# Mutli Object Tracking principles

Simon Lebeaud

GreenAI U.P.P.A

17-10-2022

# Glossary

**SOT** Single Object Tracking

**MOT** Multiple Object Tracking

**IoU** Intersection over Union

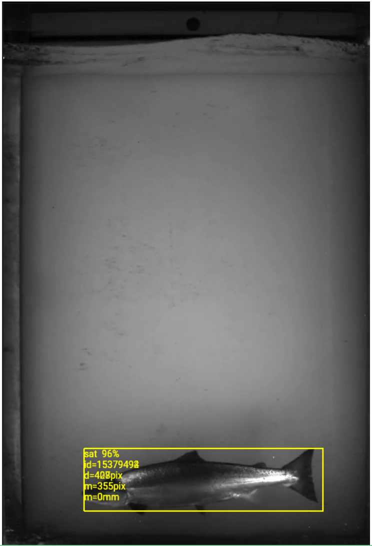
1. Motivations
2. Litterature
3. Conclusion

1. Motivations
   - Goal
   - Challenges
2. Litterature
3. Conclusion

# Goal

## Automatic fish count system

- Detection and classification
- Tracking for counting individuals



# What is Tracking ?

**Tracking** is the task of estimating or predicting the position of a **moving object** or image at any given time.
Steps of tracking are usually like so:

1. Know where the objects are

# What is Tracking ?

**Tracking** is the task of estimating or predicting the position of a **moving object** or image at any given time.
Steps of tracking are usually like so:

1. Know where the objects are
2. Assign unique id to each object

## What is Tracking ?

**Tracking** is the task of estimating or predicting the position of a **moving object** or image at any given time.
Steps of tracking are usually like so:

1. Know where the objects are
2. Assign unique id to each object
3. Estimate where the object is in the next frame

# Tracking types

**Two** types of tracking :

# Tracking types

**Two** types of tracking :

## Image Tracking

Tracking of a **moving image** in a video in order to superimpose content onto it (Augmented reality)

# Tracking types

**Two** types of tracking :

## Image Tracking

Tracking of a **moving image** in a video in order to superimpose content onto it (Augmented reality)

## Vidéo Tracking

Tracking of a **moving object** in a video :

Establish a relationship between current detected objects and previous ones.

# Object Tracking types

**Two** types of object tracking :

## Object Tracking types

**Two** types of object tracking :

### SOT

Have only **one object of interest**, always in the images.
Mostly based on sophisticated models to deal with scale, rotation and illumination variation.

## Object Tracking types

**Two** types of object tracking :

### SOT

Have only **one object of interest**, always in the images.
Mostly based on sophisticated models to deal with scale, rotation and illumination variation.

### MOT

Multiple object with similar appearances and geometries.
Detection $\rightarrow$ Identification paradigm.

1. Motivations
   - Goal
   - Challenges
2. Litterature
3. Conclusion

## Challenges

- Frequent occlusions,

## Challenges

- Frequent occlusions,
- Initialization and termination of tracks,

## Challenges

- Frequent occlusions,
- Initialization and termination of tracks,
- Appearance similarity,

## Challenges

- Frequent occlusions,
- Initialization and termination of tracks,
- Appearance similarity,
- interactions between objects

# Euclidean distance tracking

Utilize the distance between the centroids of the objects detected between two consecutive frames in a video.

Step 1 Objects are detected, get bounding boxs and calculate their centroid for **t-1** and assign IDs

Step 2 Detect object and their centroid for **t**

Step 3 Calculate euclidean distance between all objects of **t-1** and **t**

Step 4 Distance smaller than a treshold then two objects are the same and we attribute ID

Step 5 Distance greater than a treshold, add a **new ID**

Step 6 If no object attributed to ID, remove the ID

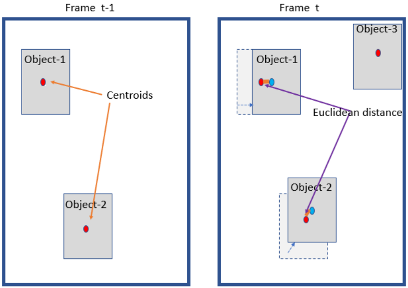
# Euclidean distance tracking

## Pros

- Low computation cost.

## Cons

- Works better at high FPS,
- ID switch for close objects,
- Not resilient to occlusion.

