Progress report

# Spatio-Temporal Association

5th stage
🫢

# Content

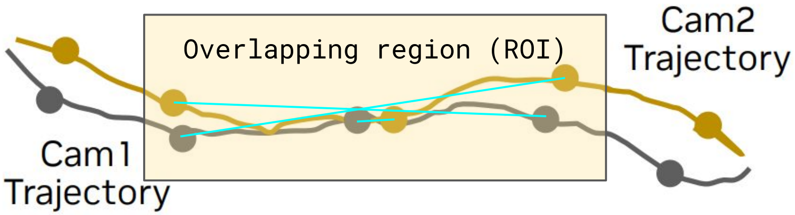
1. Recap
2. Mapping strategy
3. Experimental results
4. Error analysis
5. Addressing SCT issues

# 1. Recap

# 1.1. Previous stage

Track-level mapping:

1. Infer foot point from bounding box
2. Perspective transform to the cam 2's view
3. For each pair of 2 tracks, sample corresponding foot points by
    - ROI
    - Time
4. Calculate cost for the pair: average of Euclid distances
5. Hungarian assignment



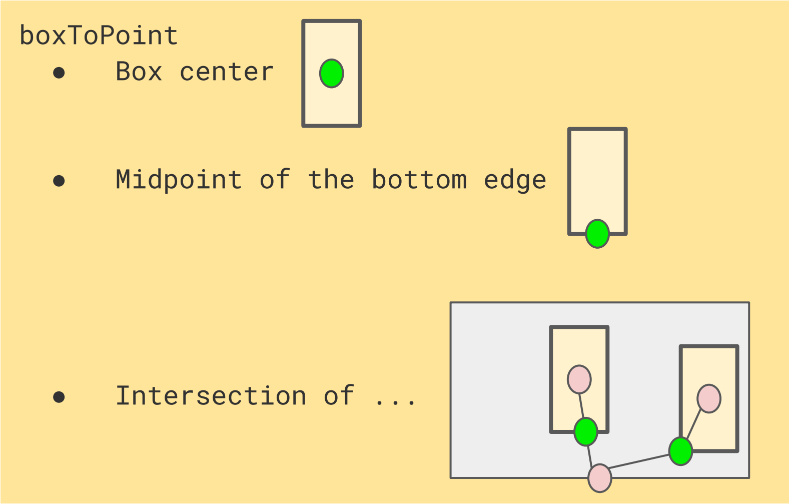
Bounding box is manually drawn, so the Precision and Recall are 100%. Mapping between tracks is 1-to-1.

Error Analysis on frame-level matching indicates that: Most of the wrong frame-level matching is due to bad detection results:

- Box was missing.
- Box did not fit the object well.
- Derived foot point was not precise.

Scene bias:

- Having a large overlapping area
- Near camera
- Period of time a person moving in the scene was long.

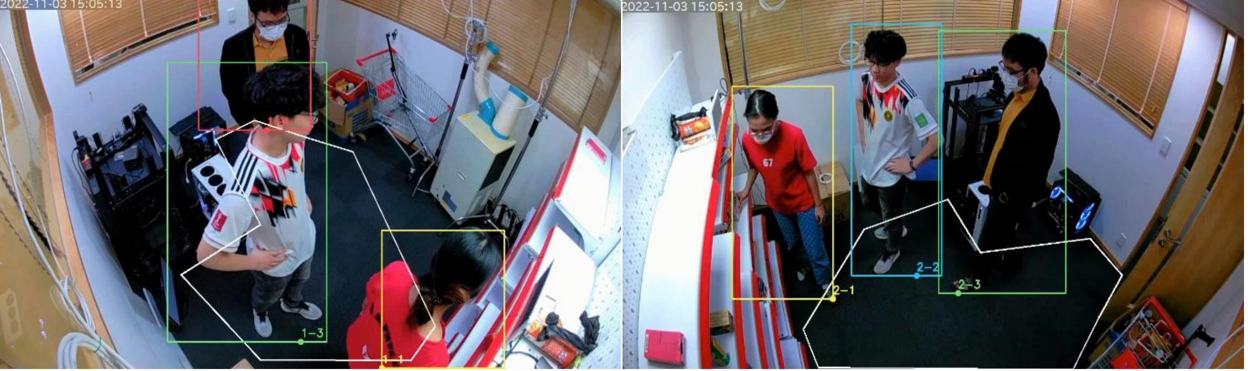


boxToPoint

- Box center
- Midpoint of the bottom edge
- Intersection of ...

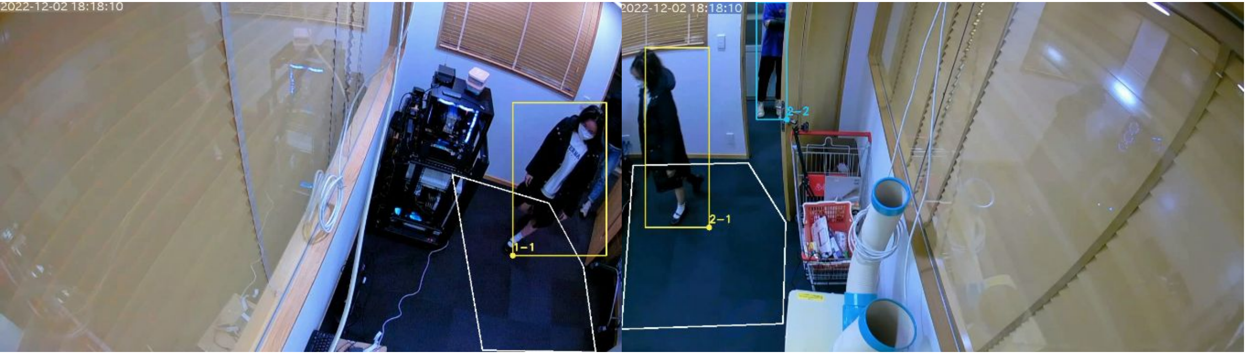
# 1.2. Target

1. Map track to track between 2 camera using tracker’s prediction (instead of manual labels)
2. Record more challenging videos
3. Indicate issues of the approaches in frame-level
4. Address the issues of SCT



