Progress report

Spatio-Temporal Association

4nd stage

Content

1. Proposal Plan
2. Data Preparation
3. Method: Homography (w/o calibration) with 2D camera
4. Experimental result
5. Next plan

1. Proposed Plan

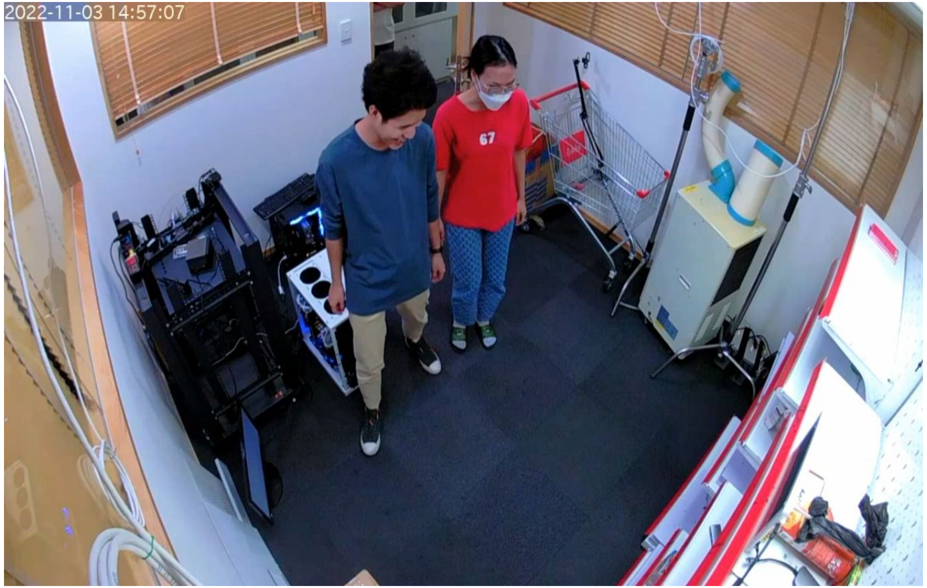
Task	Week 1 (24/10 - 30/10)	Week 2 (31/10 - 06/11)	Week 3 (07/11 - 13/11)	Week 4 (14/11 - 20/11)	Week 5 (21/11 - 27/11)	Week 6 (28/11 - 04/12)	Week 7 (05/12 - 11/12)	Week 8 (12/12 - 18/12)	Week 9 (19/12 - 25/12)	Week 10 (26/12 - 30/12)
Prerequisites for Spatio-Temporal Association										
Setup camera, collect video										
Run SCT, save database										
Make evaluation metrics and ground-truth for MCT										
Do experiments with Spatio-Temporal Association										
Homography w/o calib, cam 2D							Report	Report		
Homography w/o calib, cam 360										
Epipolar line constraint, cam 2D										
Pixel to physical coord, cam 2D										
Report										

1. Data Preparation

1.1. Camera setup & video collection

Experiment 1:

- Pair of sync 2D-cameras
- Number of video pairs: 16, including:
  - Number of people: 2- 3
  - Moving direction: same, different
  - Entry/Exit: same, different



1.1. Camera setup & video collection

Experiment 2:

- Pair of sync 2D-cameras
- Number of video pairs: 6, including:
  - Number of people: 3-4
  - Moving direction: same, different
  - Entry/Exit: same, different
- Smaller overlapping area than Exp.1

1.1. Camera setup & video collection

Experiment 3:

- Pair of sync 360-cameras
- Number of video pairs: 4, including:
    - Number of people: 3
    - Moving direction: same, different
    - Entry/Exit: same, different

1.2. Ground-truth

- Pre-trained StrongSORT + Hand labeling
- Each track includes:
    - `trackid`
    - `camid`
    - `videoid`
    - `detections: List[{timestamp, frameid, box, score}]`

1.3. Evaluation metric

Employing from [1]:

