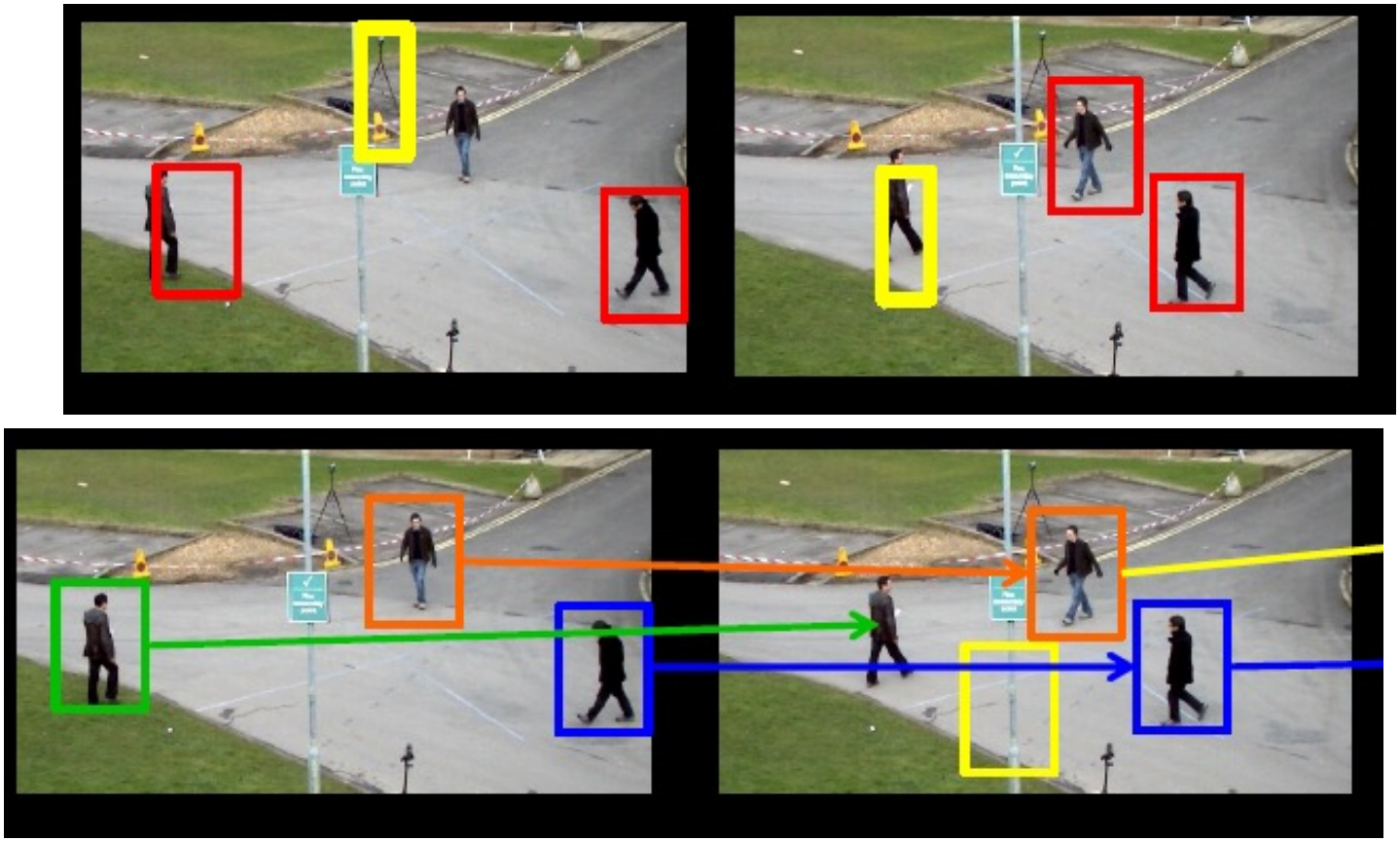
Tracking by Detection

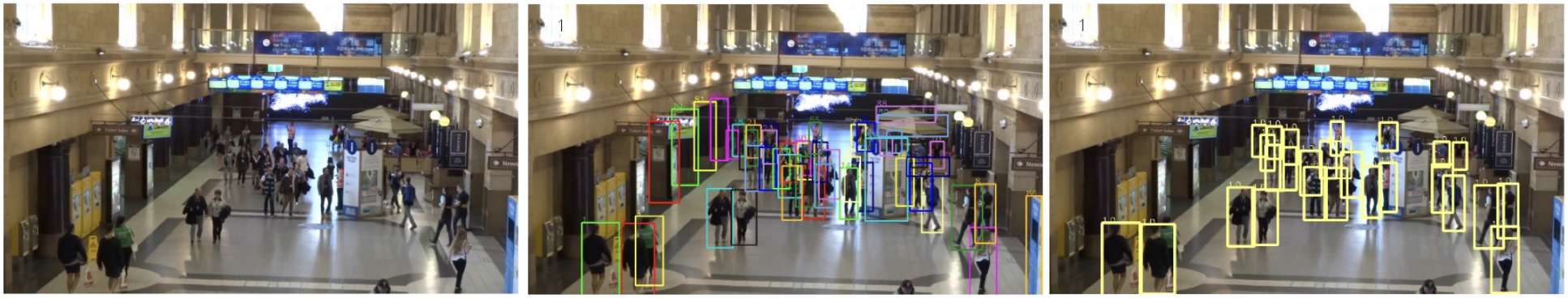
D.A. Forsyth

University of Illinois at Urbana-Champaign

Multi-object tracking



MOT-20 example

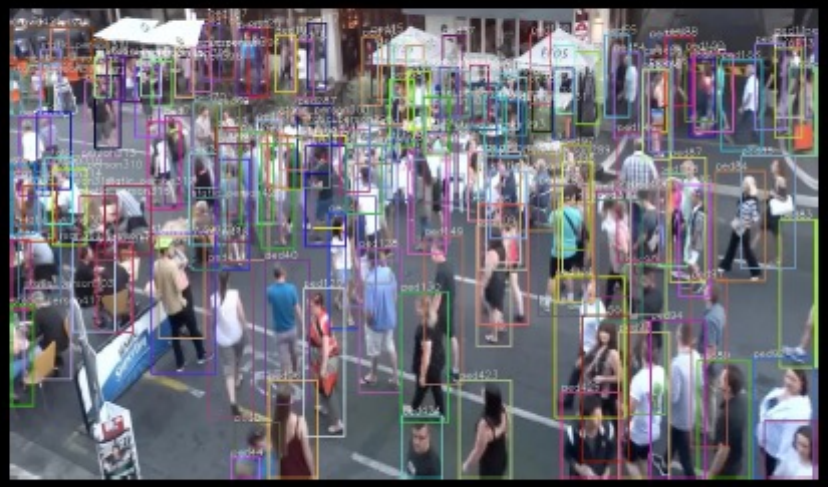


Tracking – Why?

