ELEC 2885: Image Processing and Computer Vision

Christophe De Vleeschouwer

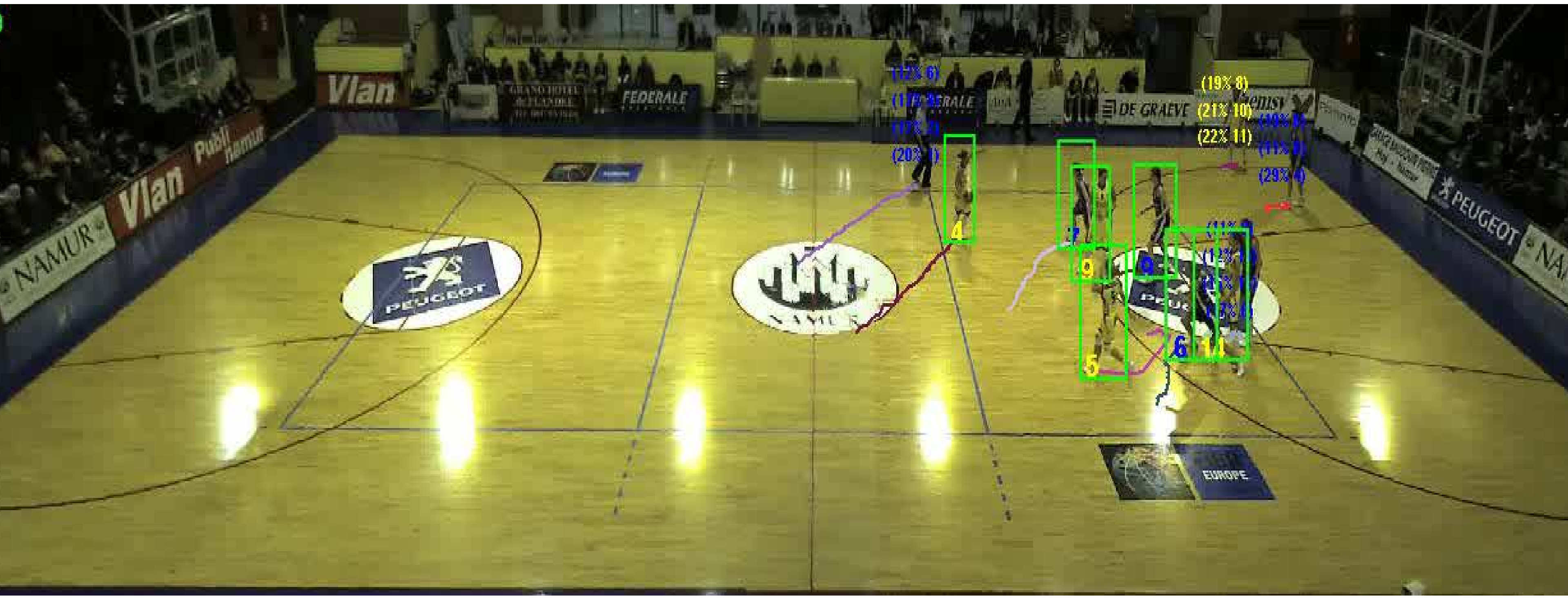
christophe.devleeschouwer@uclouvain.be

Outline

- (Detection-based) tracking motivation

- Graph-based tracking formalisms
  - Shortest path
  - K-shortest path
  - Iterative hypothesis testing
  - Label propagation

What is tracking ?



- Definition:

  Tracking can be defined as the problem of estimating the trajectory of an object in a sequence of images, as the object moves in the scene observed by the camera.

Why is tracking useful ?

- Object/people tracking = Key step towards:
  - Dynamic scene understanding & behavior analysis:
    - Videosurveillance;
    - Traffic monitoring;
    - Sport analytics;
    - Cells migration analysis in biology.
  - Automatic & real-time control of robots:
    - Vehicle navigation e.g. for fully automatic cars or drones;
    - Human/Computer interaction (gesture recognition, eye gaze tracking, etc).
    - Autonomous video content capture and production (In LLN: https://synergysports.com/solutions/live-sport-production/);

Challenges & methods

- Challenges:
  - loss of information due to projection of 3D world on a 2D image;
  - non-rigid or articulated nature of objects;
  - partial and full object occlusions;
  - complex object motion, shapes & textures;
  - scene illumination changes, noisy images;
  - real-time processing requirements;
  - latency constraints.

- Solution: account for application-specific prior knowledge, e.g.:
  - assume that the object motion is smooth with no abrupt changes (e.g. constant velocity or constant acceleration);
  - assume constrained appearance or constant number of objects.

- Two paradigms:
  - Track2Detect;
  - Detect2Track.

Track2Detect

- Motivation:
  - An object does not move/change a lot between consecutive frames.

- Big picture:
  - Recursive procedure:
    - Use detection in previous frame to initialize search in current frame;
    - Key issue: appearance model definition and update.
  - Usage:
    - Suited to low latency scenario;
    - Not suited to handle long term tracking with sporadic (e.g. due to occlusions) appearance cues;
    - Not suited to exploit exclusivity constraints in a multi-object tracking scenario.