Ground-truth pairs	Predicted pairs	
21,19,2,27,19,2		False Negative
21,19,3,27,19,3		
21,19,4,27,19,4		
21,20,2,27,20,2	21,20,2,27,20,2	
21,20,3,27,20,3	21,20,3,27,20,3	
21,20,4,27,20,4	21,20,4,27,20,4	
21,21,1,27,21,2	21,21,1,27,21,2	True Positive
21,21,2,27,21,3	21,21,2,27,21,3	
21,21,3,27,21,4	21,21,3,27,21,4	
21,22,1,27,22,2	21,22,1,27,22,2	
	21,22,3,27,22,1	
False Positive	21,22,2,27,22,2	

tion and fusion results using *Recall* (R) and *Precision* (P). R is the fraction of accurate associations to the true number of associations. P is the fraction of accurate associations to the total number of achieved associations. Let ξ_Ω be the ground truth for pairs of trajectories on the overlapping region Ω and let E_Ω be the estimated results. Then R and P are calculated as:

$$R = \frac{|\xi_\Omega \cap E_\Omega|}{|\xi_\Omega|}, \quad (13)$$

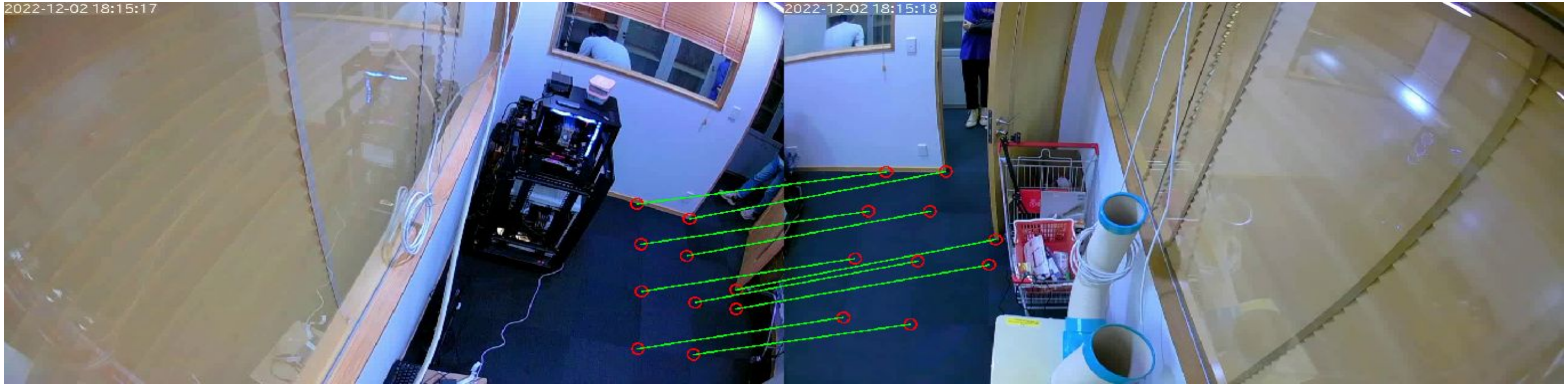
$$P = \frac{|\xi_\Omega \cap E_\Omega|}{|E_\Omega|}, \quad (14)$$

where $|.|$ is the cardinality of a set.