- Motion capture
  - build models of moving people from video
- Recognition from motion
  - eg cyclists move differently than runners
- Surveillance
  - who is doing what?
    - for security (eg keep people out of sensitive areas in airports)
    - for HCI (eg kinect, eyetoy, etc.)

CHALLENGES

- Multiple objects of the same type
- Heavy occlusions
- Appearance is often very similar



Multiple slides lifted from L. Leal-Taixe's slides at ICVSS 2022

Tracking - What

- Establish state of object using time sequence
  - state could be:
    - position; position+velocity; position+velocity+acceleration
    - or more complex, eg all joint angles for a person
  - Biggest problem -- Data Association
    - which image pixels are informative, which are not?
- Key ideas
  - Tracking by detection
    - if we know what an object looks like, that selects the pixels to use

OFFLINE VS. ONLINE

- Online tracking
  - Processes frames as they become available
  - For real-time applications, e.g., autonomous driving, AR/VR
  - Prone to drifting → hard to recover from errors or occlusion

- Offline tracking
  - Processes a batch of frames
  - Good to recover from occlusions (short ones as we will see)
  - Not suitable for real-time applications
  - Suitable for video analysis, automatic labeling, video editing.

Tracking by detection

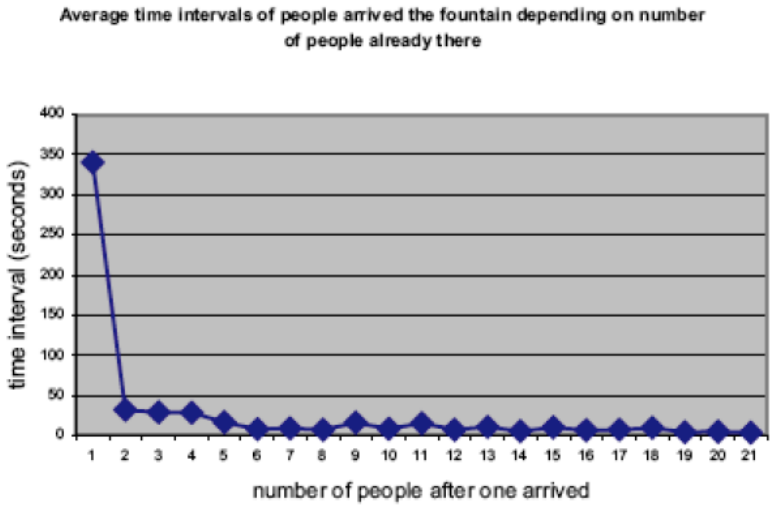
- Assume
  - a reliable detector
  - Link detects across time
    - weighted bipartite matching

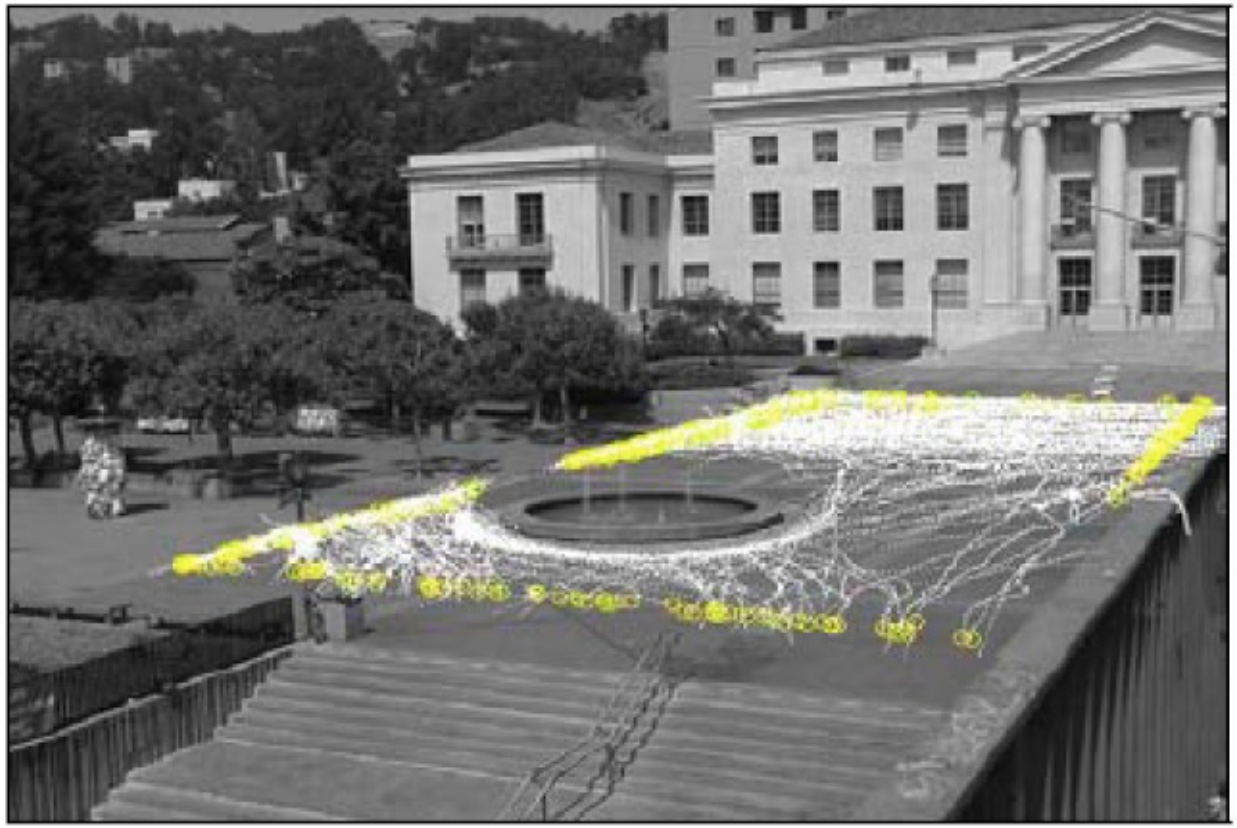
Tracking by detection (STbD) is a simple tracking strategy that assumes you have an object detector (Chapter ??; Section ??). A straightforward procedure is to detect objects in frame 1. Now iterate:

- Detect objects in frame $i + 1$.

- Compute correspondences between the objects in frame i and those in frame $i + 1$. Each object in frame i is linked to at most one object in frame $i + 1$. Each object in frame $i + 1$ is linked to at most object in frame i.

