

Description

You are given **(1)** tracks on the ground plane based on position information *only* and **(2)** identity detections (team and jersey number). The goal is to generate identified tracks for each player.

A main issue with the input tracks is that, since they are based on position information only, they will contain ID switches when players are nearby. The idea is to leverage the identity detections to correct this type of mistakes. Although less important, the input tracks may also contain other type of errors such as missing players and false positives.

Further reading: "[Multi-commodity network flow for tracking multiple people](#)", HB Shitrit, J Berclaz, F Fleuret, et. al.

The task is open ended on purpose and we do not expect that you solve it completely considering all possible aspects. Rather, we would like you to come up with an approach to provide the tracks with player ID (or even to come up with new tracks with ID). Then, we would like to discuss your approach both from a conceptual point of view as well as going through your implementation.

Resources

- Sample video files from each view.
- Soft-id or position-only tracks: *mtp-out-vBGtestv1-50traj.trj*
- Upper body detections with team and number classifications: *detections.dat*
- Configuration relating the occupancy map and the camera views: *room.pom*

File formats

1. Position-only tracks.

The first line contains a positive integer denoting the number of tracks `num_tracks`. Then, `num_tracks` lines follow.

Each of the `num_tracks` lines has the following format:

```
track_soft_id start_frame num_frames score cell_0 cell_1 ... cell_{num_frames-1}
```

Where:

- `track_soft_id` is a soft qualifier (meaning without real identify information) for the track;
- `start_frame` is the 0-based index denoting the start of the track;
- `num_frames` is the length of the track;
- `score` is a positive real number denoting the goodness of the track based on position information only; and

- and cell_i indicates what cell the track occupies at time step start_frame+i.

Due to batch operation of the tracker, start_frame can have an offset with respect to the video. For the sample file provided, the offset is 29 (that is, if a track has start_frame 0 in the track files, that corresponds to the track starting at the 30th frame of the video).

2. Upper body detections with team and number classifications.

CSV file with the following line format:

```
camera_view,frame,bounding_box_x,bounding_box_y,width,height,ref,ref_score,team-a,team-a_score,team-b-team-b_score,keeper-a,keeper-a_score,keeper-b,keeper-b_score,jersey#1,jersey#1_score,jersey#2,jersey#2_score,jersey#3,jersey#3_score,jersey#4,jersey#4_score,jersey#5,jersey#5_score
```

For example:

```
10,99,2270,204,64,53,ref,0.0000,team-a,1.0000,team-b,0.0000,keeper-a,0.0000,keeper-b,0.0000,fv,0.9999,07,0.0000,16,0.0000,18,0.0000,05,0.0000
```

- camera_view *1-based index* to the camera with the detection (this corresponds to the camXX in the video file names). Current BallJames systems are composed of 14 cameras.
- bounding_box_x and bounding_box_y denote the pixel coordinates of the top left corner of the bounding box of the upper body detection; width and height denote the dimensions of the bounding box in pixels.
- Team class:
ref,ref_score,team-a,team-a_score,team-b-team-b_score,keeper-a,keeper-a_score,keeper-b,keeper-b_score
In the provided file only team-a and team-b are used.
- Top-5 jersey number classification results:
jersey#1,jersey#1_score,jersey#2,jersey#2_score,jersey#3,jersey#3_score,jersey#4,jersey#4_score,jersey#5,jersey#5_score
The jersey# “ref” and “fv” denote that the jersey number detector did not find a number in the image patch with the upper body.

3. Configuration relating the occupancy map and the camera views.

There is a line

```
ROOM 2328 1752 14 31104
```

denoting that the video frames have 2328x1752 resolution, that there are 14 camera views, and that the field has been discretized into a grid with 31104 cells. Each other line has the following format

```
RECTANGLE camera_view cell_index [rectangle description]
```

For example:

```
RECTANGLE 0 12240 284 76 299 133  
RECTANGLE 13 27935 notvisible
```

The tracks are given in terms of cell indices. The information in this file relates the cells to the camera views (*remark: camera_view is 0-based in this file, contrary to the detections file*). In particular, a cell in the ground plane corresponds to a rectangle in the camera views. A cell that is not visible from a certain camera view is denoted by notvisible. If visible, the rectangle is described in the standard bounding box format: 4 integers denoting the pixel coordinates of the top left corner, the width, and height.