

# Object Tracking Project Report

## 1 Introduction

The goal of this project is to implement and analyze various object tracking algorithms, ranging from single object tracking using Kalman Filters to advanced multi-object tracking using visual appearance features. We progress through four stages (TP1 to TP4), improving the complexity and robustness of the tracking system with each step.

## 2 Code Structure and Analysis

### 2.1 TP1: Single Object Tracking with Kalman (Centroid-Tracker)

The first TP focuses on tracking a single object (a black dot) using a Kalman Filter.

- **Detector.py:** Provides a simple detection mechanism to find the centroid of the object in each frame.
- **KalmanFilter.py:** Implements the prediction and update phases of the Kalman Filter. It maintains the state of the object (position and velocity) and predicts its future location.
- **objTracking.py:** The main loop. It predicts the next state using the Kalman Filter, attempts to associate the prediction with the closest detection using Euclidean distance, and then updates the filter. If no measurement is found, it relies on the prediction.

### 2.2 TP2: IOU-based Tracking (Bounding-Box Tracker)

TP2 introduces multi-object tracking using the Intersection over Union (IoU) metric.

- **iou\_tracker.py:** Implements the `IOUTracker` class.
- **Association:** It computes an IoU matrix between all current tracks and new detections. The Hungarian algorithm (`scipy.optimize.linear_sum_assignment`) is used to find the optimal assignment that maximizes the total IoU.
- **Lifecycle:** Tracks are created for unmatched detections and deleted if they are not updated for `max_age` frames. This allows the system to handle people entering and leaving the scene.

### 2.3 TP3: Kalman-Guided IoU Tracking (Bounding-Box Tracker)

TP3 combines the Kalman Filter from TP1 with the IoU matching from TP2 to create a more robust tracker.

- **kalman\_iou\_tracker.py:** The `Track` class now encapsulates its own `KalmanFilter`.
- **Prediction:** Before association, every track predicts its new bounding box using the Kalman Filter.
- **Association:** IoU is calculated between the predicted positions and the new detections (rather than the previous known positions). This helps bridge gaps when objects move quickly or are slightly occluded, as the Kalman Filter anticipates the movement.

## 2.4 TP4: Appearance-Aware IoU-Kalman Object Tracker

TP4 adds visual features to solve the Identity Switching problem common in IoU-based trackers.

- **appearance\_aware\_tracker.py**: Integrates a deep learning model for Re-Identification (ReID).
- **Feature Extraction**: Uses an ONNX model (`osnet`) to extract a 512-dimensional feature vector for each detection.
- **Cost Function**: The matching cost is a weighted combination of spatial distance (IoU) and visual similarity (Cosine distance of feature vectors).
- **Advantage**: Using appearance features allows the tracker to re-identify objects even after they have been fully occluded or if their motion is unpredictable, as long as they look the same.

## 3 Critical Analysis

### 3.1 TP1 Analysis

The video shows a rectangle that consistently follows the black dot. The Kalman Filter successfully predicts the path of the single object, smoothing the movement and handling minor noise in detection.

### 3.2 TP2 Analysis

We are able to track multiple people in the scene. However, the tracking is not stable, the IDs assigned to each person keep changing. This occurs because pure IoU tracking fails when detections are missed or when the overlap between the previous and current frame is insufficient.

### 3.3 TP3 Analysis

The tracking stability is improved compared to TP2. We can track each person, and some IDs remain consistent even when someone passes behind someone else.

- **Observation**: For example, user 66 passes behind user 62 and successfully keeps her ID. This was not the case in previous versions.
- **Limitation**: However, it is not perfect. Some IDs still change. For instance, user 81 passes behind user 62 and incorrectly becomes user 87. This shows that spatial prediction alone (Kalman Filter) is sometimes insufficient during complex occlusions.

### 3.4 TP4 Analysis

This version shows the most robust performance. Most IDs remain stable even through challenging occlusions where TP3 failed.

- **Enhancements**:
  - ID 2 remains ID 2 when passing behind ID 6.
  - ID 46 remains ID 46 behind ID 44.
  - ID 58 remains ID 58 behind ID 44.
- **Conclusion**: Since the total number of unique IDs generated is lower than in the previous videos, we can confirm the system is better. Fewer IDs means fewer false new tracks and fewer ID switches, indicating a more stable tracking system.

## 4 Conclusion

The project successfully demonstrates the evolution of object tracking systems. Starting from basic single-object tracking, we identified the limitations of spatial overlap (IoU) and motion prediction (Kalman Filter). The final implementation, incorporating appearance features (ReID), provides a significantly more robust solution capable of handling occlusions and maintaining consistent identities in a crowded scene.