Simple Online and Real-time Tracking for the Multi-object tracking Problem

Mazen Elaraby 19P9804
Muhammad Ahmed 19P6131

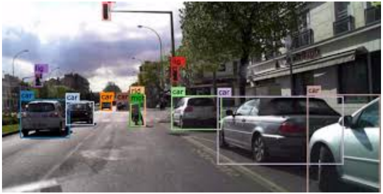
Ain Shams University

May 18, 2023

Multi-Object Tracking

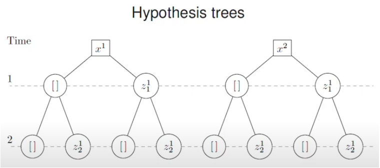
Definition of Multi-Object Tracking

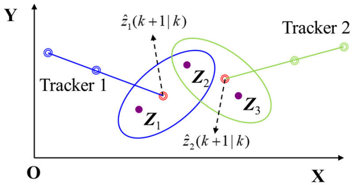
Multiple object tracking (MOT) is a fundamental problem in computer vision that involves estimating the trajectories of multiple objects over time, given a sequence of observations from one or more sensors.



Multi-Object Tracking

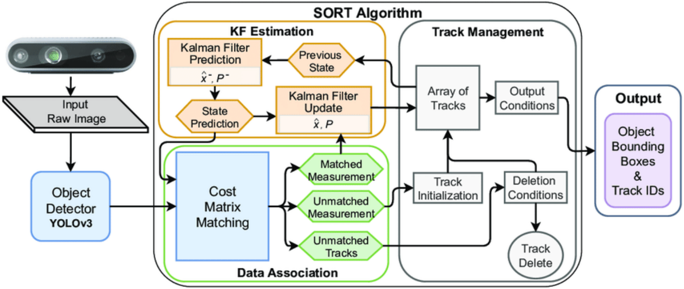
- Traditional multiple object tracking algorithms like Multiple Hypothesis Testing (MHT) and Joint Probabilistic Data Association (JPDA) suffer from high combinatorial complexity that can lead to a combinatorial explosion.
- The problem arises when the number of objects to be tracked increases, as the algorithms need to consider all possible associations between detections and tracks, leading to an exponential increase in computational requirements.





Introduction to SORT

- The Simple Online Real-time Tracking algorithm, commonly known as SORT, is a popular object tracking algorithm that has gained significant attention in recent years due to its simplicity and reliability
- SORT abides by the principle of Occam's Razor, which means it focuses on efficiency and simplicity rather than robustness against edge cases.



Detection

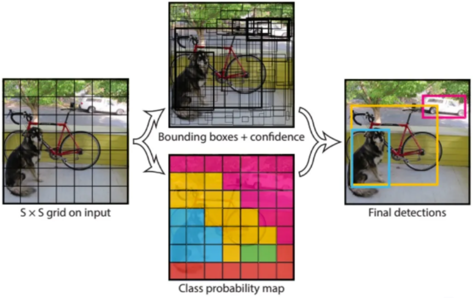
- The detection phase is critical in the Simple Online Real-time Tracking (SORT) algorithm as it provides the initial set of detections for tracking objects over time.
- To achieve accurate results, reliable methods must be used for detecting objects in a scene. Noisy or inaccurate detections can cause tracking algorithms to fail and result in incorrect object associations.

Detection: YOLOv4

- YOLOv4, a state-of-the-art object detection algorithm based on deep neural networks, has shown exceptional performance in various benchmark evaluations due to its accuracy and speed.
- YOLO performs object detection, localization, and classification all at once
- It uses a single-pass approach, making it faster and more efficient than older algorithms

Detection: YOLOv4

- The pipeline consists of feature extraction, learning, and non-maximal suppression stages
  - Feature extraction involves using a CNN to extract relevant features from the input image
  - The learning stage predicts bounding boxes and class probabilities for each object in the image based on extracted features
  - Non-maximal suppression removes redundant bounding boxes and classifies the remaining detections.



Estimation Model: Recursive Bayesian State Estimation

Definition of the Bayes Filter

Recursive Bayesian state estimation is a method for estimating the state of a dynamic system over time using probabilistic modeling. It is performed recursively over time using incoming measurements and a mathematical process model.