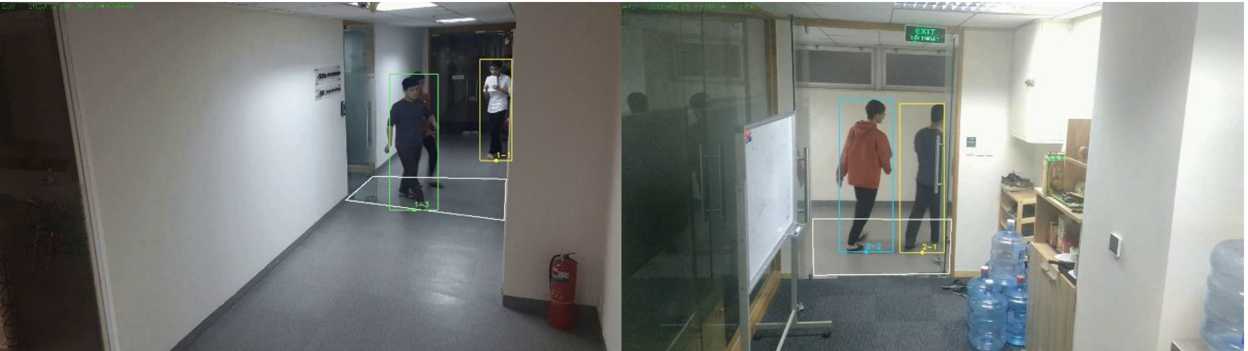
Video set 1:

- Close to the camera
- Camera angle is steep
- Large overlapping area



Video set 2:

- Close to the camera
- Camera angle is steep
- Small overlapping area



Video set 3:

- Far from the camera
- Nearly horizontal camera angle
- Small overlapping area

# 2. Mapping strategy

# 2.1. Experimental procedure

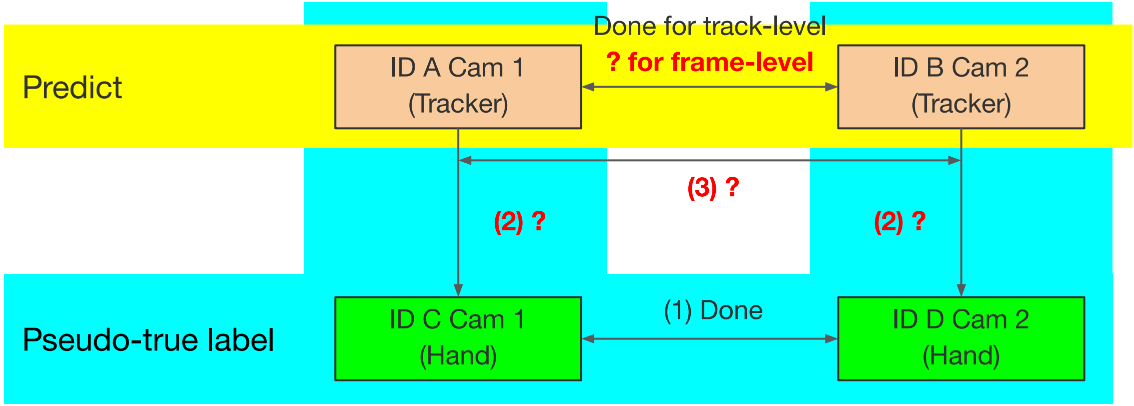
2 tasks:

1. Predict mapping between different cameras. E.g A ←→ B
2. Make ground truth in order to evaluate that mapping. Either
   - Make by hand ⇒ Impossible, due to **fragmented and swapping tracks**, and **tracking algorithms' performance**
   - Generate pseudo ground truth:

     Step (1): Make SCT ground truth for each video, e.g make C, D and map C ←→ D ⇒ No ID switch.

     Step (2): Within a camera, map prediction of tracker to ground truth from step 1, e.g A ←→ C

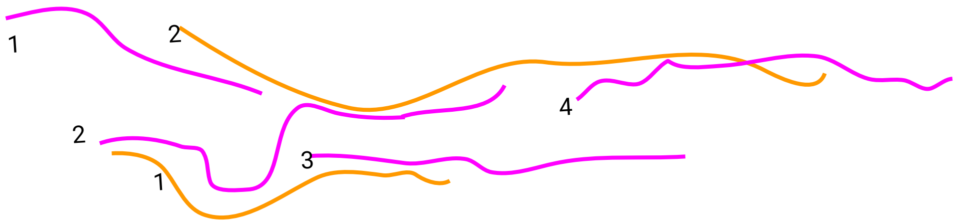
     Step (3): Generate pseudo ground truth of MCT mapping, e.g A → C → D → B, so that we map A ←→ B



# 2.2. Frame-level mapping

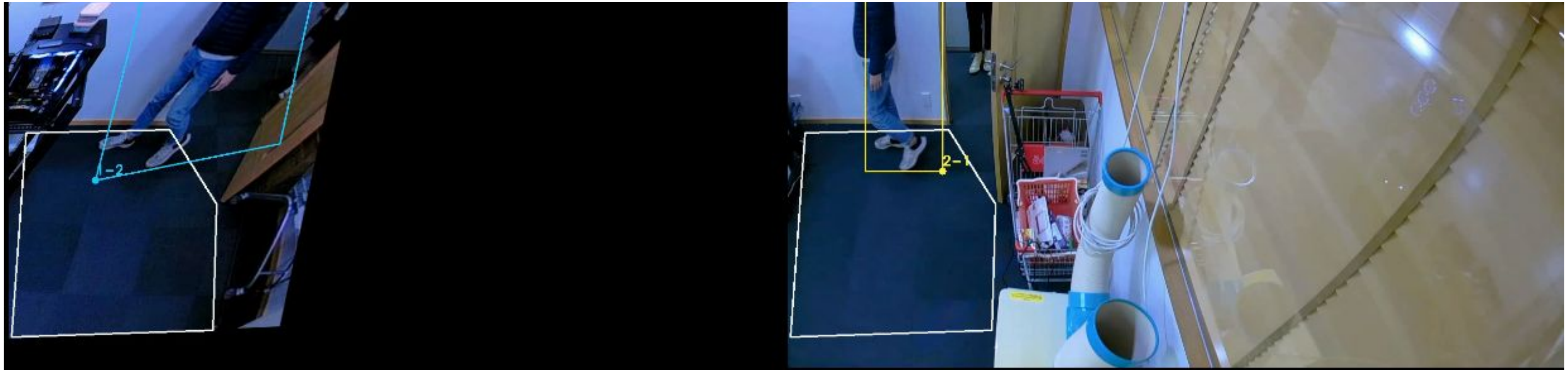
Motivation to start with frame-level mapping:

1. Can produce track-level mapping later on.
2. Can detect the problem of SCT: ID switch (include fragmented track and swapping track)
3. Avoid some problem of track-level mapping
    a. Confusing circumstance (figure below)
    b. Unreliable evaluation result (due to numerous ID switches).



# 2.2. Frame-level mapping

Problem (2) with track-level mapping: Unreliable evaluation result.

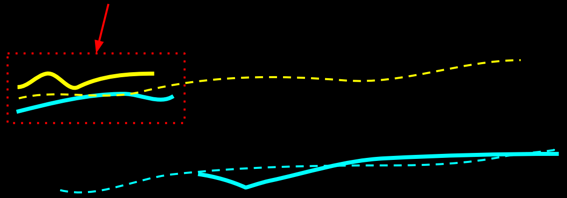


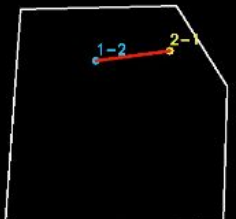
Cam 1: ID 2 was on Mr. Hoang (short), but then switched to Mrs. Thao (long), so it was assigned to Mrs. Thao

Cam 2: ID 1 was assigned to Mr. Hoang

⇒ Those IDs were never paired in pseudo ground truth.

⇒ This part was counted as FP in frame-level error analysis.





| | |
|---|---|
| TP | 74 |
| FP | 63 |
| FN | 7 |
| Precision | 0.54 |
| Recall | 0.91 |
| F1 | 0.67 |

# 2.3. Implementation

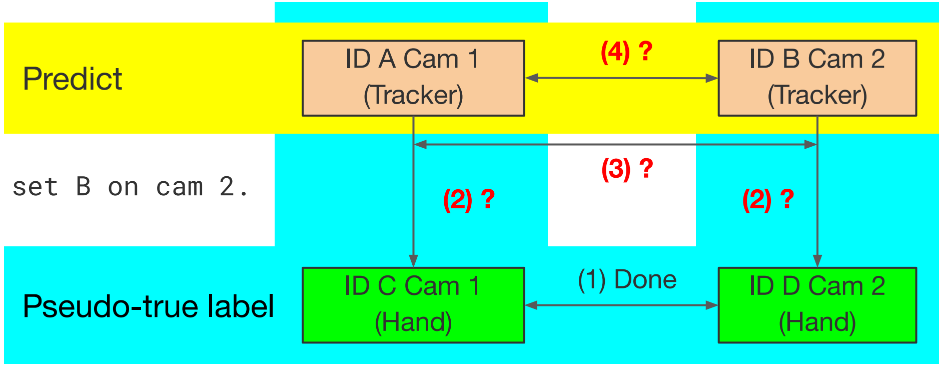
### (2) Mapping (frame-level) SCT tracker's prediction to ground truth within a camera

```
For each frame:
 1.   Find the present true ID and present predicted ID.
 2.   Build cost matrix using Euclid distance (IoU distance was not good in experiments).
 3.   Produce pairs with Hungarian.
```

Observe on 2 videos: (FP, FN) is (5/235, 0/235) and (4/305, 0/235) respectively.

### (3) Create pseudo ground truth (frame-level) for MCT

```
For each frame f:
 1.   Find the present predicted ID set A on cam 1 and set B on cam 2.
 2.   Consider each a in A and each b in B:
     2.1.   c = true_ID_of(a, f)
     2.2.   d = true_ID_of(b, f)
     2.3.   If c maps to d then: map a to b
```

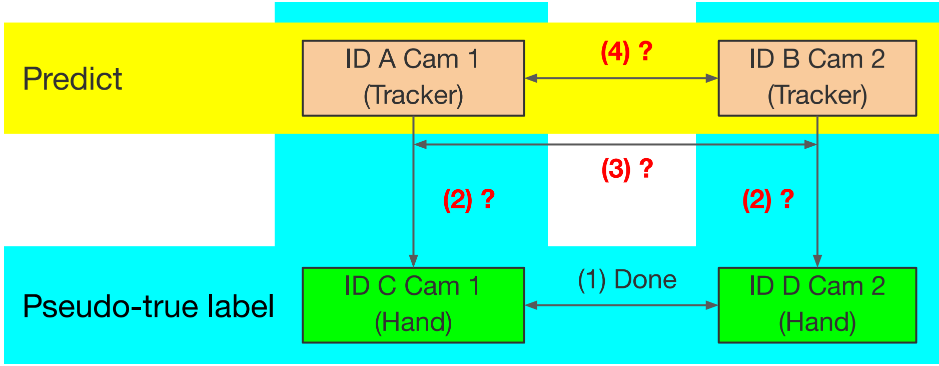


Observe on a pair of video: FP = 7/200, FN = 22/200

# 2.3. Implementation

### (4) Mapping (frame-level) MCT tracks

```
For each frame:
 1.   Find the present true ID and present predicted ID.
 2.   Derive the foot point of each bounding box.
 3.   Build cost matrix using Euclid distance.
 4.   Produce pairs with Hungarian.
```



# 3. Experimental results

# 3.1. Quick review: Detector & Tracker

## Detector

YOLOv5:

- Backbone: CSPResBlock reduce FLOPs while yield more informative gradient
- Neck: PANet allow propagation of information between detection layers more efficient
- Augmentation: Scaling, Color adjustment, Mosaic
- Auto learn anchor boxes

YOLOX:

- Decoupled head: Split into a Classification and Regression branches to avoid conflict.
- Strong data augmentation: MixUp and Mosaic (turned off for the last epoches).
- Anchor-free
- Multiple positives to reduce the imbalance between positives and negatives when training.
- SimOTA: deal with ambiguity when assigning labels to anchors

YOLOv8:

- Replace C3 block (a specific type of CSP block) to with C2f block
- Anchor-free
- Turn off Mosaic for the last epoches

# 3.1. Quick review: Detector & Tracker

## Tracker

ByteTrack:

- Bounding boxes with low detection scores, e.g occluded objects, are not thrown, but also get associated
- To filter out irrelevant low score boxes, motion similarity with current tracks is used.
- ⇒ detection threshold is less sensitive in ByteTrack as compared to SORT.