3. Method: Homography (w/o calibration) with 2D camera

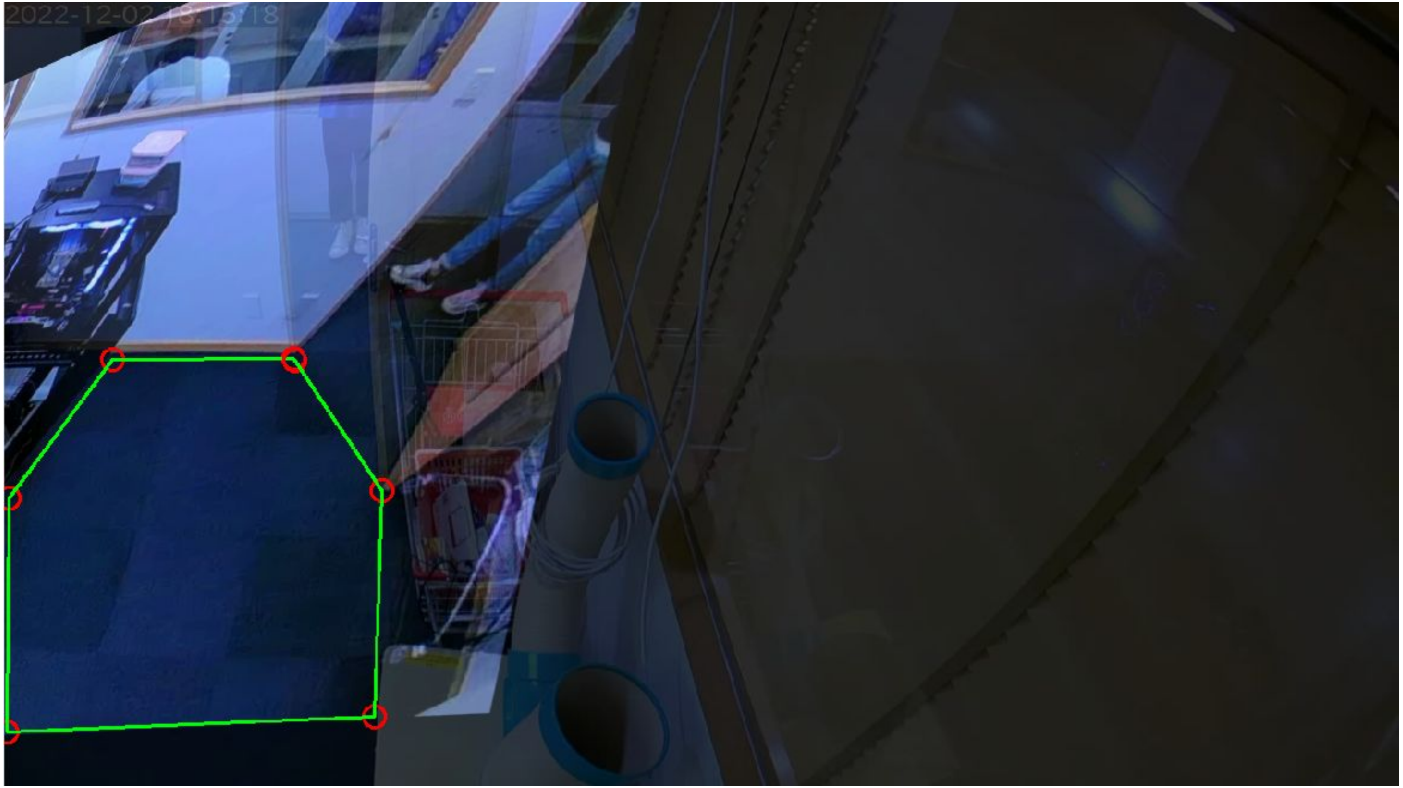
3.1. Homography & overlapping region

Select matches on the floor



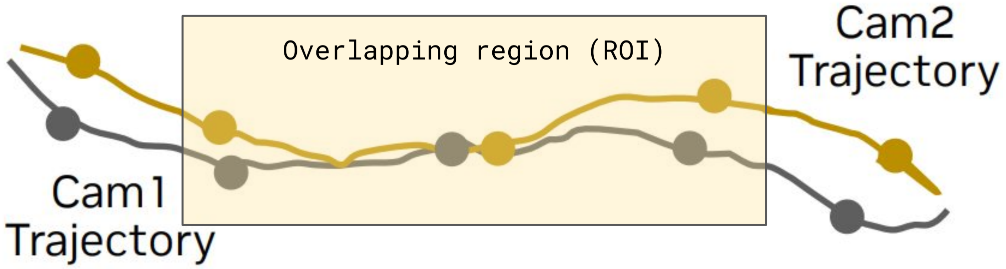
3.1. Homography & overlapping region

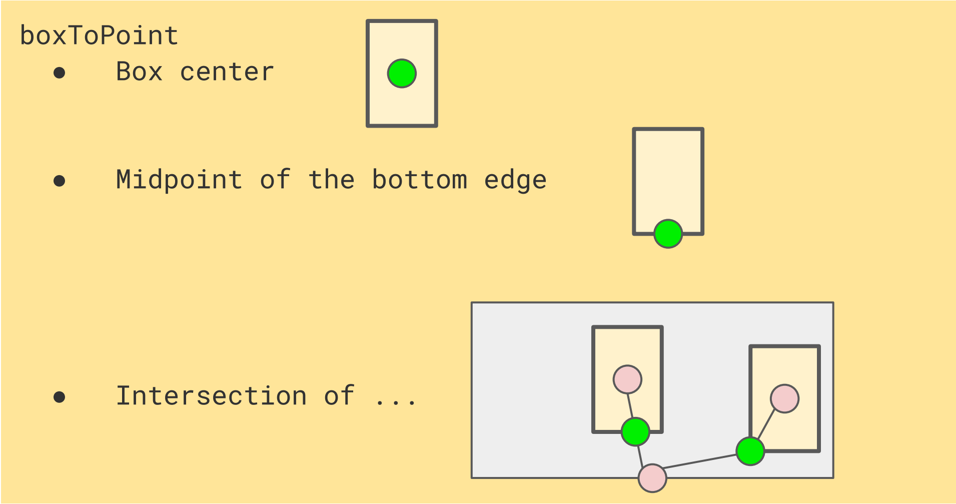
Select overlapping area on the floor



3.2. Matching algorithm

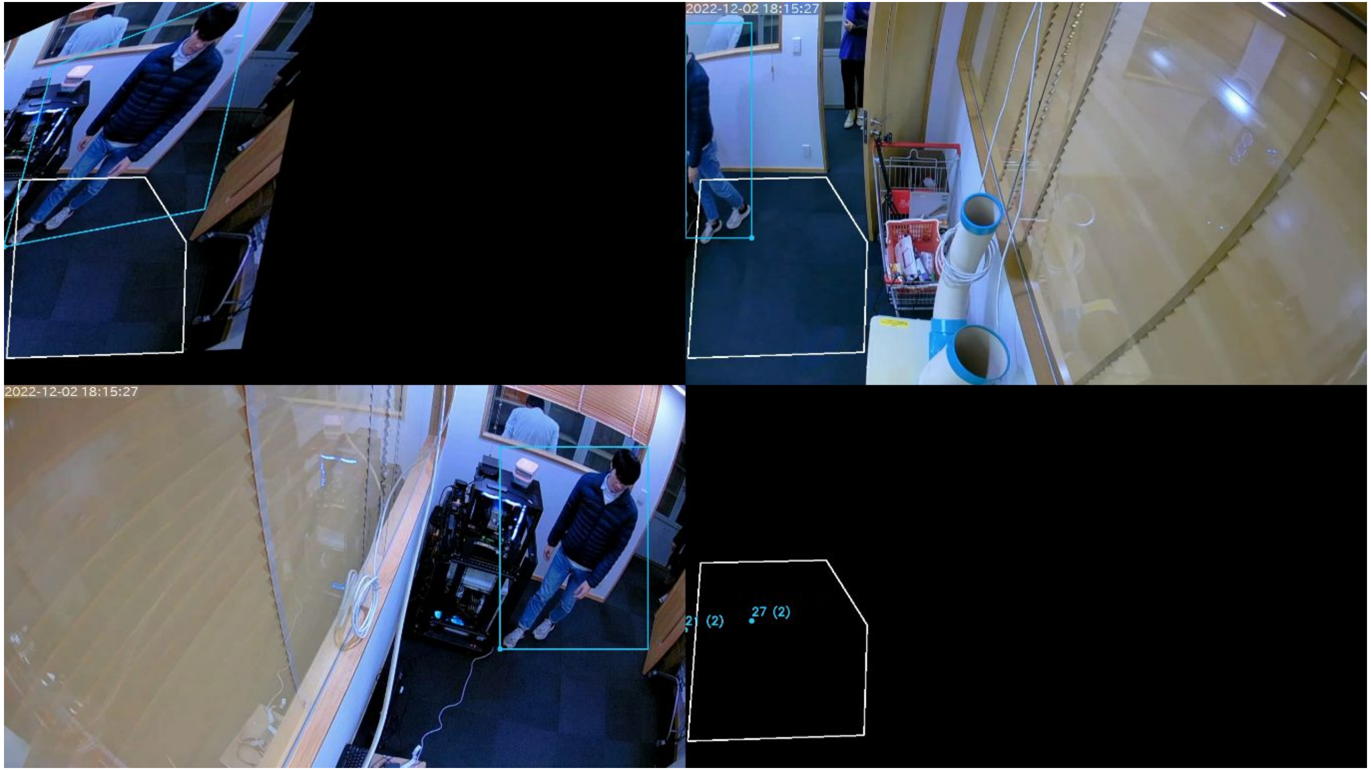
```
Function mapTrack(cam1, cam2):

  cost_matrix = Array[n1, n2]

  For each track1 in cam1:
    For each track2 in cam2:
      track1 <- perspectiveTransform(track1)

      points1 <- boxToPoint(track1)
      points2 <- boxToPoint(track2)

      points1 <- sampleROI(points1)
      points2 <- sampleROI(points2)

      correspondences = sampleTime(points1, points2)

      distance = computeDistance(correspondences)
      cost_matrix[track1, track2] <- distance

  matches <- Hungarian(cost_matrix)

  Return matches
```