- Recursive Bayesian state estimation is advantageous because it can handle uncertainty and noise in the system by representing the system's state as a PDF.
- This makes it useful in situations with random disturbances or measurement errors.

Estimation Model: Recursive Bayesian State Estimation

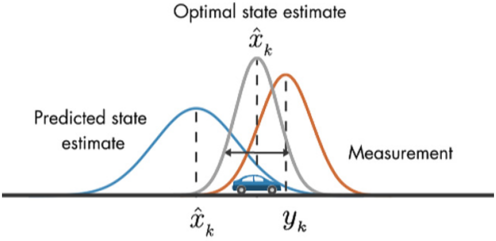
The Bayes Filter uses a Prediction-Correction Mechanism.

- Prediction Step:

$$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}).bel(x_{t-1})dx_{t-1}$$

- Correction Step:

$$bel(x_t) = \eta.p(z_t|x_t).\overline{bel}(x_t)$$

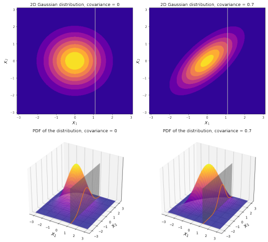


Definition

The Kalman Filter is a realization of the Bayes Filter for linear motion and observation models, and uses a Gaussian distribution to represent probability distributions. The filter provides an optimal estimate of the system state along with an estimate of uncertainty.

Estimated distributions by the Kalman filter assume the following form:

$$p(x) = det(2\pi\Sigma)^{-1/2} \exp\left(-1/2(x-\mu)^T\Sigma^{-1/2}(x-\mu)\right)$$



Estimation Model: The Kalman Filter

Both the observation and motion model can be modeled as the following linear models:

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$$

$$z_t = C_t x_t + \delta_t$$

where:

$$A_t : \text{The State Transition Matrix}$$

$$B_t : \text{The Control-input Model}$$

$$C_t : \text{The Observation model}$$

$$\epsilon_t \& \delta_t : \text{Random variables representing the process}$$

$$\text{and measurment noise with covariance } R_t \& Q_t$$

Estimation Model: The Kalman Filter

Motion under Gaussian noise leads to:

$$p(x_t|u_t, x_{t-1}) = det(2\pi R_t)^{-1/2}$$

$$\exp\left(-1/2(x_t - A_t x_{t-1} - B_t u_t)^T R^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right)$$

Measuring under Gaussian noise leads to:

$$p(z_t|x_t) = det(2\pi Q_t)^{-1/2} \exp\left(-1/2(z_t - C_t x_t)^T Q^{-1}(z_t - C_t x_t)\right)$$

Estimation Model: The Kalman Filter

Plugging these distribution into the recursive Bayes filter yields the final form of the Kalman filter algorithm represented in the following pseudo-code:

$$\text{Kalman Filter}(\mu_{t-1}, \Sigma t-1, u_t, z_t)$$

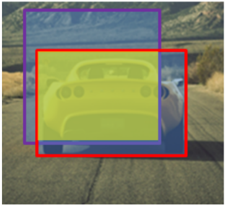
$$\bar{\mu_t} = A_t \mu_{t-1} + B_t u_t$$
$$\bar{\Sigma}_t = A_t \Sigma t - 1 A_t^T + R_t$$
$$K_t = \bar{\Sigma}_t C_t^T \left(C_t \bar{\Sigma}_t C_t^T + Q_t \right)^{-1}$$
$$\mu_t = \bar{\mu_t} + K_t (z_t - C_t \bar{\mu_t})$$
$$\Sigma_t = (I - K_t C_t)\, \bar{\Sigma}_t$$

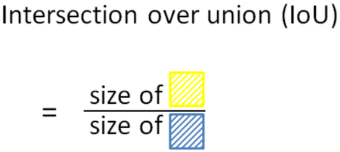
$$\text{return } \mu_t, \Sigma_t$$

Data Association

Data association is the process of associating observations or measurements with existing tracks in a tracking algorithm.