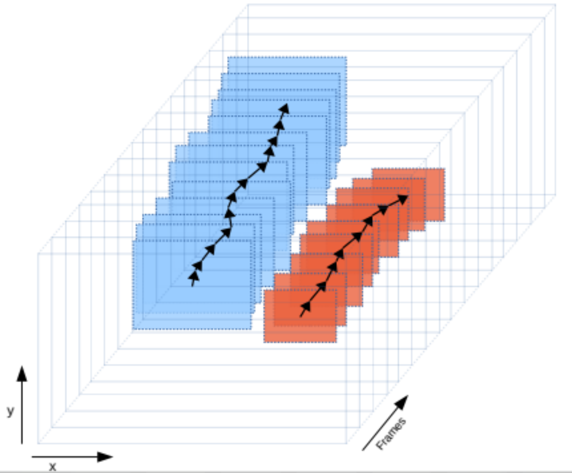


1 Motivations

2 Litterature
- Euclidean distance
- Intersection over Union
- SORT
- DeepSORT
- FairMOT
- ByteTrack

3 Conclusion

# IoU Tracker



1. Motivations

2. Litterature
   - Euclidean distance
   - Intersection over Union
   - SORT
   - DeepSORT
   - FairMOT
   - ByteTrack

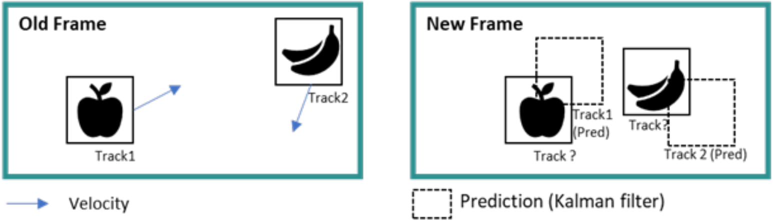
3. Conclusion

## Simple Online and Realtime Tracking [Bewley et al., 2016]

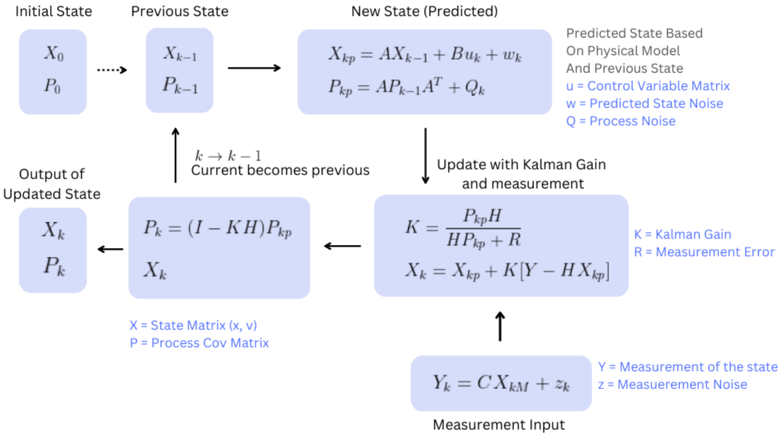
**SORT :** Three Stage Tracking

| | |
|---|---|
| **Detection** | How to identify individuals objects/tracks? |
| **Kalman Filter** | How to predict where track will move? |
| **Hungarian Method** | What track do the detected objects belong to |

# Estimation Model - Kalman Filter [Kalman, 1960]



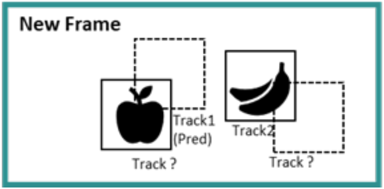
# Estimation Model - Kalman Filter



# Data Association - Hungarian [Kuhn, 1955]

Also called the Kuhn-Munkres algorithm, this method is used to solve the **association problem**. Given a adjacency matrix (of IoU in this case) we can find the optimal assignment such that we minimise the total cost, i.e find the correct associations with only $O(n^3)$ complexity.

# Data Association - Hungarian




| | Apple (new frame) | Banana (new frame) |
|---|---|---|
| **Track1 (Pred)** | 0.2 | 0 |
| **Track2 (Pred)** | 0 | 0.15 |

| | Apple (new frame) | Banana (new frame) |
|---|---|---|
| **Track1 (Pred)** | 0.2 | 0 |
| **Track2 (Pred)** | 0 | 0.15 |

# SORT

## Pros

- Low computation cost.