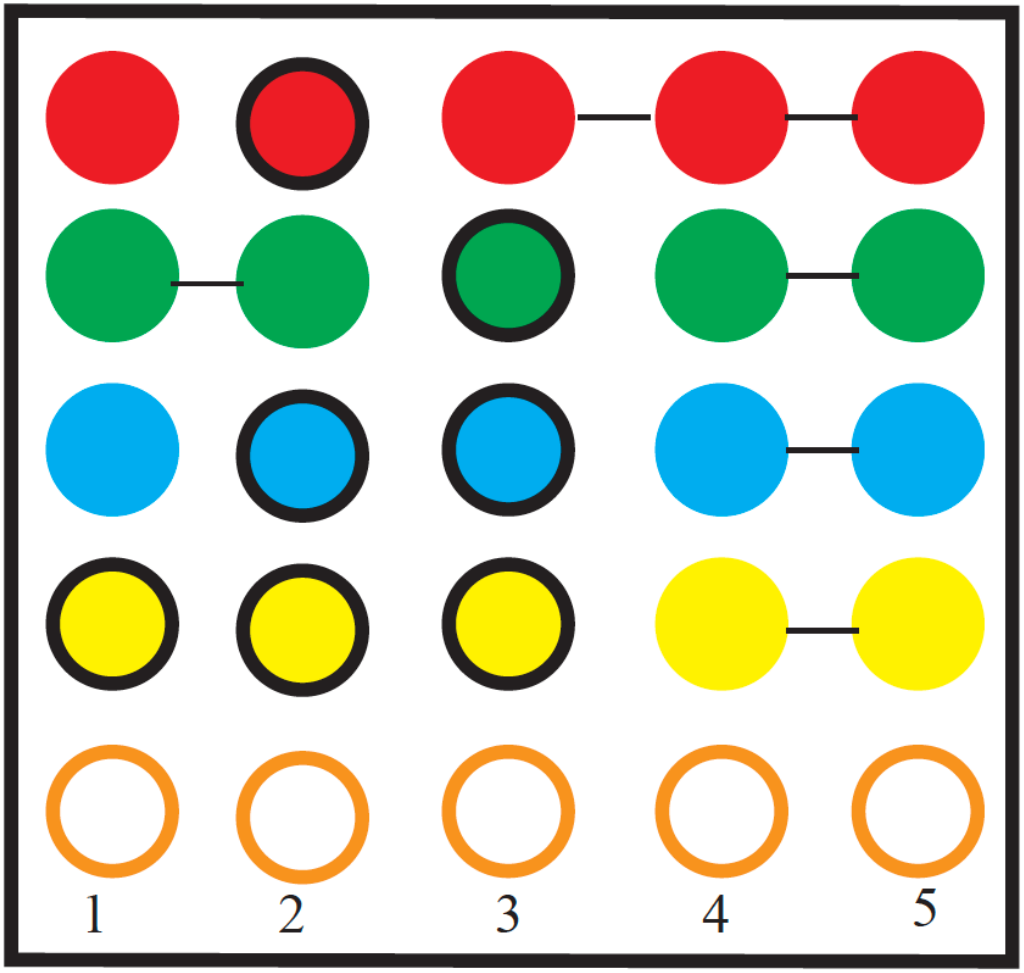
Point tracks reveal public behaviour





Yan+Forsyth, 04

Simplest -> broken tracks

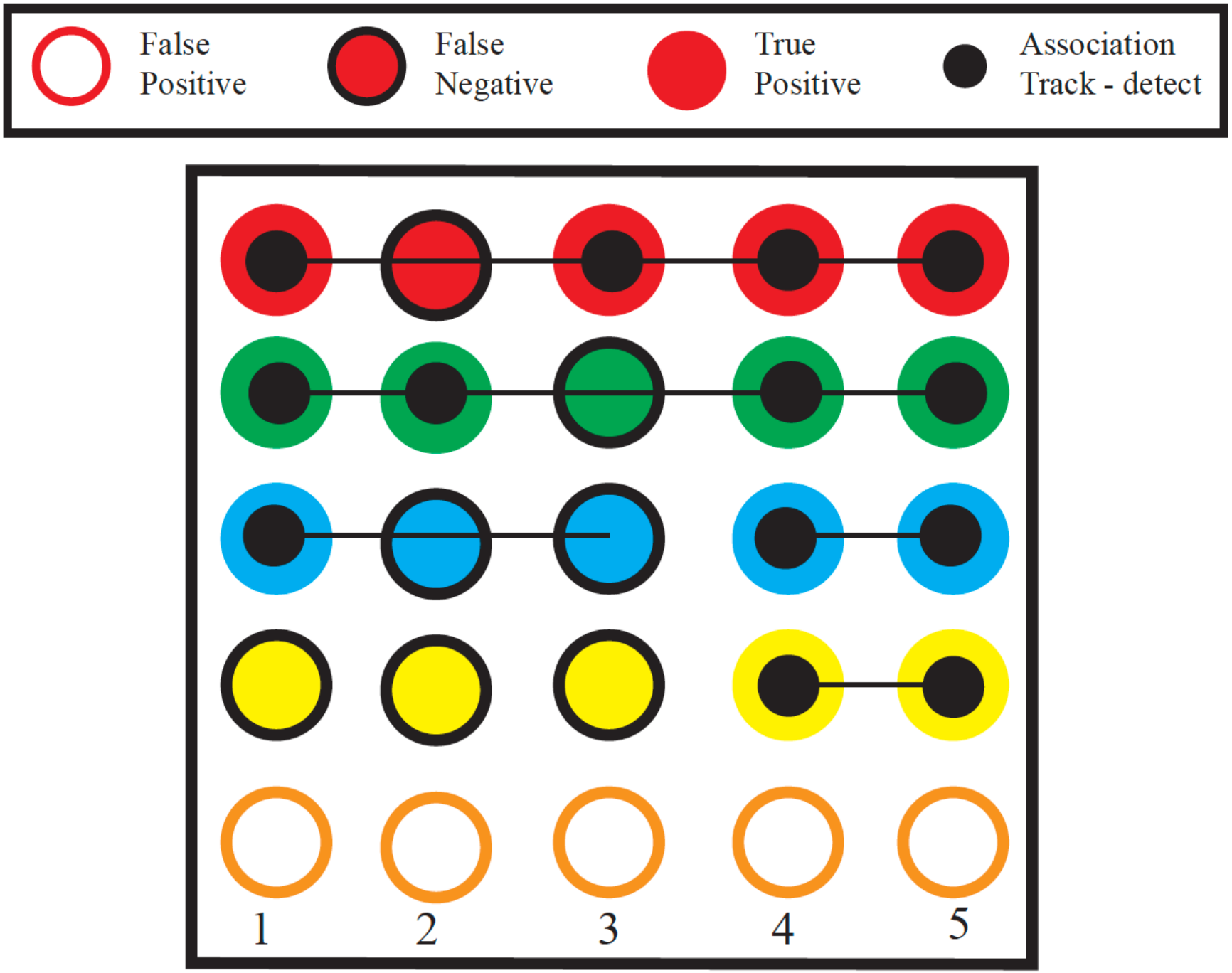


Solution: abstract tracks

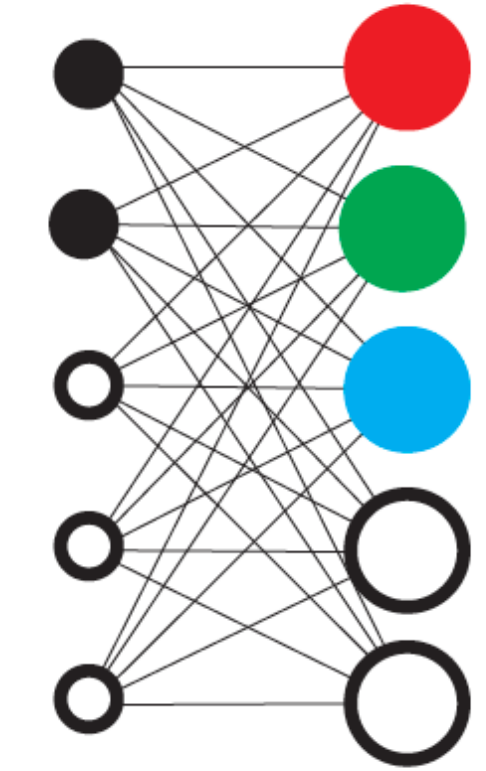
Procedure: 1.1 *Tracking by detection links detector responses to abstract tracks.*

Detect objects in frame 1, and create one track per object. Now maintain tracks by iterating:

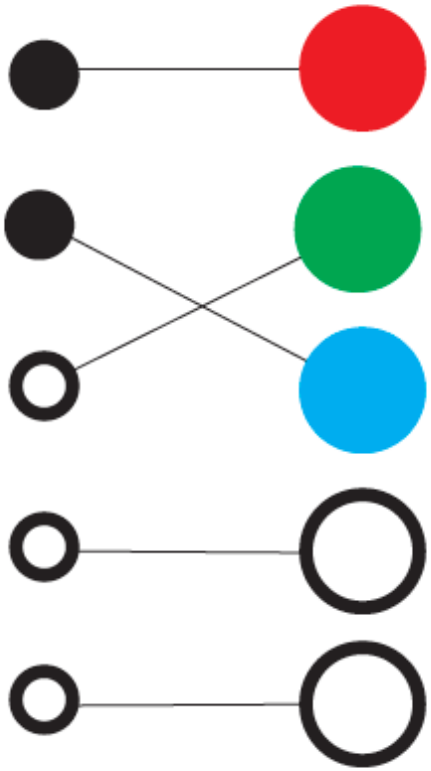
- Detect objects in frame $i + 1$.
- Compute correspondences between the objects in frame $i + 1$ and the tracks. Each track gets at most one object. Each object goes to at most one track.
- Update each track that received an object in frame $i + 1$.
- If a track does not have a corresponding detect, hold onto it for some number of frames. If it has had no associated detect for several frames, reap the track.
- If you have a detect that cannot be associated with any current track, create a new track.


