ECE359: Signal Processing for Multimedia Major Task Report

Mazen Elaraby 19P9804

Muhammad Ahmed 19P6131

May 18, 2023

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

Submitted To:

Dr. Mohamed Atef Abdelmoniem

Contents

1 Introduction to the Multiple Object Tracking (MOT) Problem

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. This problem arises in many applications such as surveillance, autonomous driving, robotics, and sports analysis. The goal of MOT is to accurately track the location, size, shape, and motion of multiple objects simultaneously, despite occlusions, clutter, and other sources of noise in the data.

Tracking multiple objects is challenging due to several factors such as variations in appearance, non-rigid deformations, complex interactions between objects, and changes in lighting conditions. Furthermore, the number of objects and their identities may vary over time, making it difficult to maintain consistency in tracking. These challenges require the development of robust algorithms that can handle these variations and provide accurate and reliable tracking results.

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. This makes them inefficient in real-time multiple object tracking in dynamic environments where objects move quickly, change direction frequently, and occlude each other.

2 Simple Online Real-time Tracking (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 is designed to provide efficient and reliable online object tracking in complex and dynamic environments, where objects move quickly and occlude each other frequently. Unlike traditional multiple object tracking algorithms that can suffer from high combinatorial complexity and require considerable computational resources, SORT abides by the principle of Occam's Razor, which means it focuses on efficiency and simplicity rather than robustness against edge cases. This approach has made SORT an attractive solution for various applications, including surveillance, robotics, and autonomous driving, where real-time tracking accuracy is crucial.

2.1 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. In order to achieve good results, it is essential to use reliable and accurate methods for detecting objects in the scene. If the detections are noisy or inaccurate, then the algorithm may fail to track objects correctly, leading to incorrect object associations. 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. By using YOLOv4 for detection, SORT can leverage high-quality detections with faster processing times, allowing it to efficiently track objects in dynamic and complex environments. The combination of YOLOv4 and SORT provides an effective solution for real-time object tracking applications, such as surveillance, autonomous driving, and robotics, where tracking accuracy and speed are of utmost importance.

You Only Look Once (YOLO) is a popular object detection algorithm that is widely used in computer vision applications. YOLO uses a single-pass approach to detect objects in an image, making it faster and more efficient than older algorithms that require multiple passes. The YOLO pipeline consists of several stages, including feature extraction, learning, and non-maximal suppression. In the feature extraction stage, YOLO takes an input image and extracts relevant features using a convolutional neural network (CNN). Next, the learning stage processes the extracted features to predict bounding boxes and class probabilities for each object in the image. Finally, the non-maximum suppression stage removes redundant bounding boxes and classifies the remaining detections.

One of the most significant advantages of YOLO is its speed. By using a single pass through the network, YOLO can process high frame rates in real-time, making it suitable for edge AI applications such as surveillance systems, drones, and autonomous vehicles. Additionally, YOLO has a small memory footprint, making it ideal for deployment on resource-constrained devices. Overall, YOLO's fast performance, accuracy, and suitability for edge AI applications make it one of the most popular object detection algorithms in use today.

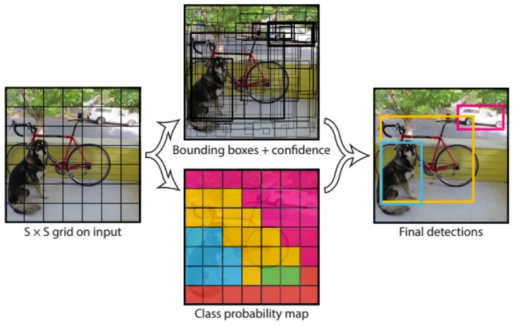

Figure 1: Overview of YOLO functionality

2.2 Estimation Model

The estimation model used by the SORT algorithm is based on recursive Bayesian state estimation realized by a linear Kalman filter. The Kalman filter is a mathematical model that allows us to estimate the state of a system based on a motion model and noisy measurements over time.

The following is an exposition of the underlying theory of the estimation model.

2.2.1 Recursive Bayesian State Estimation

Recursive Bayesian state estimation is a probabilistic method for estimating the state of a dynamic system over time, in which the system's state is characterized by an unknown probability density function (PDF). This technique is based on Bayes' theorem, which allows us to update our belief about the system's state given new observations. The main idea behind recursive Bayesian state estimation is to start with a prior PDF that reflects our initial knowledge or assumptions about the system's state and then update it recursively as new measurements become available.