DeepSORT:

- Motion distance as a threshold: Mahalanobis distance of the mean track location and a newly arrived detection box.
- Appearance distance as a cost: distance between a newly arrived detection box to a track is the distance to the closest visual feature of the feature bank.
- Matching cascade: give higher priority to the more frequently seen track.

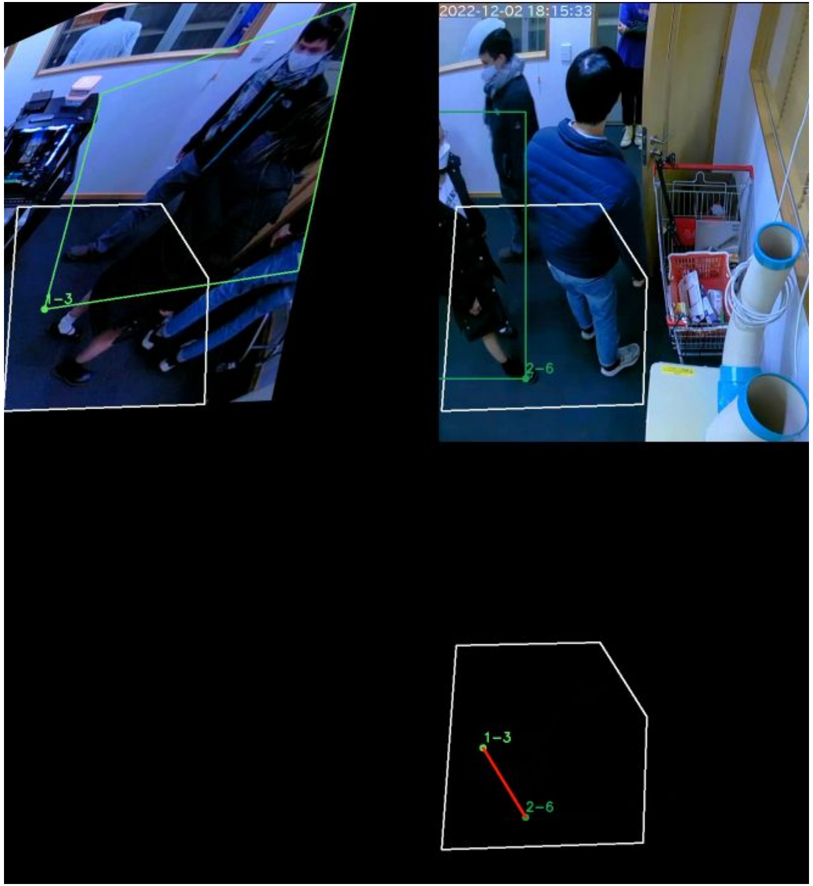
StrongSORT:

- Replace feature bank in DeepSORT with a moving visual feature.
- Account for object confident score in the Kalman filter noise covariance.
- Replace matching cascade in DeepSORT with previous assignment strategy.
- Adopt ECC for camera motion.
- Introduce AFLink to associate short tracklets, GSI to predict missing detection

# 3.2. Quantitative results

| *GPU: NVIDIA RTX 3090*<br>*CPU: Intel i7-11700K 3.60GHz* | FPS | | | HOTA | | | MCT (w/o GMM) | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Detector** | **Tracker** | **Combine** | **DetA** | **AssA** | **HOTA** | **P** | **R** | **F1** |
| YOLOv5l_pretrained-640-ByteTrack | 118.5 - | 1478.4 - | 109.7 - | 61.60 | 45.89 | 52.41 | 0.614 | 0.940 | 0.739 |
| YOLOXs_pretrained-640-ByteTrack | 107.8 - | | 100.5 - | 57.83 | 44.47 | 50.21 | 0.616 | 0.929 | 0.736 |
| YOLOXm_pretrained-640-ByteTrack | 88.9 - | | 82.8 - | 60.79 | 44.96 | 51.62 | 0.624 | 0.934 | 0.745 |
| YOLOXl_pretrained-640-ByteTrack | 74.5 - | | 70.8 - | 60.99 | 44.85 | 51.56 | 0.621 | 0.933 | 0.741 |
| YOLOv8s_pretrained-640-ByteTrack | 164.7 - | | 148.5 - | 57.22 | 44.26 | 49.95 | 0.603 | 0.940 | 0.728 |
| YOLOv8m_pretrained-640-ByteTrack | 131.8 - | | 122.2 - | 61.56 | 47.08 | 53.05 | 0.625 | 945 | 0.747 |
| YOLOv8l_pretrained-640-ByteTrack | 110.3 - | | 103.5 - | 62.37 | 46.95 | 53.28 | 0.628 | 0.947 | 0.750 |
| YOLOv8l_pretrained-640-StrongSORT | | 74.8 - | 44.1 - | 67.89 | 52.49 | **58.92** | 0.594 | 0.946 | 0.725 |

# 3.3. Extension: FP elimination

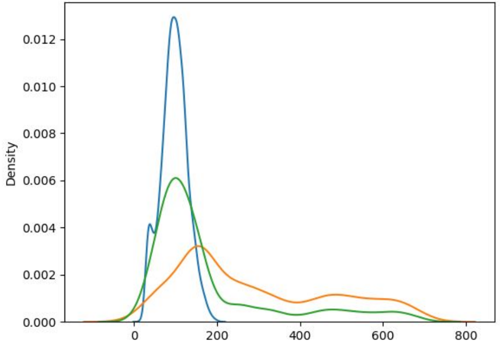


False Positive pairs can be due to missing detection.