False
Positive
False
Negative
True
Positive
Association
Track - detect
1
2
3
4
5

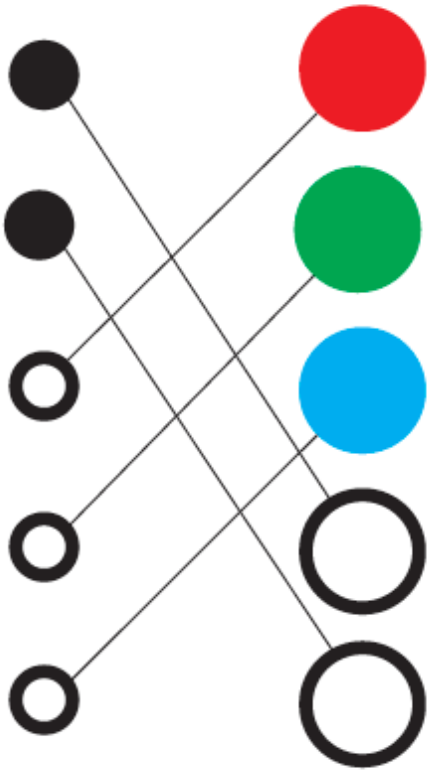
Correspondence



Bipartite graph



A



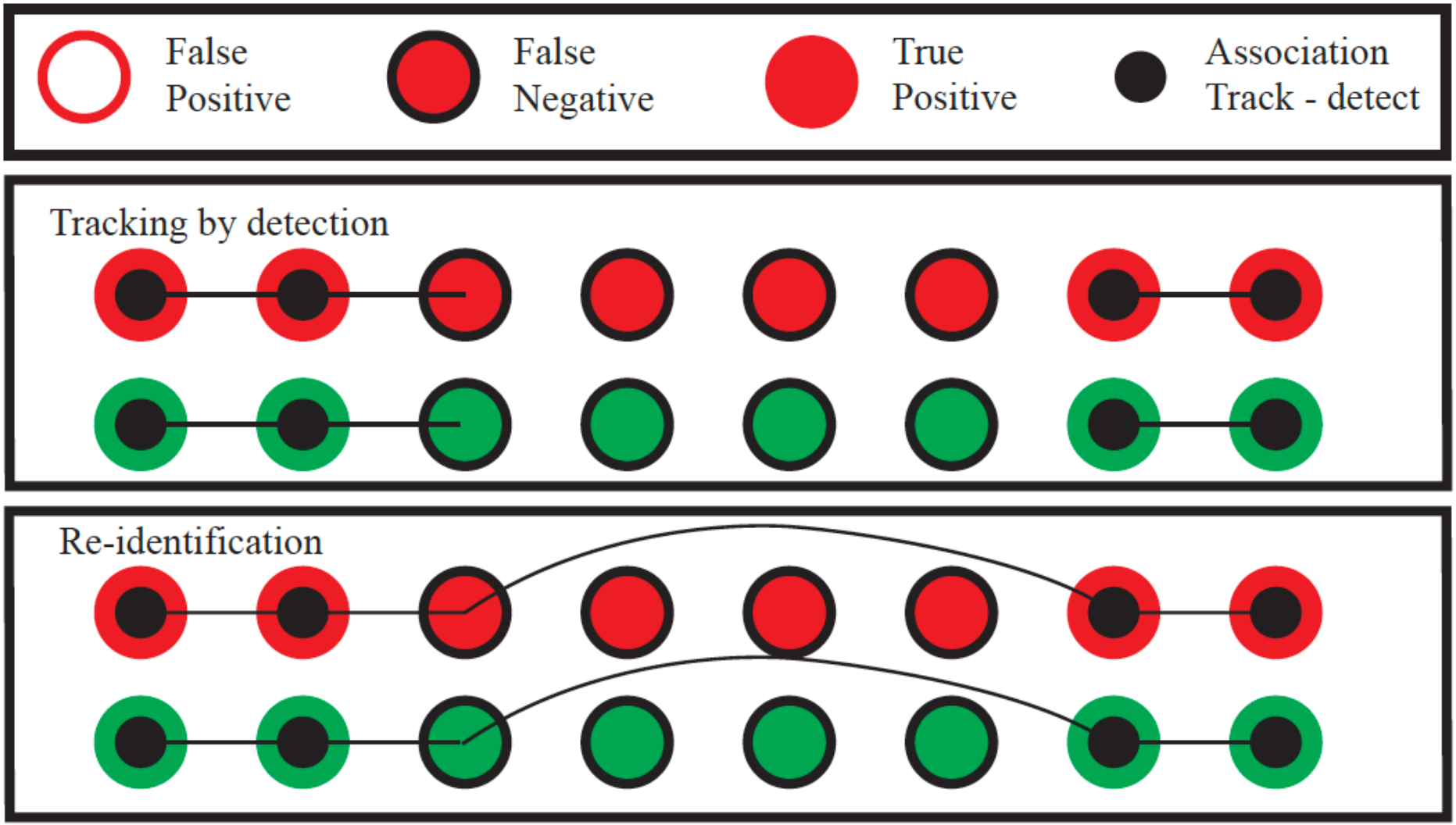
B

Use the Hungarian algorithm
Weights from appearance, dynamics

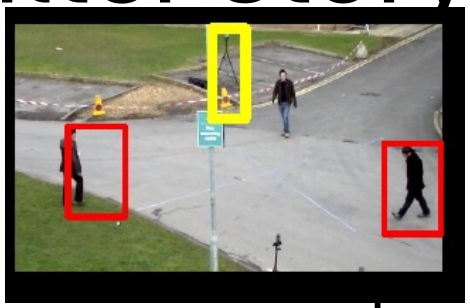
Appearance weights

- Distance between feature vectors
  - tracks could have multiple
    - memory
- Some test of distance
  - between predicted pos’n and true pos’n
    - 1-IOU has a good record

Reidentification



Recall Kalman Filter story



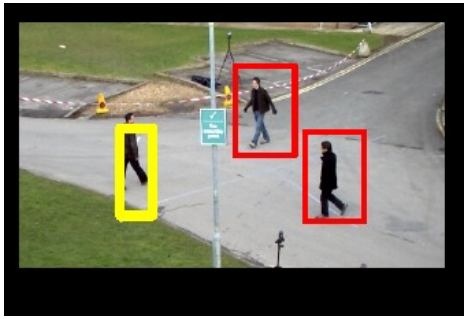
Have:

Mean and covariance of posterior after i-1'th measurement

Construct:

Mean and covariance of predictive distribution just before i'th measurement