The process involves two key components: a mathematical model that describes how the system's state evolves over time, and a measurement model that relates the system's state to the observed measurements. By combining these two models using Bayes' theorem, we obtain a posterior PDF that represents our updated belief about the system's state given all available information.

The posterior PDF can be used to make predictions about future states of the system, as well as to estimate the current state based on the most recent observations.

One of the advantages of recursive Bayesian state estimation is its ability to handle uncertainty and noise in the system. By representing the system's state as a PDF rather than a single point estimate, we can account for a range of possible values and their associated probabilities. This makes the approach particularly useful in situations where the system is subject to random disturbances or measurement errors. Overall, recursive Bayesian state estimation provides a powerful framework for tracking the changing state of a system in real-time, enabling better decision-making and control.

The following is the mathematical formulation of the above-mentioned theory:

Prediction:

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

Correction:

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

2.2.2 The Kalman Filter

The Kalman Filter is a recursive algorithm used for state estimation in systems that are modeled by linear equations. It is a realization of the Bayesian approach to state estimation, where the posterior probability of the system state is updated recursively based on the current measurement and prior knowledge of the system dynamics. The KF assumes that both the system dynamics and the observation model are linear, and it uses a Gaussian distribution to represent the probability distributions of both the system state and the measurement noise. By combining the measurements with the prior knowledge of the state, the KF provides an optimal estimate of the true system state, while also providing an estimate of the uncertainty associated with this estimate.

Since the Kalman filter is optimal for linear models and Gaussian distributions, the probability distributions mentioned in the previous section will be modeled as a multivariate Gaussian:

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

And 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 : The State Transition Matrix

B_t : The Control-input Model

C_t : The Observation model

$\epsilon_t \& \delta_t$: Random variables representing the process and measurment noise with covariance $R_t \& Q_t$

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:

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

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

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

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

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

The following is an illustration of the predict-correct mechanism of the Kalman filter:

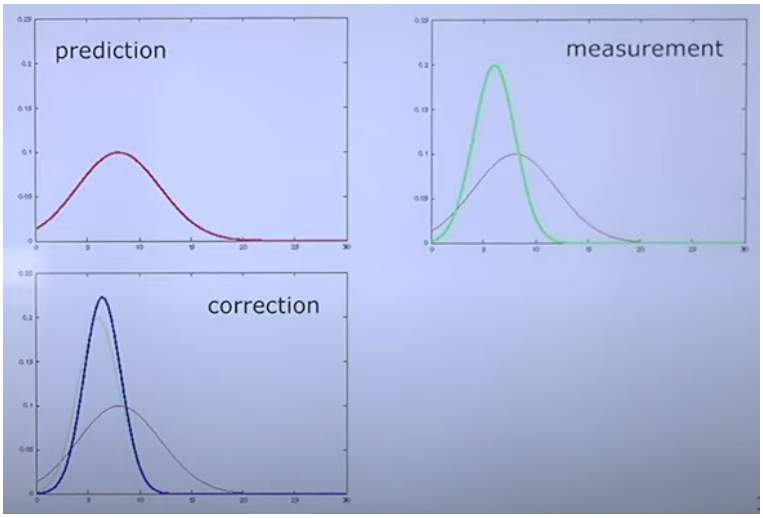

Figure 2: The Predict-Correct Mechanism

2.2.3 Implementation

In our implementation, The detection measurement is a 2D bounding box:

$$Z = [x, y, w, h]$$

The estimated state vector is as follows:

$$X = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}, \dot{r}]$$

The equations to convert measurement to state is pre-processed to avoid nonlinear observation model:

$$u = x + w/2$$

$$v = y + h/2$$

$$s = w.h$$

$$r = w/h$$

Assume the detection noise is zero-mean Gaussian, with a covariance R that corresponds to a standard deviation of 1 for the center position and the aspect-ratio. It also has a standard deviation of $\sqrt{10}$ pixels for the scale.

$$Q_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{10} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Our state transition matrix assumes the following partitioned matrix with δ_t denoting frames per second:

$$A_t = \begin{bmatrix} I_4 & \delta_t \mathrm{x} I_4 \\ I_4 & I_4 \end{bmatrix}$$

2.3 Data Association

In assigning detections to existing targets, each target's bounding box geometry is estimated by predicting its new location in the current frame. The assignment cost matrix is then computed as the intersection-over-union (IOU) distance between each detection and all predicted bounding boxes from the existing targets. The assignment is solved optimally using the Hungarian algorithm. Additionally, a minimum IOU is imposed to reject assignments where the detection to target overlap is less than the IOU minimum value.