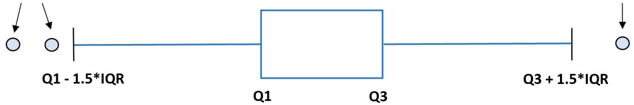
**Assumption**: FP pairs due to missing detection has larger distance than other pairs ⇒ treat as outliers.

Approaches: Do mapping along the frame axis as previous, store the distance of the **matched** pairs. Base on the above assumption, the pairs whose distance much greater than the average are the FP pairs.

- Fit a 2-component GMM, take the component with a smaller mean, then eliminate the pairs whose distance > mean + 3 * std



- Calculate Interquartile range, then eliminate the pairs whose distance > Q3 + 1.5 * IQR



# 3.3. Extension: FP elimination

| *YOLOv8l_pretrained-640-ByteTrack* | without elimination | | | GMM | | | IQR | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Video set 1 | 0.566 | 0.937 | 0.704 | 0.575 | 0.862 | 0.679 | 0.579 | 0.929 | 0.712 |
| Video set 2 | 0.645 | 0.958 | 0.769 | 0.725 | 0.935 | 0.816 | 0.664 | 0.958 | 0.782 |
| Video set 3 | 0.703 | 0.954 | 0.801 | 0.771 | 0.900 | 0.822 | 0.732 | 0.954 | 0.821 |

| *YOLOv8l_pretrained-640-StrongSORT* | without elimination | | | GMM | | | IQR | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Video set 1 | 0.562 | 0.941 | 0.703 | 0.588 | 0.819 | 0.666 | 0.573 | 0.933 | 0.709 |
| Video set 2 | 0.637 | 0.962 | 0.765 | 0.724 | 0.941 | 0.817 | 0.681 | 0.962 | 0.797 |
| Video set 3 | 0.615 | 0.943 | 0.735 | 0.680 | 0.875 | 0.753 | 0.631 | 0.941 | 0.747 |

# 3.3. Extension: FP elimination

Analyze YOLOv8l_pretrained-640-ByteTrack on Video set 1: 11/16 videos got better when applied GMM elimination.

Worse case (video 12) in the set:

TP: 55

FP: 90

FN: 407

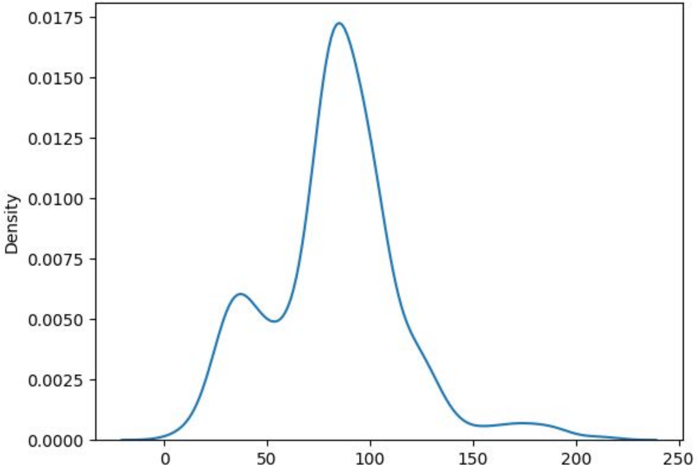
Pre: 0.3793103448275862

Rec: 0.11904761904761904

F1: 0.18121911037891267

mean = [33.80373621] and std = [6.7026928]

⇒ IQR is a safer choice.

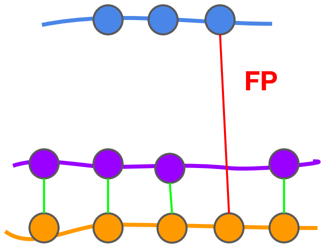


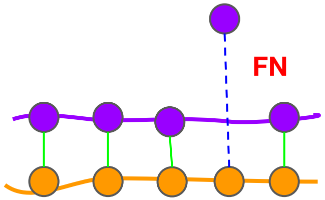
# 3.4. Extension: Cost by window

Motivation:

- Support IQR elimination in case of missing boxes
- Compensate for mis-located box or foot point due to partial occlusion, etc.

⇒ Still match in frame-level, but the cost for each pair is calculated within a window centered by the current frame.





# 3.3. Extension: Cost by window

| *YOLOv8l_pretrained-640-ByteTrack* | without window | | | window_size = 11 | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| Video set 1 | 0.566 | 0.937 | 0.704 | 0.567 | 0.939 | 0.706 |
| Video set 2 | 0.645 | 0.958 | 0.769 | 0.642 | 0.954 | 0.767 |
| Video set 3 | 0.703 | 0.954 | 0.801 | 0.703 | 0.950 | 0.800 |

| *YOLOv8l_pretrained-640-StrongSORT* | without window | | | window_size = 11 | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| Video set 1 | 0.562 | 0.941 | 0.703 | 0.564 | 0.943 | 0.704 |
| Video set 2 | 0.637 | 0.962 | 0.765 | 0.644 | 0.972 | 0.773 |
| Video set 3 | 0.615 | 0.943 | 0.735 | 0.621 | 0.946 | 0.740 |