This predicts where Boxes might be in Next frame



Measurement arrives:

Now construct:

Mean and covariance of posterior distribution just after i'th measurement

posterior mean is weighted combo of prior mean and measurement

Attaching a Kalman filter

- Use filter to represent, update tracks
  - one filter to manage state for each track
- Associate measurements to tracks with WBM
- Now update filter
  - usually, a constant velocity model is enough
  - sometimes, velocity is zero and noise is large

Evaluation

- Fiddly
  - detection errors
  - correspondence errors
- Different metrics weight these differently
- Details in text

TbD variants- Exploit FasterRCNN

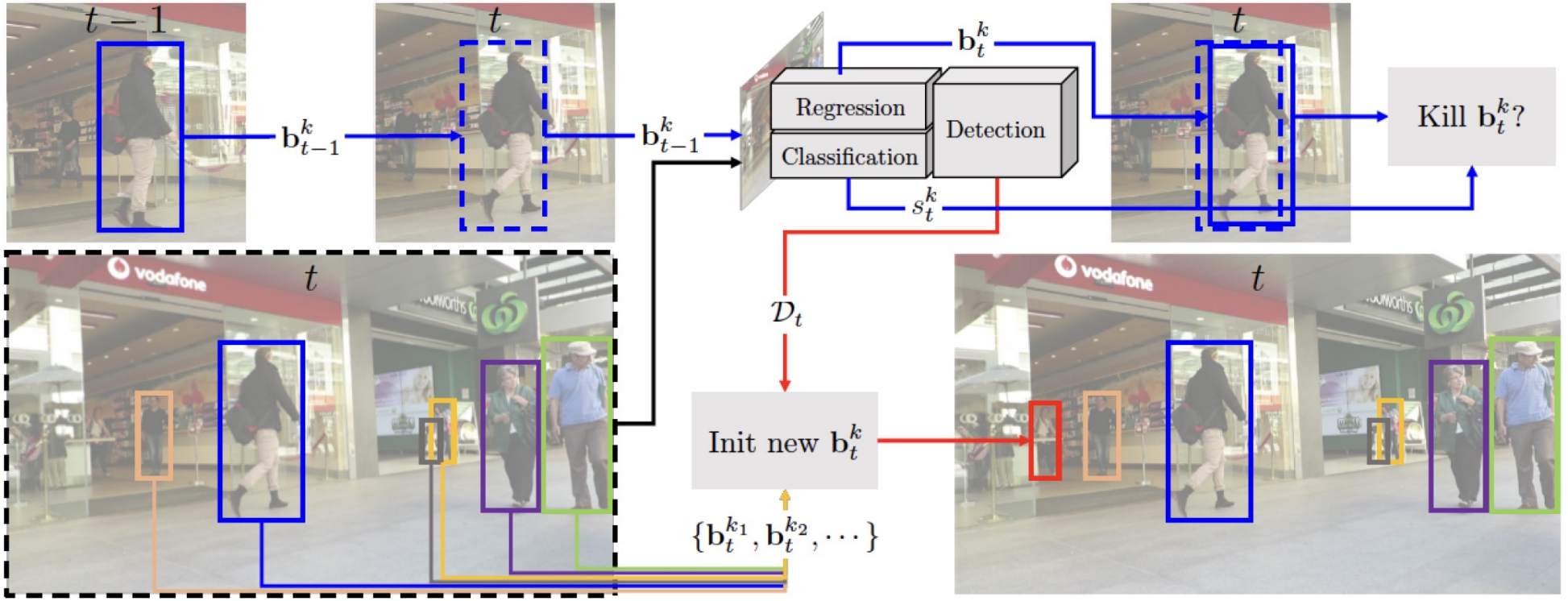
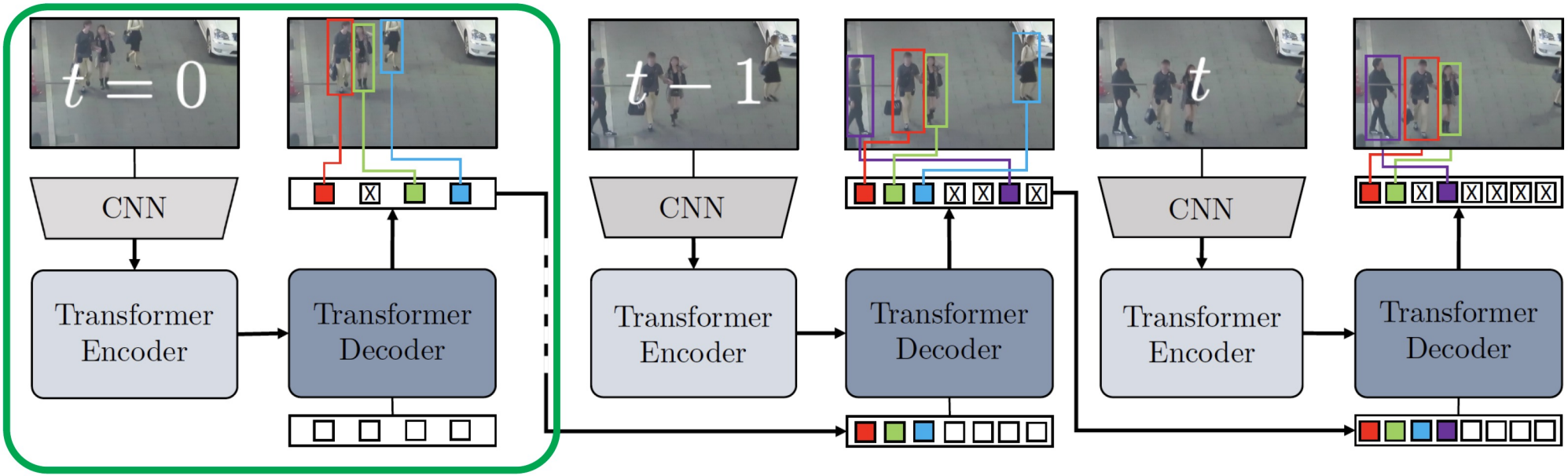