2.3.1 Association Metric

Intersection over Union (IoU) similarity is a common association metric used in object detection and image segmentation tasks. It measures the overlap between two bounding boxes or masks by calculating the ratio of their intersection to their union. SORT relies on a one-to-one association between detections and tracks by finding the minimal cost of association. This is also known as global nearest neighbor (GNN) in the field of multi-object tracking.

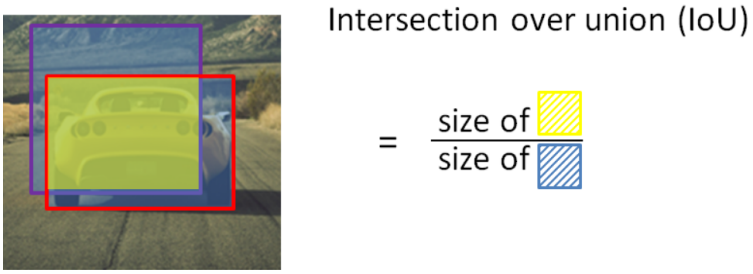

Figure 3: Intersection over Union

2.3.2 The Assignment Problem

The assignment problem is a classic optimization problem in computer science that involves finding the most efficient assignment of tasks to resources. Given a set of tasks and a set of resources, each with a cost or benefit assigned to it, the objective is to determine an optimal assignment that minimizes the total cost or maximizes the total benefit. The Hungarian algorithm is a well-known method for solving this problem optimally. It works by iteratively constructing a matrix of costs or benefits and then applying a series of operations to reduce the number of zeros in the matrix. The algorithm guarantees that it will find the optimal solution in polynomial time, making it a popular choice in many real-world applications, such as scheduling, vehicle routing, and resource allocation.

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

Figure 4: An Example for the Assignment Problem

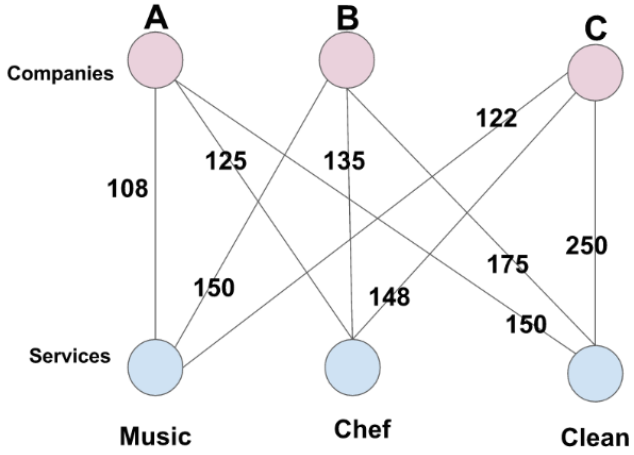

Figure 5: Graph Matching Approach

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

2.4 Track Management

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.

3 Performance Analysis

3.1 The CLEAR MOT Metrics

The CLEAR multi-object tracking metrics provide a standard set of tracking metrics to evaluate the quality of tracking algorithm. These metrics are popular for video-based tracking applications and they are as follows:

- 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

below is the evaluation of these metrics on our implementation:

```
>> SORT
      Tracker      MOTA (%)   MOTP (%)   Mostly Tracked (%)   Partially Tracked (%)   Mostly Lost (%)   False Positive   False Negative   Recall (%)
    ___________    ________   ________   __________________   _____________________   _______________   ______________   ______________   __________

    "ACF+SORT"      27.315     65.216          64.286                35.714                 0                290              172           73.457
    "YOLOv4+SORT"   82.099      90.48          78.571                14.286               7.1429             21               94           85.494
```

Figure 5: Evaluation Metrics 1

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

Figure 6: Evaluation Metrics 2

SORT achieves the highest MOTA score for the online trackers and is comparable to the state-of-the art methods, which are significantly more complex. Additionally, as SORT aims to focus on frame-to-frame associations the number of lost targets (ML) is minimal despite having similar false negatives to other trackers. Furthermore, since SORT focuses on frame-to-frame associations to grow tracklets, it has the lowest number of lost targets in comparison to the other methods.

3.2 Benefits & Shortcomings of SORT

SORT is extremely fast and efficient. It can handle high frame rates and is not computationally complex due to its simple linear motion model. But this leaves it vulnerable for bad performance in case of videos with low frame rates. Also, since it does not include methods to handle all edge cases and rely on simplicity, it suffers from relatively high identity switches. Especially with long-term occlusions.