Algorithm 1 Proposed SORT Tracking Pipeline

Input: Detection set Δ_t from frame t
Output: Tracked object set Θ_t

Initialize Kalman filters $\mathcal{K}$ for all targets
for each frame step t **do**
    Use $\mathcal{K}$ to predict next states of all active tracks
    Compute SIoU cost matrix $\mathcal{I}$ between predictions and Δ_t by the MDESM
    Solve assignment using Hungarian Algorithm
    for each matched pair (track i, detection j) **do**
        if $\mathcal{I}(i, j) > \tau$ **then**
            Update track i with detection j
            Correct the track label by the DESM module
        else
            Mark track i as unmatched
        end if
    end for
    for each unmatched detection in Δ_t **do**
        Initialize new track with detection
    end for
    for each unmatched track **do**
        Increase track age
        if track age $> \gamma$ **then**
            Remove track
        end if
    end for
end for

Algorithm 2 Cost Matrix with MDESM

Function ComputeCostMatrix($\hat{\Theta}_t$, Δ_t, Φ_t)

Input: Predicted tracks $\hat{\Theta}_t$, detections Δ_t, appearance features Φ_t

Output: Fused cost matrix $\mathcal{C}$

for each predicted track $\theta_i \in \hat{\Theta}_t$ **do**
  for each detection $\delta_j \in \Delta_t$ **do**
    Compute SIoU distance: $\mathcal{I}_{i,j} \leftarrow 1 - \text{SIoU}(\theta_i, \delta_j)$
    Compute appearance distance: $\mathcal{D}_{i,j} \leftarrow \|\phi_i - \phi_j\|_2$
    Fuse costs: $\mathcal{C}_{i,j} \leftarrow \lambda \cdot \mathcal{I}_{i,j} + (1 - \lambda) \cdot \mathcal{D}_{i,j}$
  end for
end for
for each row $\mathcal{C}_i \in \mathcal{C}$ **do**
  if $\mathcal{C}_i$ have multiple similar values **then**
    Split $\mathcal{C}_i$ into multiple rows $\{\mathcal{C}_{i_0}, \mathcal{C}_{i_1}, ...\} \leftarrow \mathcal{C}_i$
    Set rows $\{\mathcal{C}_{i_0}, \mathcal{C}_{i_1}, ...\}$ at end of $\mathcal{C}$
  end if
end for
return $\mathcal{C}$