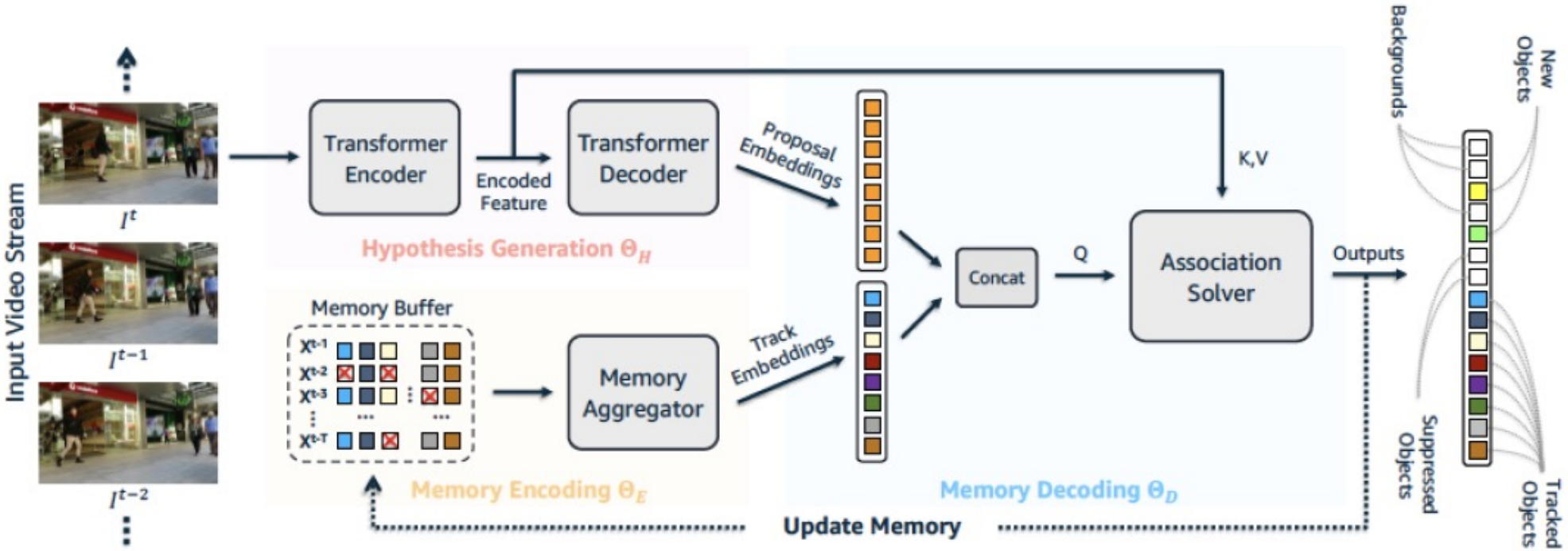
Figure 1 of "Tracking without bells and whistles", Bergmann et al, 2019.

TbD variants- Exploit DetR



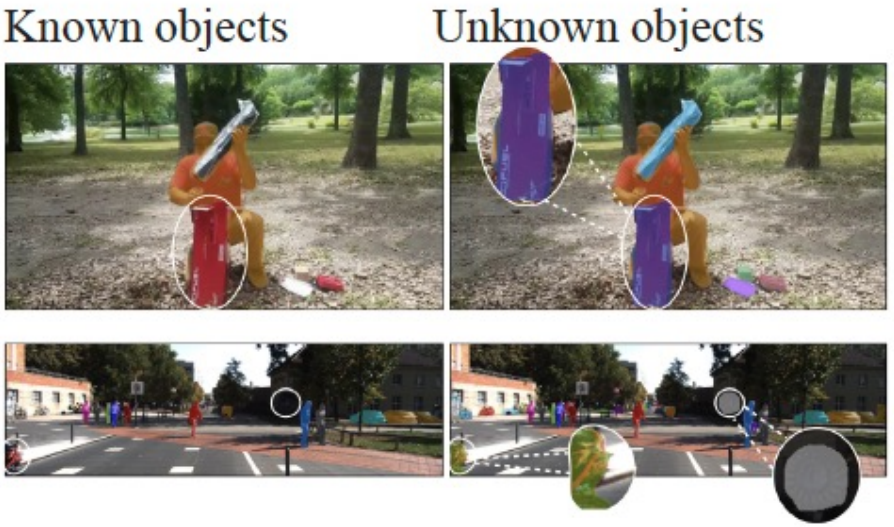
of "Tracking without bells and whistles'TrackFormer: Multi-Object Tracking with Transformers", Meinhardt et al, 2021

TbD variants - Memory



objects are then used to update the memory. Image credit: *Figure 2 of "MeMOT: Multi-Object Tracking with Memory", Cai et al, 2023.*

TbD variants – open world



- Use RPN to detect objects
  - adjust RPN score
- Use appearance match for correspondence

