and azimuthal angle for the transmitting and receiving antennas, respectively. We assume co-located radars and the far-field approximation, i.e. $\theta_r = \theta_t = \theta$. c is the speed of propagation. In the presence of an object at angle θ, the n-th Rx receiver receives the signal of a sum of M attenuated and delayed transmitting waveforms:

$$x_n(t) = \alpha \sum_{m=0}^{M-1} s_m(t - \tau_{mn})\, e^{j2\pi f_c(t-\tau_{mn})}. \tag{39}$$

Subsequently, the baseband signal after LNA and carrier frequency down conversion is as follows:

$$\tilde{x}_n(t) = x_n(t)\, e^{-j2\pi f_c t} \tag{40}$$

$$\approx \tilde{\alpha} \sum_{m=0}^{M-1} s_m(t - \tau_0)\, e^{-j2\pi f_c \frac{2vt}{c}} e^{-j2\pi(md_t + nd_r)\frac{\sin(\theta)}{\lambda}}, \tag{41}$$

where $\tau_0 = \frac{2R_0}{c}$ is the time taken from the transmission to the reception, and $\lambda = \frac{c}{f_c}$ is wave length. We assume that

$s_m\left(t-\tau_{mn}\right)=s_m\left(t-\tau_0\right)$, and $\tilde{\alpha}$ absorbs constant phase factors. By using LO, the signals at all receivers are mixed with the source chirp to generate the analog beat signal:

$$b_n\left(t\right)=\tilde{x}_n\left(t\right)\sum_{k=0}^{K-1}s_p^*\left(t-kT_{\mathrm{PRI}}\right), \tag{42}$$

where $*$ denotes its conjugate. This analog beat signal is then sampled at $t=kT_{\mathrm{PRI}}+l\Delta T$ with ADC sampling, where ΔT and T_{PRI} are the fast-time and slow-time sampling intervals, respectively, and digital beat signal is represented as follows:

$$b_n\left(l,k\right)=\tilde{\alpha}\sum_{m=0}^{M-1}c_m\left(k\right)\underbrace{e^{-j2\pi f_r l}}_{\text{Range}}\underbrace{e^{-j2\pi f_d k}}_{\text{Doppler}}\underbrace{e^{-j2\pi\left(f_s^t m+f_s^r n\right)}}_{\text{Virtual Spatial Array}}, \tag{43}$$

where $f_r=\left(\beta\tau_0+2f_c\frac{v}{c}\right)\Delta T$ is normalized range (fast-time) frequency, $f_d=2f_cT_{\mathrm{PRI}}\frac{v}{c}$ is the normalized Doppler (slow-time) frequency, and f_s^t and f_s^r are the normalized spatial frequency at the transmitting and receiving antennas. f_s^t is usually different from f_s^r due to different Tx/Rx spacings. In other words, the beat signal $b_n\left(l,k\right)$ at n-th receiver is the sum of the object responses originating from all transmitted waveforms, coded using $c_m\left(k\right)$. The beat signal at each of N Rx receiver forms a matrix:

$$\mathbf{B}_n=\begin{bmatrix} b_n\left(1,1\right) & b_n\left(2,1\right) & \dots & b_n\left(L,1\right) \\ b_n\left(1,2\right) & b_n\left(2,1\right) & \dots & b_n\left(L,2\right) \\ \vdots & \vdots & \ddots & \vdots \\ b_n\left(1,K\right) & b_n\left(2,2\right) & \dots & b_n\left(L,K\right) \end{bmatrix}. \tag{44}$$

The induced modulation from the target is captured by the baseband signal processing block (including fast Fourier transforms (FFT) over range, Doppler, and spatial domains). All these processes lead to a multi-dimensional spectrum. With the constant false alarm rate (CFAR) detection step that compares the spectrum with an adaptive threshold, radar point clouds are generated in the range, Doppler, azimuth, and elevation domains [5, 26, 27, 50]. Considering the computing and cost constraints, automotive radar manufacturers may define the radar point clouds in a subset of the full four dimensions. For example, traditional automotive radar generates detection points in the range-Doppler domain, whereas some produce the points in the range-Doppler azimuth plane [44]. In the *Radiate* dataset [47] considered in this paper, the radar point cloud is defined in the range azimuth plane with a 360° field view. The resulting polar coordinate point cloud is further transformed into an ego-centric Cartesian coordinate system, then a standard voxelization can convert the point cloud into a radar frame as $I_t\in\mathbb{R}^{1\times H\times W}$.