## Cons

- Very dependant on detection
- Lots of ID switch for similar objects,
- Not resilient to occlusion.
- No Re-id

# DeepSort[Wojke et al., 2017]

Same base as SORT :

- Estimation with Kalman filter

But add **two** different metrics:

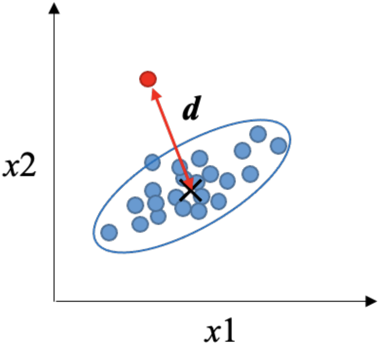
1. Mahalanobis distance
2. Deep appearance cosine distance

# Squared Mahalanobis distance

Compute the **distance between** predicted Kalman states (**distribution**) and newly arrived measurements (**points**)

$$d^{(1)}(i,j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i)$$

Distribution denoted by $(y_i, S_i)$ and bounding box denoted by $d_j$



## Assignment Problem

**Mahalanobis Distance :**

$$d^{(1)}(i,j) = (d_j - y_i)^T S_i^{-1}(d_j - y_i) \quad (1)$$

Smallest **cosine distance** between the $i$-th track and $j$-th detection in **appearance space** :

$$d^{(2)}(i,j) = min\{1(r_j)^\top r_k^{(i)} | r_k^{(i)} R_i\} \quad (2)$$

**Combination** of the two with weighted sum

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1\lambda) d^{(2)}(i,j) \quad (3)$$

# Matching Cascade

## Motivations

When objects are occluded for a long period of time, the **location uncertainty increases**.

When two track compete, Mahalanobis distance only chooses the closer target.

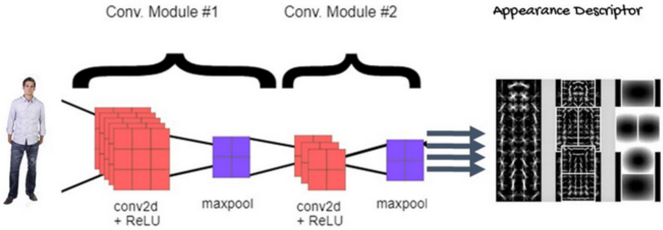
Therefore we need to **give priority** to more frequently seen objects.

**Listing 1** Matching Cascade

**Input:** Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

Compute cost matrix $\boldsymbol{C} = [c_{i,j}]$ using Eq. 5
Compute gate matrix $\boldsymbol{B} = [b_{i,j}]$ using Eq. 6
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
**for** $n \in \{1, \ldots, A_{\max}\}$ **do**
  Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
  $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(\boldsymbol{C}, \mathcal{T}_n, \mathcal{U})$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
**end for**
**return** $\mathcal{M}, \mathcal{U}$

# Deep Appearance Descriptor




| Name | Patch Size/Stride | Output Size |
|---|---|---|
| Conv 1 | $3 \times 3/1$ | $32 \times 128 \times 64$ |
| Conv 2 | $3 \times 3/1$ | $32 \times 128 \times 64$ |
| Max Pool 3 | $3 \times 3/2$ | $32 \times 64 \times 32$ |
| Residual 4 | $3 \times 3/1$ | $32 \times 64 \times 32$ |
| Residual 5 | $3 \times 3/1$ | $32 \times 64 \times 32$ |
| Residual 6 | $3 \times 3/2$ | $64 \times 32 \times 16$ |
| Residual 7 | $3 \times 3/1$ | $64 \times 32 \times 16$ |
| Residual 8 | $3 \times 3/2$ | $128 \times 16 \times 8$ |
| Residual 9 | $3 \times 3/1$ | $128 \times 16 \times 8$ |
| Dense 10 | | 128 |
| Batch and $\ell_2$ normalization | | 128 |

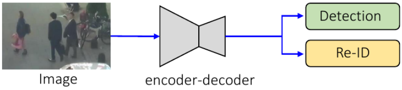
# Problem with current One Shot Trackers

## CenterTrack [Zhou et al., 2020]


Image $I^{(t)}$ Image $I^{(t-1)}$ Tracks $T^{(t-1)}$ Detections $\hat{Y}^{(t)}$ Size $\hat{S}^{(t)}$ Offset $\hat{O}^{(t)}$

**Problem :** There is no re-id so if the object disappear, the tracking also.

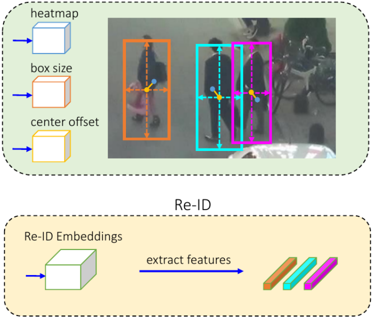
# FairMOT [Zhang et al., 2021b]