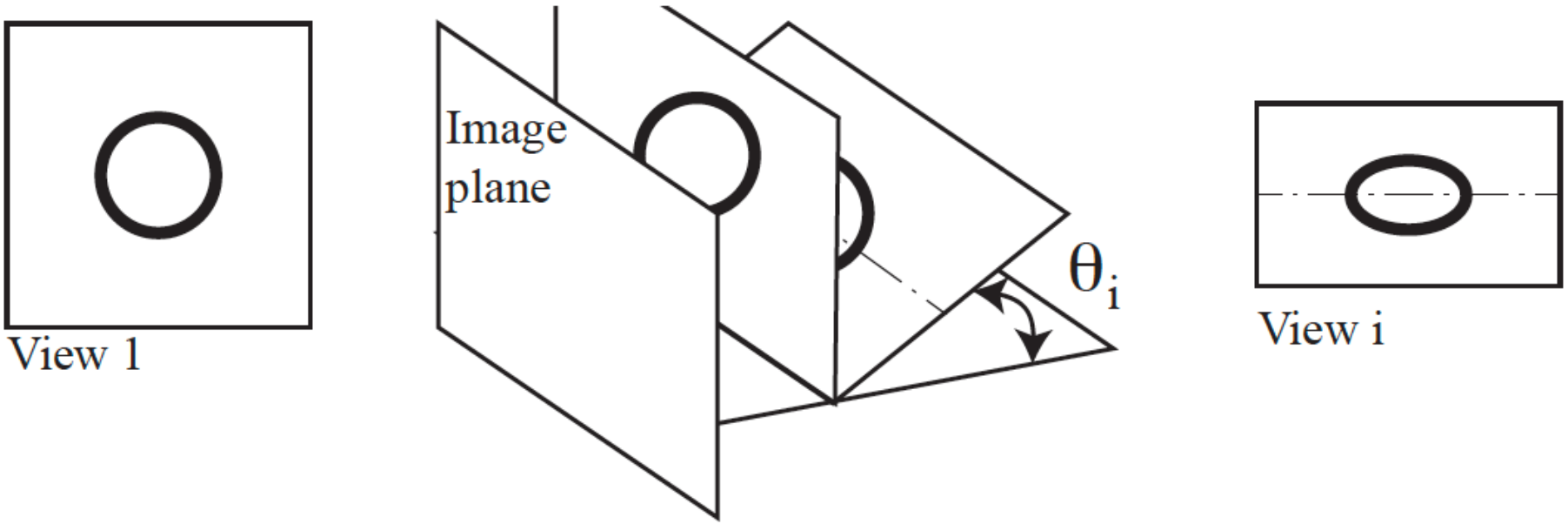
across time is straightforward, details in text. Image credit: *Figure 1 of "Opening up OpenWorld Tracking", Liu et al, 2022.*

TbD variants – tracking segments

shows the correspondence from frame to frame. Image credit: *Figure 1 of "Tracking Anything with Decoupled Video Segmentation", Liu et al, 2022.*

Why not points?

- Idea
  - apply TbD recipe to interest points
- Obstacle
  - appearance match using SIFT is a problem
  - Foreshortening!