# 4. Error analysis

# 3.3. Error analysis

**Limitation of Detection:**

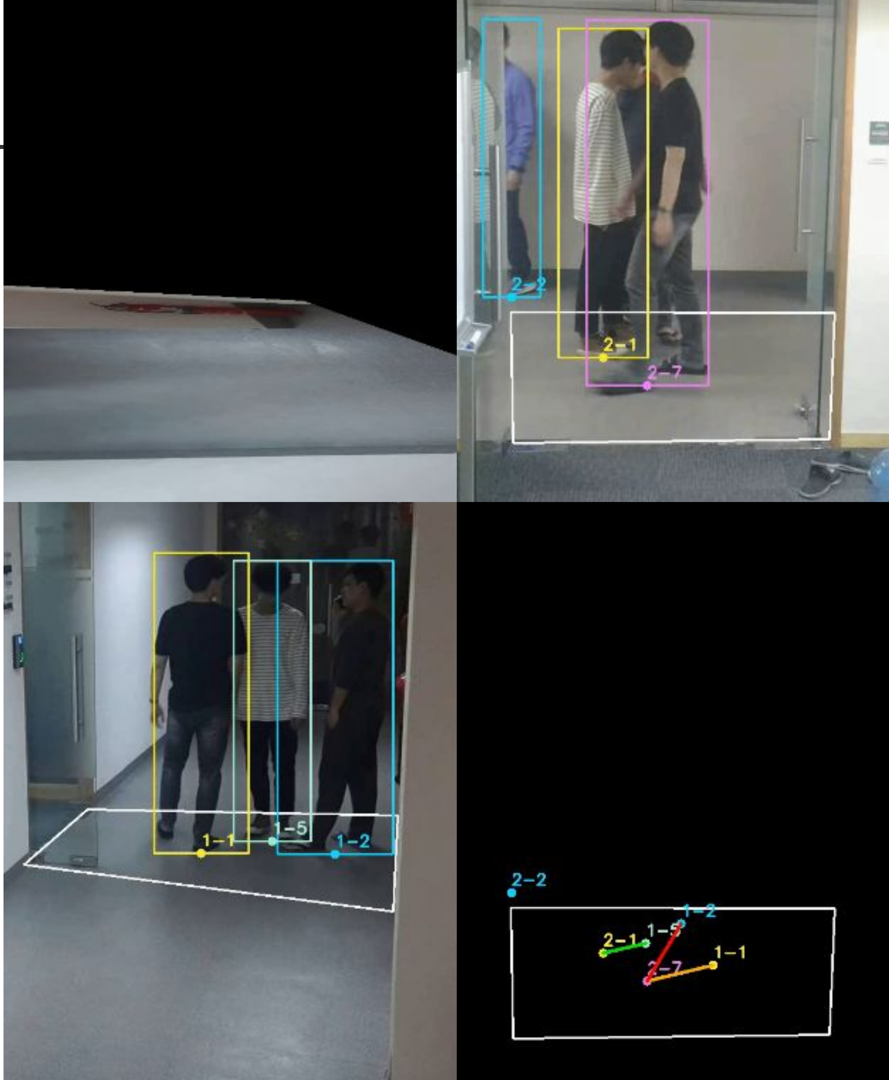
1. Wrong box location or missing detected boxes (Either occlusion or domain-shift)
2. Estimated foot is still not totally correct.

**Limitation of Homography:**

1. Depends heavily on the accuracy of the footing point
2. Overlapping area between cameras (large and small)

Improvement:

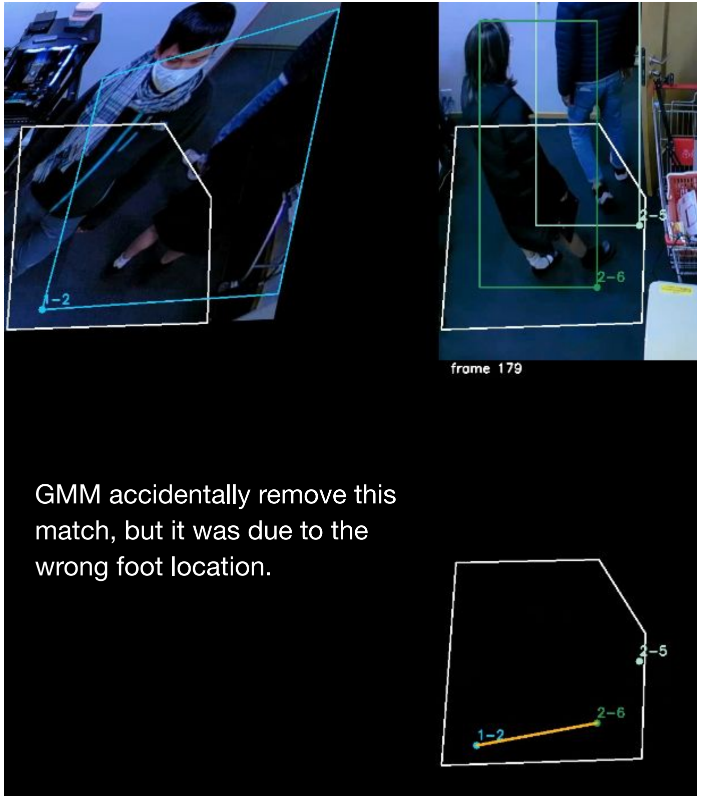
1. Replace Detection by Pose estimation
2. Research and experiment other approach to replace homography.



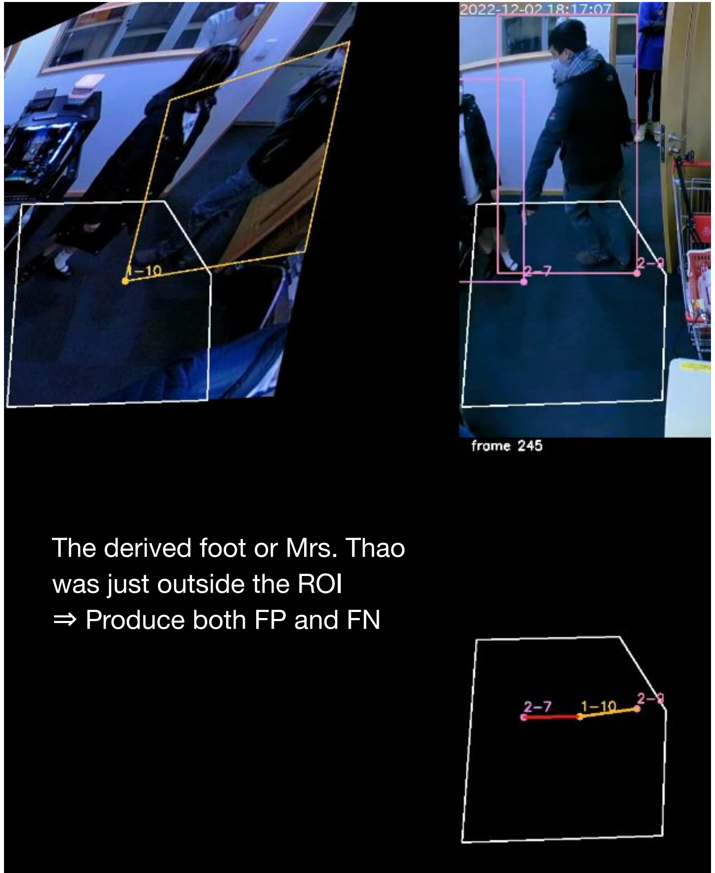
# 3.3. Error analysis

Drawbacks by detection:

1. Imprecise box location or missing boxes (Either occlusion or domain-shift)
2. Foot derived from bound box was not precise.



GMM accidentally remove this match, but it was due to the wrong foot location.



The derived foot or Mrs. Thao was just outside the ROI
⇒ Produce both FP and FN

# 5. Addressing SCT issues

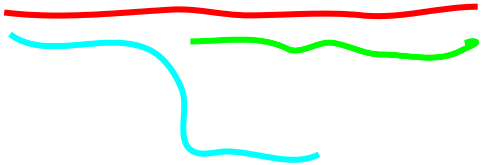
# 5.1. Detect SCT issues

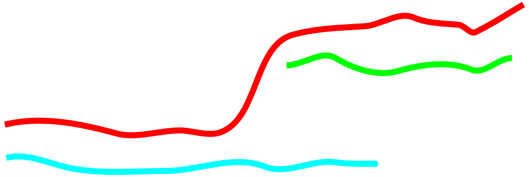
**Swapping tracks** : a track traces 2 different people

Given track A from cam 1, track B and C from cam 2. If

- A and B are mapped at some frames, and points to the same person
- A and C are mapped at some frames, and points to the same person
- A, B, C co-occur at some frames

Then the object swap happened with at least one of A, B, or C. However, we do not know exactly which of those tracks was swapped?





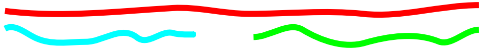
**Fragmented tracks**: a person is traced by 2 different tracks

Assumption: there is no object swap any tracks, i.e object swap is solved

Given track A from cam 1, track B and C from cam 2.

- A and B are mapped at some frames, and points to the same person
- A and C are mapped at some frames, and points to the same person

Then B, C can be merged.



# 5.1. Detect SCT issues

**2d_v2 Video 19**

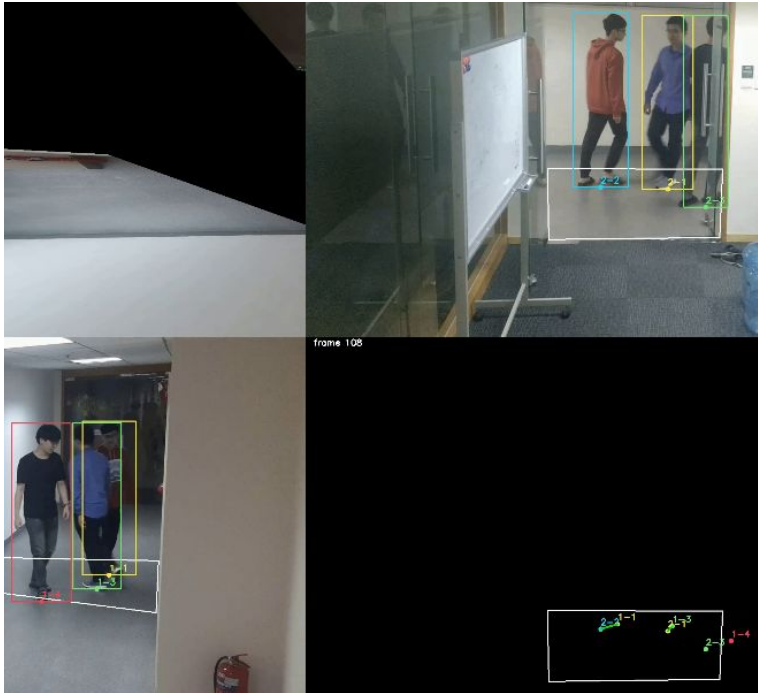
YOLOv8l_pretrained-640-ByteTrack
IQR (25, 75, 1.5) (P: 0.681, R: 0.944) ⇒ T/F = 3/8
IQR (30, 70, 1.5) (P: 0.693, R: 0.939) ⇒ T/F = 3/5
IQR (35, 65, 1.5) (P: 0.703, R: 0.906) ⇒ T/F = 3/3
IQR (35, 65, 1.5) (P: 0.704, R: 0.867) ⇒ T/F = 3/2