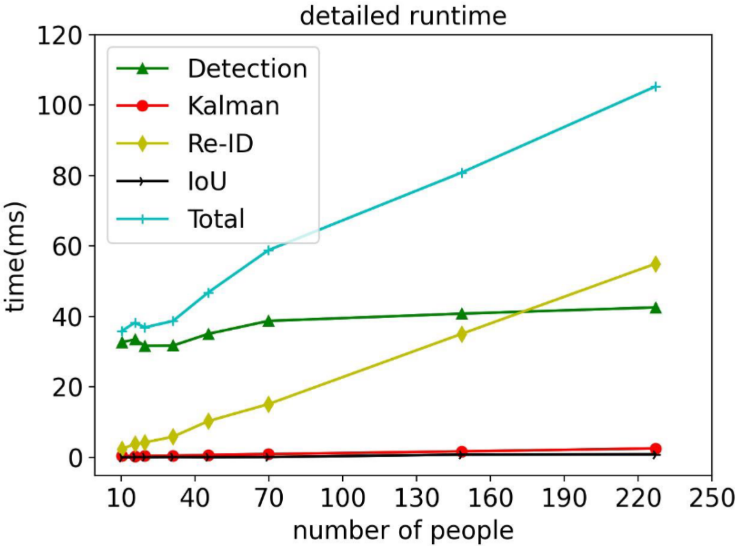

## Re-ID

Extract low level feature to be compared with new box features.

Keep feature maps in memory to counter occlusion.



# FairMOT


detailed runtime
Detection
Kalman
Re-ID
IoU
Total
time(ms)
0
20
40
60
80
100
120
10
40
70
100
130
160
190
220
250
number of people

## Why we don't want fairMOT

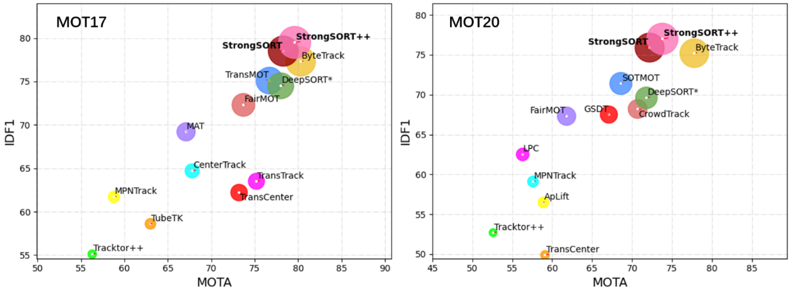
Only good for autonomus vehicules where objects are easy to discriminated

# ByteTrack[Zhang et al., 2021a]

**Algorithm 1:** Pseudo-code of BYTE.

```
Input: A video sequence V; object detector Det; detection score
       threshold τ
Output: Tracks 𝒯 of the video
Initialization: 𝒯 ← ∅
for frame f_k in V do
    /* Figure 2(a) */
    /* predict detection boxes & scores */
    𝒟_k ← Det(f_k)
    𝒟_high ← ∅
    𝒟_low ← ∅
    for d in 𝒟_k do
        if d.score > τ then
            𝒟_high ← 𝒟_high ∪ {d}
        end
        else
            𝒟_low ← 𝒟_low ∪ {d}
        end
    end

    /* predict new locations of tracks */
    for t in 𝒯 do
        t ← KalmanFilter(t)
    end

    /* Figure 2(b) */
    /* first association */
    Associate 𝒯 and 𝒟_high using Similarity#1
    𝒟_remain ← remaining object boxes from 𝒟_high
    𝒯_remain ← remaining tracks from 𝒯

    /* Figure 2(c) */
    /* second association */
    Associate 𝒯_remain and 𝒟_low using similarity#2
    𝒯_re-remain ← remaining tracks from 𝒯_remain

    /* delete unmatched tracks */
    𝒯 ← 𝒯 \ 𝒯_re-remain

    /* initialize new tracks */
    for d in 𝒟_remain do
        𝒯 ← 𝒯 ∪ {d}
    end
end
Return: 𝒯
```

# Benchmark [Du et al., 2022]



## What's next

- Keep update the fish database
- Continue improve the detection model,
- Test tracking techniques on fish database,

*Thanks!*

# References I

[Bewley et al., 2016] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016).
Simple online and realtime tracking.

[Du et al., 2022] Du, Y., Song, Y., Yang, B., and Zhao, Y. (2022).
Strongsort: Make deepsort great again.

[Kalman, 1960] Kalman, R. E. (1960).
A new approach to linear filtering and prediction problems.

[Kuhn, 1955] Kuhn, H. W. (1955).
The Hungarian Method for the Assignment Problem.

# References II

[Wojke et al., 2017] Wojke, N., Bewley, A., and Paulus, D. (2017).
Simple online and realtime tracking with a deep association metric.

[Zhang et al., 2021a] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2021a).
Bytetrack: Multi-object tracking by associating every detection box.

[Zhang et al., 2021b] Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2021b).
FairMOT: On the fairness of detection and re-identification in multiple object tracking.

## References III

[Zhou et al., 2020] Zhou, X., Koltun, V., and Krähenbühl, P. (2020).
Tracking objects as points.