Particle video

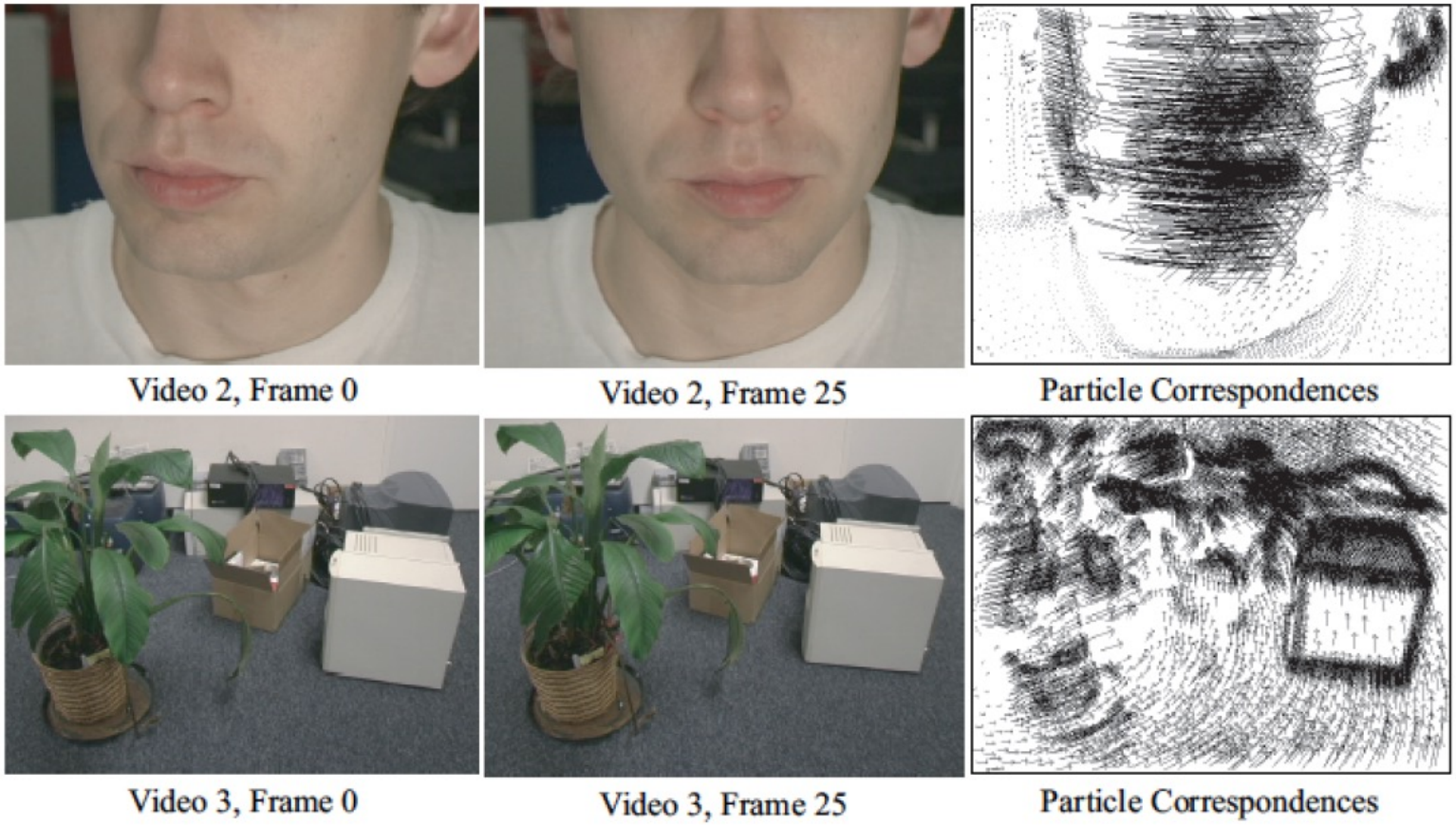


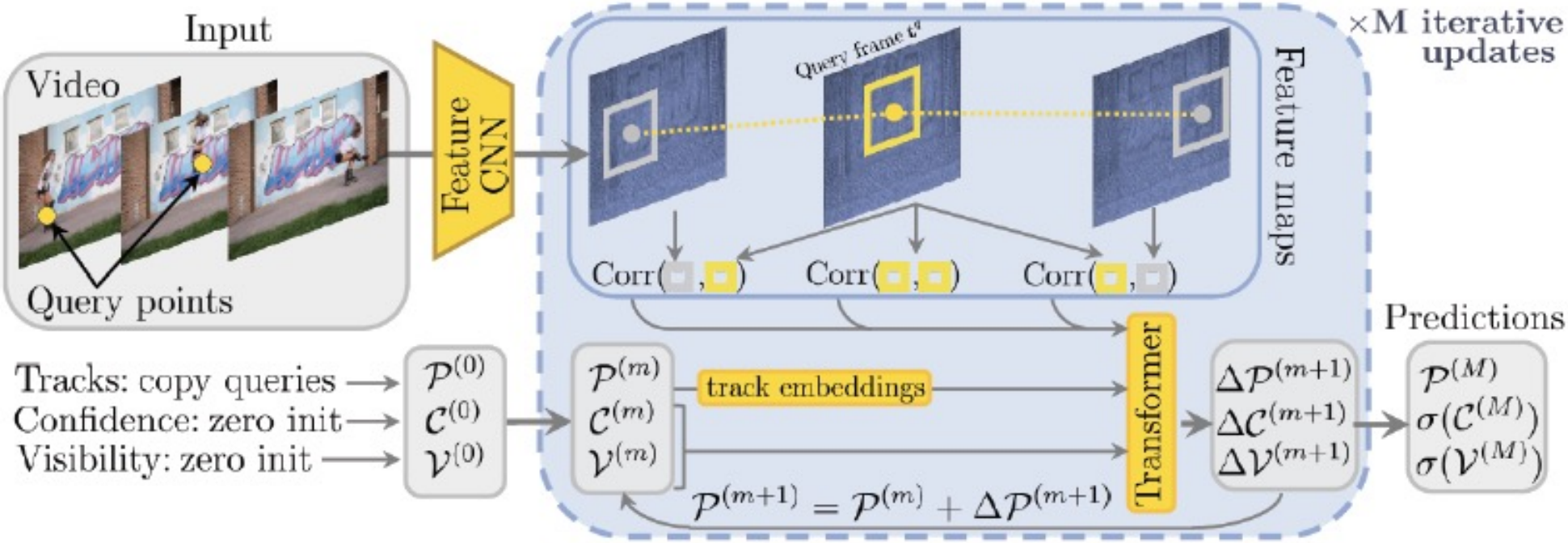
Figure constructed using parts of Figure 6 of "Particle Video: Long-Range Motion Estimation using Point Trajectories", Sand and Teller, 06.

TAPIR

- Make coarse point tracks
  - refine across space/time
- Predict visibility and confidence at each pt

frame. Figure constructed using parts of Figure 4 of "TAPIR: Tracking Any Point with per-frame Initialization and temporal Refinement", Doersch et al, 2023

CoTracker3



parts of Figure 2 of "CoTracker3: Simpler and Better Point Tracking by Pseudo-Labelling Real Videos", Karaev et al, 2024

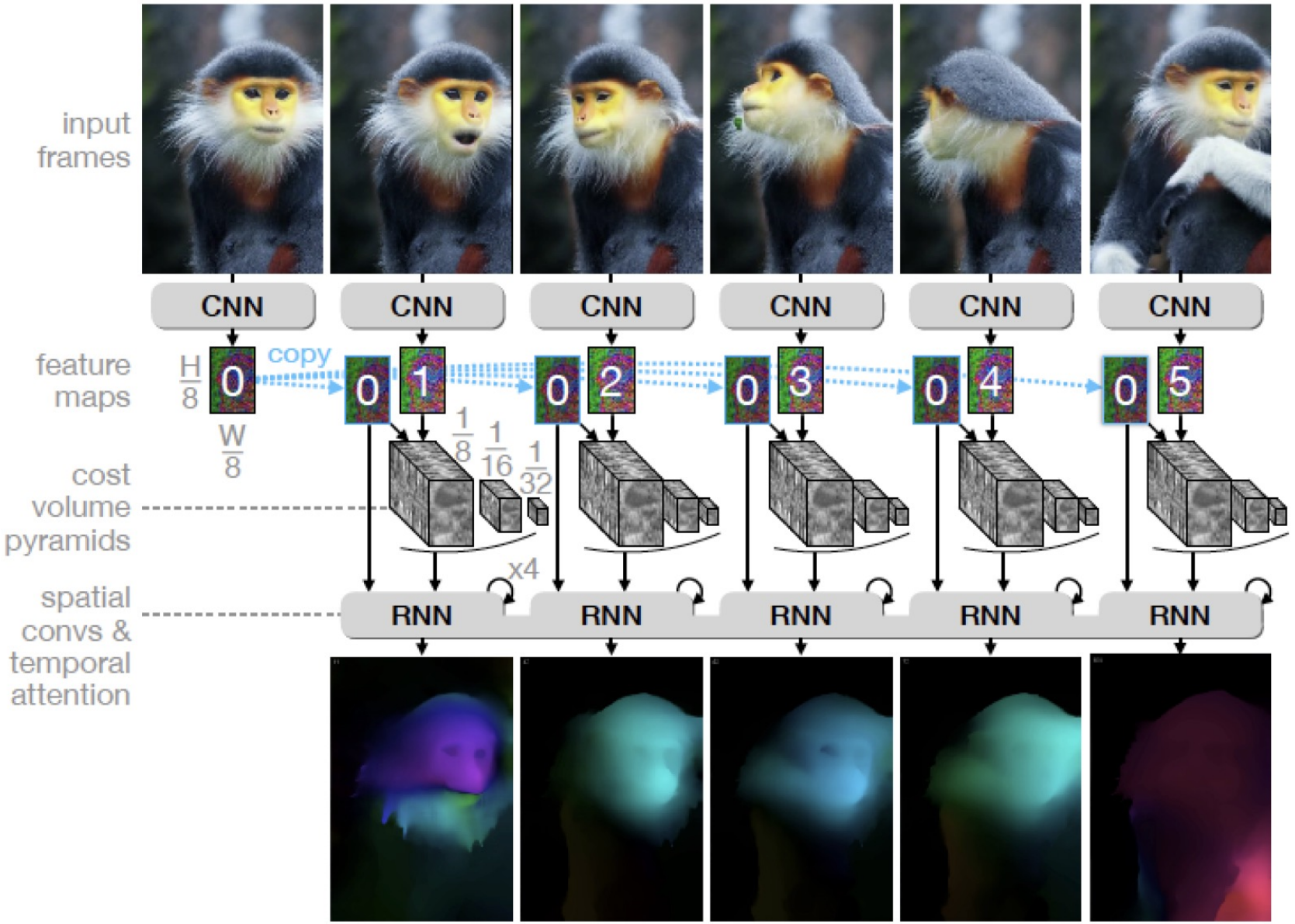


FIGURE 1.16: *Figure constructed using parts of Figure 2 of "AllTracker: Efficient Dense Point Tracking at High Resolution", Harley et al, 202X*