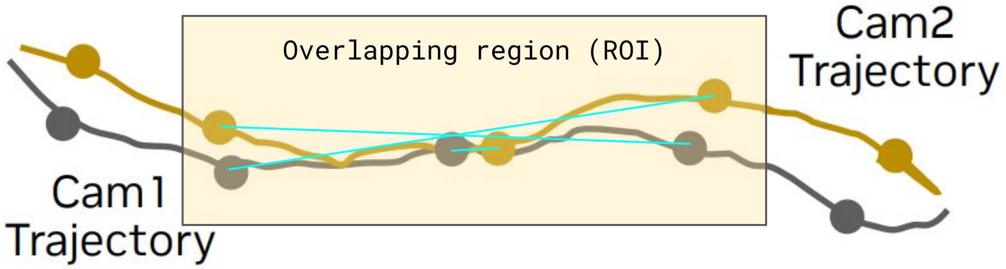
3.2. Matching algorithm

`boxToPoint` an example of the 3rd option



3.2. Matching algorithm

```
Function mapTrack(cam1, cam2):

  cost_matrix = Array[n1, n2]

  For each track1 in cam1:
    For each track2 in cam2:
      track1 <- perspectiveTransform(track1)

      points1 <- boxToPoint(track1)
      points2 <- boxToPoint(track2)

      points1 <- sampleROI(points1)
      points2 <- sampleROI(points2)

      correspondences = sampleTime(points1, points2)

      distance = computeDistance(correspondences)
      cost_matrix[track1, track2] <- distance

  matches <- Hungarian(cost_matrix)

  Return matches
```



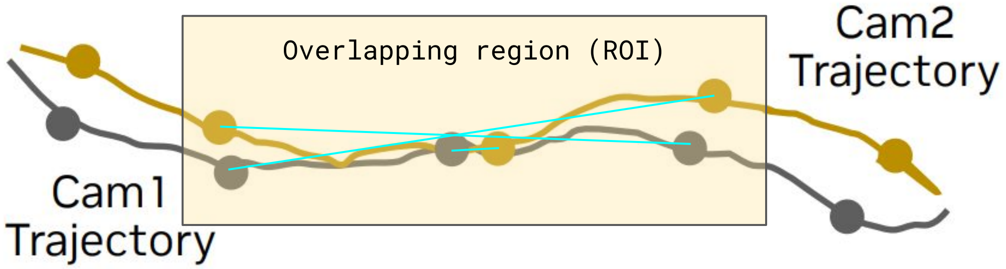
`sampleTime:`

`track1 = List[(point1, timestamp1)]`
`track2 = List[(point2, timestamp2)]`

Match `point1` with `point2` by using Hungarian bipartite matching with `cost = |timestamp1 - timestamp2|`

3.2. Matching algorithm

```
Function mapTrack(cam1, cam2):

  cost_matrix = Array[n1, n2]

  For each track1 in cam1:
    For each track2 in cam2:
      track1 <- perspectiveTransform(track1)

      points1 <- boxToPoint(track1)
      points2 <- boxToPoint(track2)

      points1 <- sampleROI(points1)
      points2 <- sampleROI(points2)

      correspondences = sampleTime(points1, points2)

      distance = computeDistance(correspondences)
      cost_matrix[track1, track2] <- distance

  matches <- Hungarian(cost_matrix)

  Return matches
```



Average Euclid distance between each corresponding points

4. Experimental results

4.1. Experiment 1

N.o	SCT properties	Matching criterias	TP	FP	FN	Precision	Recall
1	Large overlapping area + non-person not cleaned	W/o overlapping area				0.19	0.37
2	Large overlapping area + non-person not cleaned	W/o overlapping area + time-IoU weights	41	37	3	0.47	0.92
3	Large overlapping area + non-person cleaned	W/ overlapping area + time-IoU weights	44	0	0	1.0	1.0
4	Large overlapping area + non-person cleaned	W/ overlapping area	44	0	0	1.0	1.0

