Multi-Object Tracking System:
From Kalman Filtering to Deep ReID

Dimitri Calmand

14 December 2025

1 Introduction

This report presents the development of a comprehensive multi-object tracking (MOT) system implemented across four progressive exercises. The project demonstrates the evolution from basic Kalman filtering to advanced tracking algorithms that combine geometric and appearance-based features for robust object association.

2 Project Overview

2.1 Exercise 1: Kalman Filter Implementation

The foundation of the project involved implementing a 2D Kalman Filter to track a moving object. The filter estimates the state vector $\mathbf{x}_k = [x, y, v_x, v_y]^T$, where (x, y) represents position and (v_x, v_y) represents velocity.

Key Components:

- **State Transition Matrix A**: Models constant velocity motion with $x_{k+1} = x_k + v_x \cdot \Delta t$
- **Measurement Matrix H**: Extracts position observations from state vector
- **Process Noise Q**: Accounts for acceleration uncertainty
- **Measurement Noise R**: Models sensor (detector) uncertainty

The predict-update cycle enabled smooth trajectory estimation even with noisy detections. Visual output shows detection (green), prediction (blue), and corrected estimate (red) with trajectory trails.

2.2 Exercise 2: IoU-Based Multi-Object Tracking

This exercise extended single-object tracking to multiple pedestrians using the ADL-Rundle-6 dataset with YOLOv5l detections.

Implementation:

- **Data Association**: Hungarian algorithm solves the assignment problem by maximizing total IoU between tracks and detections
- **Cost Matrix**: Defined as $C = 1 - \text{IoU}$ to convert similarity to cost
- **Track Management**: New tracks created for unmatched detections; tracks deleted after 10 consecutive missed frames
- **IoU Threshold**: Set to 0.5 to filter spurious associations

The system successfully tracks multiple pedestrians but struggles with occlusions and identity switches due to reliance solely on geometric overlap.

2.3 Exercise 3: Kalman-IoU Integration

To improve tracking robustness, I integrated the Kalman Filter into the IoU-based tracker. Each track maintains its own Kalman Filter instance.

Modified Pipeline:

1. **Prediction Step**: All tracks predict their next position using Kalman motion model

2. **Association Step**: Compute IoU between predicted track positions and current detections
3. **Update Step**: Matched tracks update their Kalman state with detection measurements

Benefits:

- Predicts object location during temporary occlusions or missed detections
- Smoother trajectories due to velocity estimation
- Better handling of fast-moving objects through motion prediction

The key insight is representing bounding boxes by their centroids for Kalman filtering while maintaining full bounding box dimensions for IoU computation.

2.4 Exercise 4: ReID-Enhanced Tracking

The final exercise incorporated deep learning-based Re-Identification (ReID) to add appearance similarity to the tracking system.

ReID Implementation:

- **Model**: Pre-trained OSNet (reid_osnet_x025_market1501.onnx) for lightweight feature extraction
- **Preprocessing**: Crop patches at bounding boxes, resize to 64×128, convert BGR→RGB, normalize
- **Feature Extraction**: Extract 512-dim feature vectors, normalized to unit length
- **Similarity Metric**: Euclidean distance converted to similarity: $S_{\text{ReID}} = \frac{1}{1+d_{\text{Euclidean}}}$

Combined Score:

$$S_{\text{total}} = \alpha \cdot \text{IoU} + \beta \cdot S_{\text{ReID}}$$

With $\alpha = \beta = 0.5$, this balances geometric and appearance information. The IoU threshold was lowered to 0.3 since the combined metric is more discriminative.

Results: The ReID-enhanced tracker significantly reduces identity switches, especially during occlusions where appearance features maintain correct associations even when spatial overlap is minimal.

3 Challenges and Solutions

Challenge 1: State Representation Mismatch *Problem:* Kalman filter operates on centroids (c_x, c_y) while IoU requires full bounding boxes (x, y, w, h). *Solution:* Convert between representations: predict centroids with Kalman, reconstruct bounding boxes assuming constant width/height during prediction.

Challenge 2: Balancing IoU and ReID Weights *Problem:* Finding optimal α and β values for different scenarios. *Solution:* Used equal weights (0.5, 0.5) as baseline. In production, these should be tuned per dataset or made adaptive based on detection confidence.

Challenge 3: Track Initialization and Deletion *Problem:* Premature deletion of occluded tracks vs. accumulation of ghost tracks. *Solution:* Tuned maximum missed frames parameter (10) and threshold (0.3-0.5) through experimentation on the validation set.