- The process of assigning detections to existing targets involves estimating the location of each target's bounding box and calculating the IOU distance between each detection and all predicted bounding boxes from existing targets. This is used to determine the assignment cost matrix.

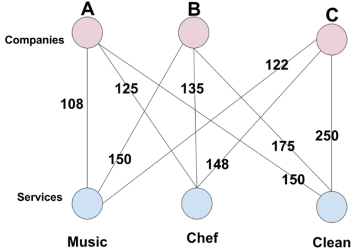




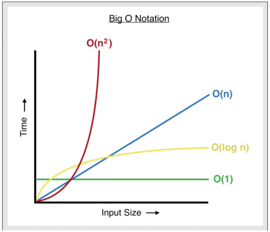
Data Association: The Assignment Problem

- The Hungarian Algorithm is used to optimally solve the assignment problem represented by the cost matrix.

Company	Cost for Musician	Cost for Chef	Cost for Cleaners
Company A	$108	$125	$150
Company B	$150	$135	$175
Company C	$122	$148	$250



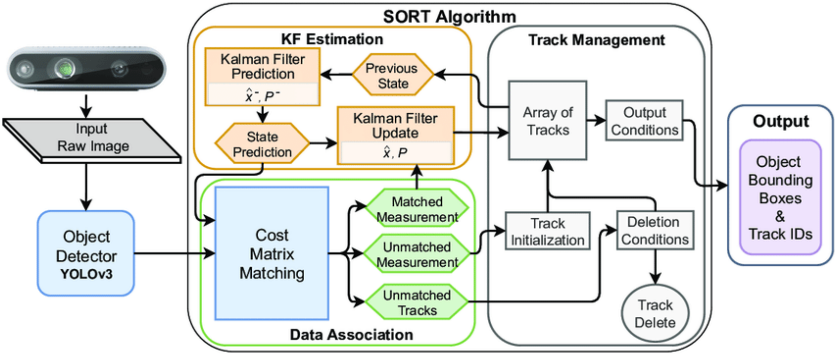
The Hungarian Algorithm manages to reduce the computational complexity of the Assignment Problem from $O(n!)$ to $O(n^3)$.



Track Maintenance

- Objects can leave the video frame or become occluded for brief or long periods. we need to define the maximum number of frames without assigned detections, T_{Lost}, before deleting a track.
- Additionally, SORT requires an object to be detected in two consecutive frames before confirming a track.

Overview of the Pipeline



Performance Analysis: The CLEAR MOT Metrics

The CLEAR multi-object tracking metrics provide a standard set of tracking metrics to evaluate the quality of tracking algorithm

- Multi-Object Tracking Accuracy (MOTA)
- Multi-Object Tracking Precision (MOTP)
- Mostly Tracked
- Partially Tracked
- Mostly Lost
- False Positive
- False Negative
- Recall
- Precision
- False Track Rate
- ID Switches
- Fragmentations

Performance Analysis: The CLEAR MOT Metrics

Below is the evaluation of these metrics on our implementation:

>> SORT

Tracker	MOTA (%)	MOTP (%)	Mostly Tracked (%)	Partially Tracked (%)
"ACF+SORT"	27.315	65.216	64.286	35.714
"YOLOv4+SORT"	82.099	90.48	78.571	14.286

Mostly Lost (%)	False Positive	False Negative	Recall (%)
0	290	172	73.457
7.1429	21	94	85.494

Recall (%)	Precision (%)	False Track Rate	ID Switches	Fragmentations
73.457	62.141	1.716	9	15
85.494	96.348	0.12426	1	9

Pros & Cons

- Pros:
  - efficient & extremely fast
  - low computational complexity due to simple linear motion model
  - achieves state-of-the-art performance in MOTA
  - low number of lost targets in comparison to the other methods.
- Cons:
  - high number of identity switches when subjected to long-term occlusions
  - simplicity of motion model is problematic at low frame rates

Applications

- Object tracking in video surveillance
- Autonomous driving and vehicle tracking
- Augmented reality and virtual reality applications
- Sports analysis and player tracking
- Robotics and unmanned aerial vehicles (UAVs) tracking

Thank you for your time!