4.1. Experiment 1

An example of false matching: Large overlapping area + non-person not cleaned

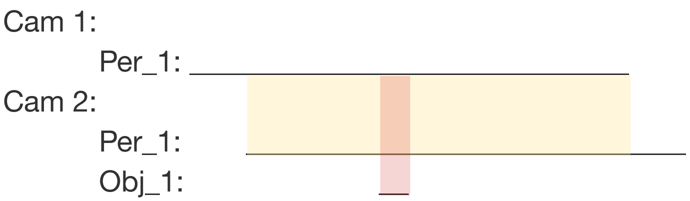
4.1. Experiment 1

Observations

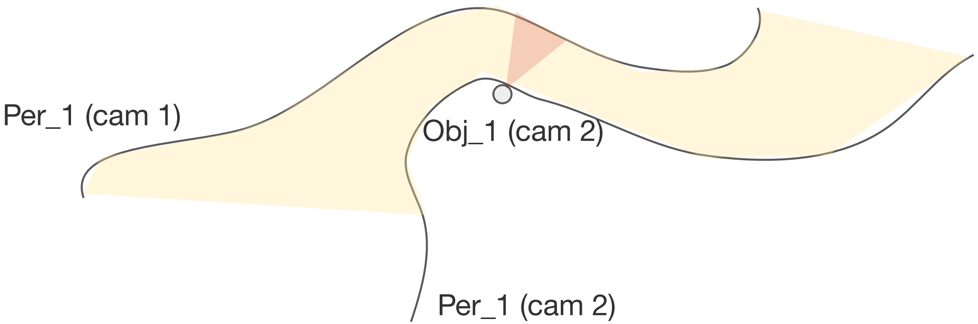
(1) poor result:

- Precision is low because the person detector produces many non-person objects.
- Recall is low because, at the same time:
    - non-person objects only appears in a few frames.
    - during its appearance, it stays close to a person.

Temporal



Spatial (view on cam 2):



(2) uses IoU of time as a weight for the distance:

- Precision is better, but still low because non-person objects are mapped to each others.
- Recall is high, (but) because the overlapping area is large so one person would quite frequently appear on both cameras at the same time.

4.1. Experiment 1

Observations

(3) and (4) give perfect result:

- All non-person objects are removed (hand-removal).
- Because the overlapping area is large, so the IoU makes no difference.

Comments: The video setting is easy

- The overlapping region is large in both camera views.
- The overlapping region is near the camera lens.
- The period that people appears in the overlapping region is long.
- The derived foot is not always precise.