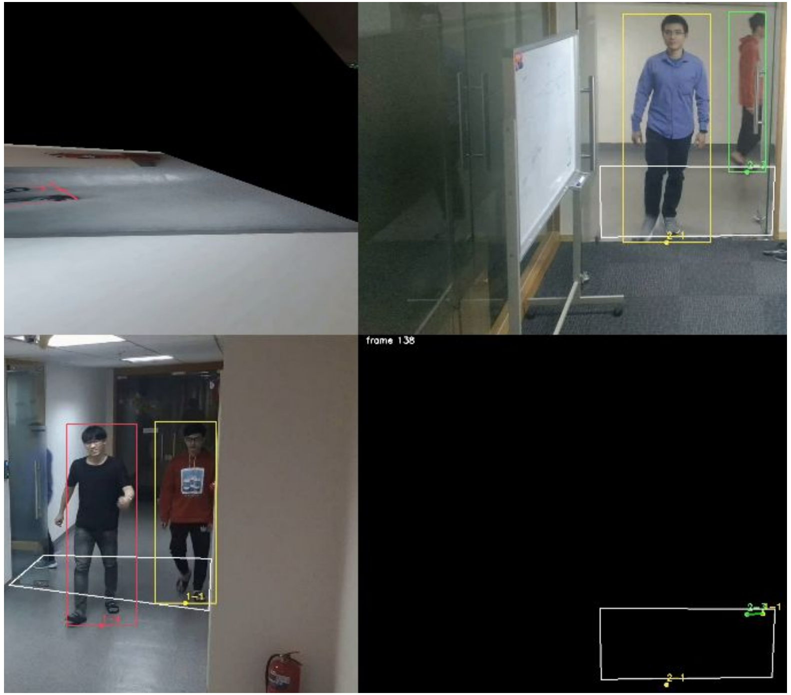
**2d_v3 Video 12**

YOLOv8l_pretrained-640-ByteTrack
IQR (25, 75, 1.5) (P: 0.827, R: 0.923) ⇒ T/F = 4/20
IQR (30, 70, 1.5) (P: 0.829, R: 0.916) ⇒ T/F = 4/20
IQR (35, 65, 1.5) (P: 0.836, R: 0.905) ⇒ T/F = 4/20
IQR (40, 60, 1.5) (P: 0.848, R: 0.894) ⇒ T/F = 4/13
IQR (45, 55, 1.5) (P: 0.848, R: 0.692) ⇒ T/F = 4/13

- All of the false detection is due to wrong mapping, not in the procedure.
- When the threshold used to eliminate FP mapping decreased, the number of false detection also decreased.

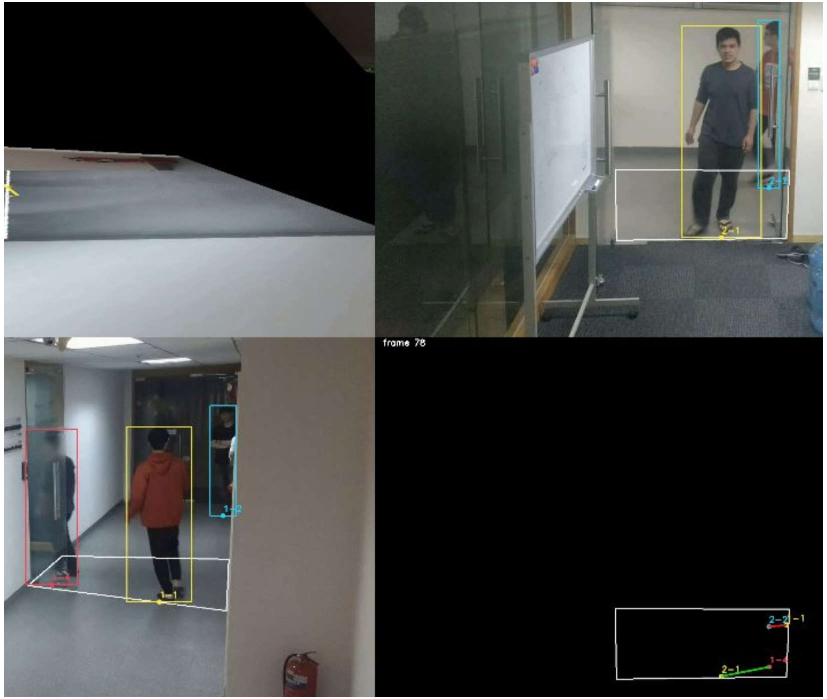
# 5.1. Detect SCT issues

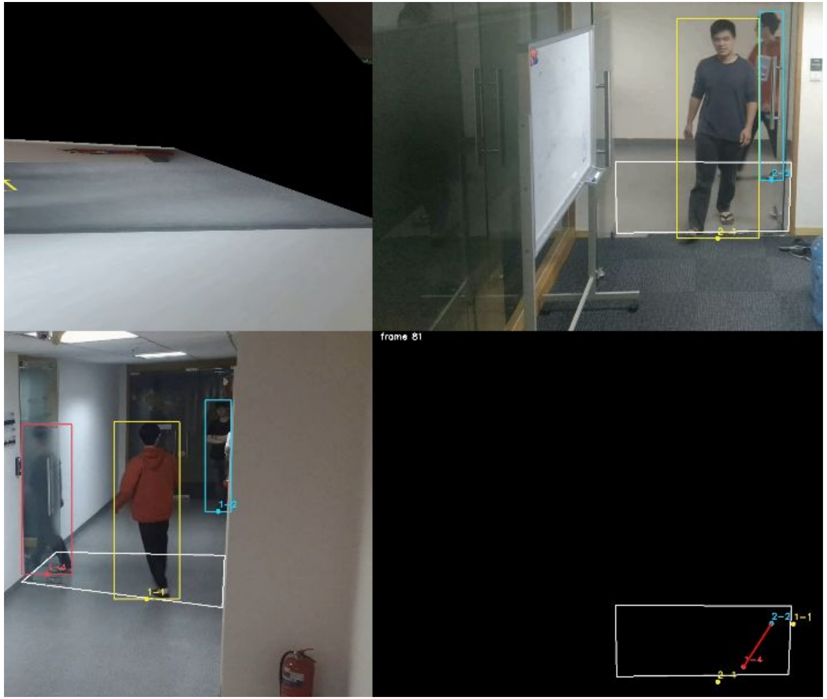




Video 3
CAM_1 ID **1** (Mr. Vuong) maps to **3** (Mr. Phong) and **2** (Mr. Vuong) while they co-occur at frame 108
switched from **2** (Mr. Vuong) (at frame 108) to **3** (Mr. Vuong) (at frame 138)
⇒ Correct detection. Cam2 ID 3 switched from Mr. Phong to Mr. Vuong

# 5.1. Detect SCT issues





Video 2
CAM_1 ID 4 (Mr. An) maps to 2 (Mr. Vuong) and 1 (Mr. An) while they co-occur at frame 78
switched from 1 (Mr. An) (at frame 58) to 2 (Mr. Vuong) (at frame 81)
⇒ Incorrect detection. Due to GMM couldn’t eliminate FP 4 - 2

Progress report

# Spatio-Temporal Association

5th stage