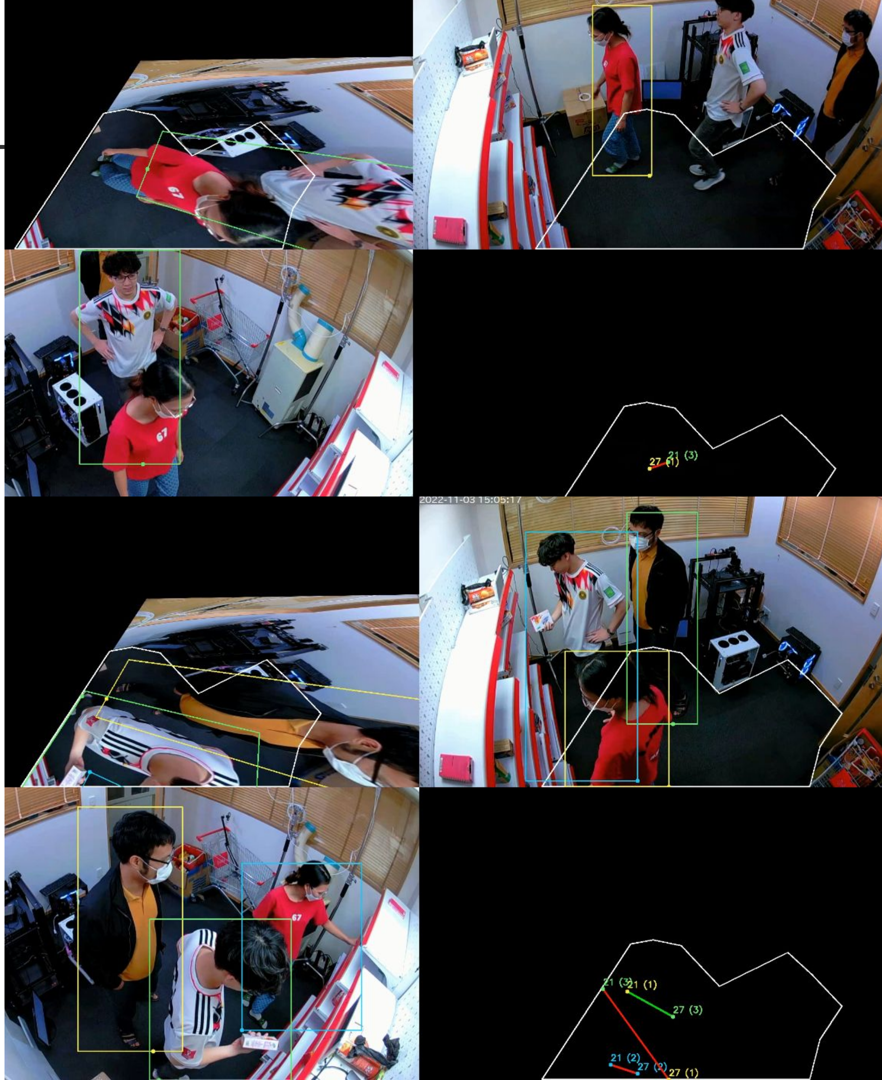
4.1. Experiment 1

Frame-level evaluation:

0 -> False: 28/808 = 0.034653465346534656
1 -> False: 59/576 = 0.10243055555555555
2 -> False: 137/680 = 0.20147058823529412
3 -> False: 270/964 = 0.2800829875518672
4 -> False: 7/555 = 0.012612612612612612
6 -> False: 27/914 = 0.029540481400437644
7 -> False: 37/1063 = 0.034807149576669805
9 -> False: 32/564 = 0.05673758865248227
8 -> False: 76/646 = 0.11764705882352941
10 -> False: 80/1112 = 0.07194244604316546
11 -> False: 34/959 = 0.035453597497393116
12 -> False: 8/772 = 0.010362694300518135
13 -> False: 1/897 = 0.0011148272017837235
14 -> False: 76/943 = 0.08059384941675504
15 -> False: 53/1067 = 0.04967197750702906

Most of the wrong frame-level matching is due to bad detection results:

- Box is missing.
- Box does not fit the object well.



4.2. Experiment 2

N.o	Matching criterias	TP	FP	FN	Precision	Recall
1	W/o overlapping area	18	2	0	0.9	1
2	W/o overlapping area + time-IoU weight	15	5	3	0.75	0.83
3	W/ overlapping area	18	0	0	1.0	1.0
4	W/ overlapping area + time-IoU weight	14	4	4	0.78	0.78

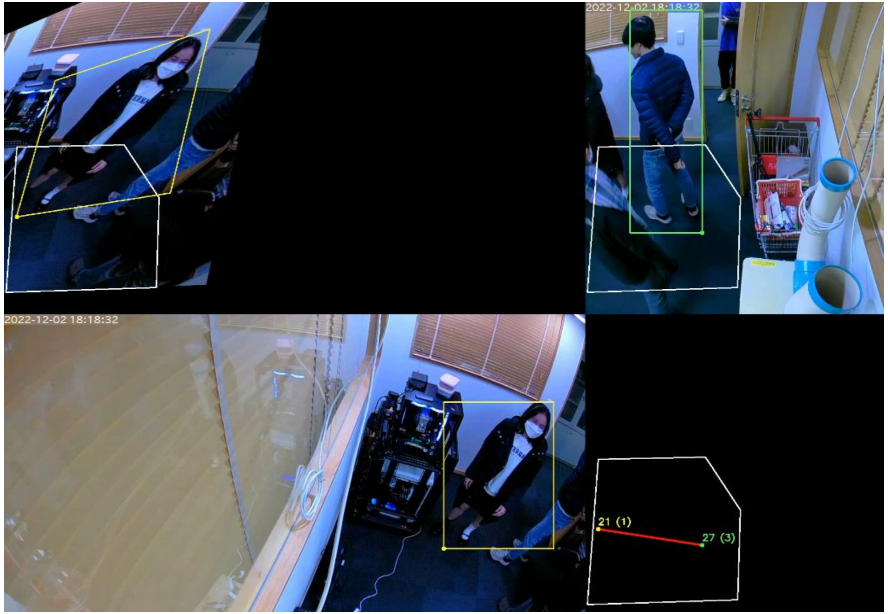
** no non-person object, because the SCT is fully hand-labeled*

4.2. Experiment 2

(1) and (3), which uses time-IoU weights, is worse than (2) and (4):

- The overlapping area is smaller, so 2 distinct people might have higher IoU than a same person.

```
Distance = [[        nan         nan         nan         nan]
            [        nan 120.05335  210.20857  162.9048  ]
            [        nan 155.42838  146.91687  198.11356]
            [        nan 209.24437  146.26242  124.35544]]

IoU = [[0.09328358 0.20746888 0.36764705 0.18315019]
       [0.21828358 0.48547718 0.8602941  0.42857143]
       [0.6660448  0.67507005 0.3809524  0.7647059 ]
       [0.26865673 0.5975104  0.9444444  0.52747256]]

Cost = [[        nan         nan         nan         nan]
        [        nan 247.28938  244.34502  380.1112  ]
        [        nan 230.24037  385.65677  259.07156]
        [        nan 350.1937   154.8661   235.75717]]
```



(3) is better than (1), because tracks that are outside the overlapping area are not matched to any other tracks.

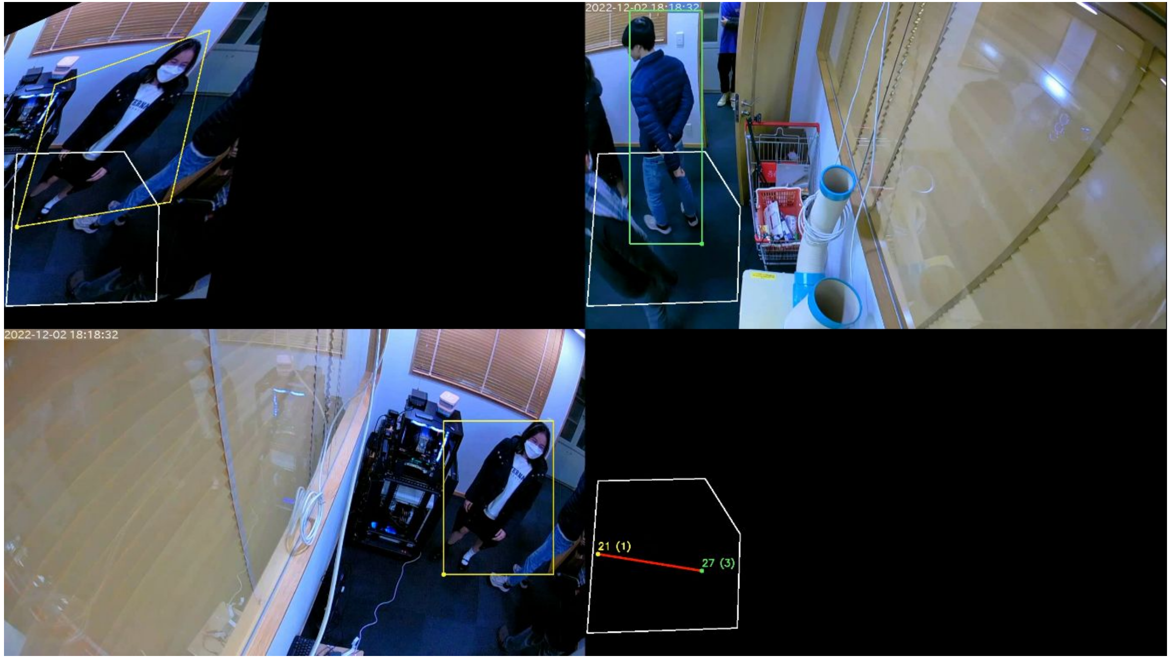
4.2. Experiment 2

Frame-level evaluation:

19 -> False: 88/332 = 0.26506024096385544
20 -> False: 76/320 = 0.2375
21 -> False: 194/447 = 0.43400447427293065
22 -> False: 53/219 = 0.2420091324200913
23 -> False: 24/291 = 0.08247422680412371
24 -> False: 149/470 = 0.3170212765957447

In comparison to Experimental 1, the frame-level error rate is higher, which indicates that the smaller overlapping area is more challenging:

- Box is missing.
- Homography is less precise.



4.2. Experiment 2

Observation:

- The video setting is still:
  - The overlapping region is near the camera lens.
  - The period the people appears in the overlapping region is quite long.
  - The derived foot is not always precise.

References

[1] Trajectory association and fusion across partially overlapping cameras, 2009

Progress report

Spatio-Temporal Association

4th stage