Detect2Track

- Motivation:
  - Efficient and effective detectors exist (cfr. face detection, instance segmentation with DL models);
  - Target detection is inherent to tracking initialization or tracking reset.

- Big picture:
  - Association procedure:
    - Object detected based on motion (foreground mask) and/or texture analysis;
    - Given the detections, tracking becomes a temporal association problem, and graph-based formalism to aggregate detections into tracks;

  - Usage:
    - Especially suited to scenarios involving multiple objects;
    - But generally requires a delay, to have access to a temporal window of detections -> better suited to behavior analysis than to real-time low latency control.

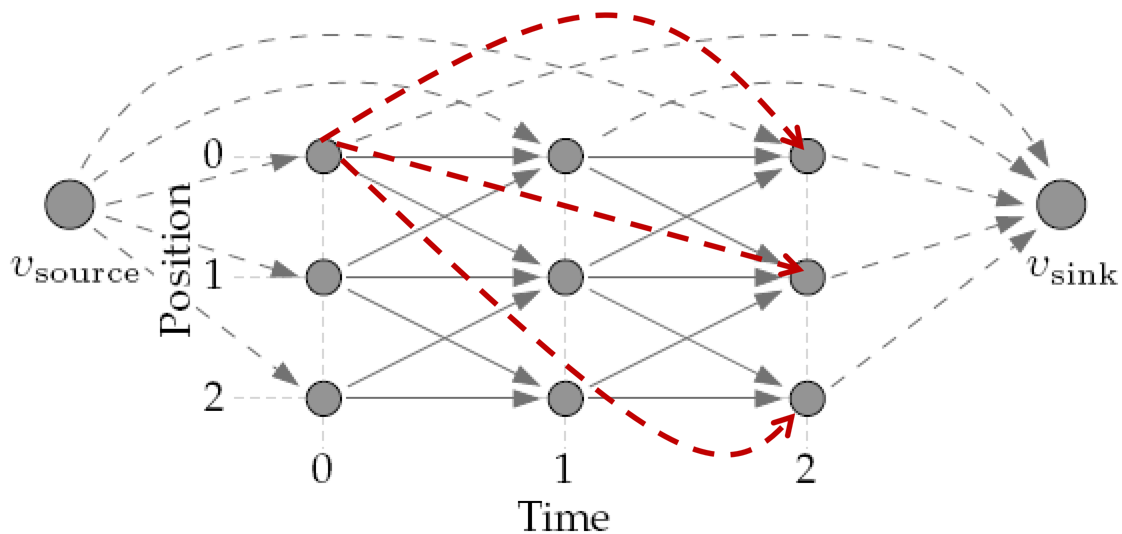
Outline

- (Detection-based) tracking motivation
- Graph-based tracking formalisms
  - Shortest path
  - K-shortest path
  - Iterative hypothesis testing
  - Label propagation

Tracking with shortest paths computation in a graph

Graph construction (Note: the picture only depicts a fraction of the relevant edges):

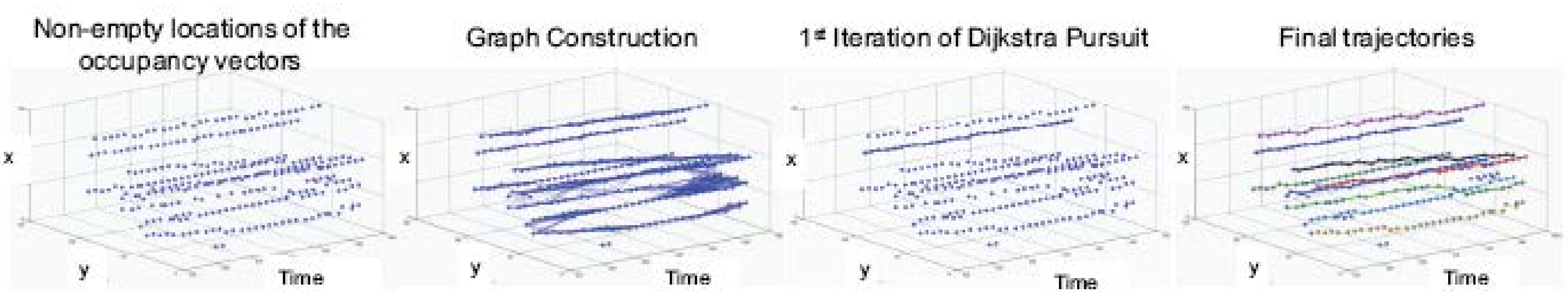
- Vertices = nodes = prior detections; Source and sink nodes are relevant to promote entrance/exit of objects in specific positions (e.g. doors).
- Solid edges are sufficient in case of reliable detection, but red dashed edges have to be introduced in practice, to deal with missed detections;
- The cost of an edge typically increases with
    - The spatial distance;
    - The temporal distance (~number of missed detections);
    - The appearance discrepancies between the detected objects.



Computing the shortest paths

- One single target -> Dijkstra.

- N-tragets:
  - Greedy and iterative selection of shortest-paths using Dijkstra.

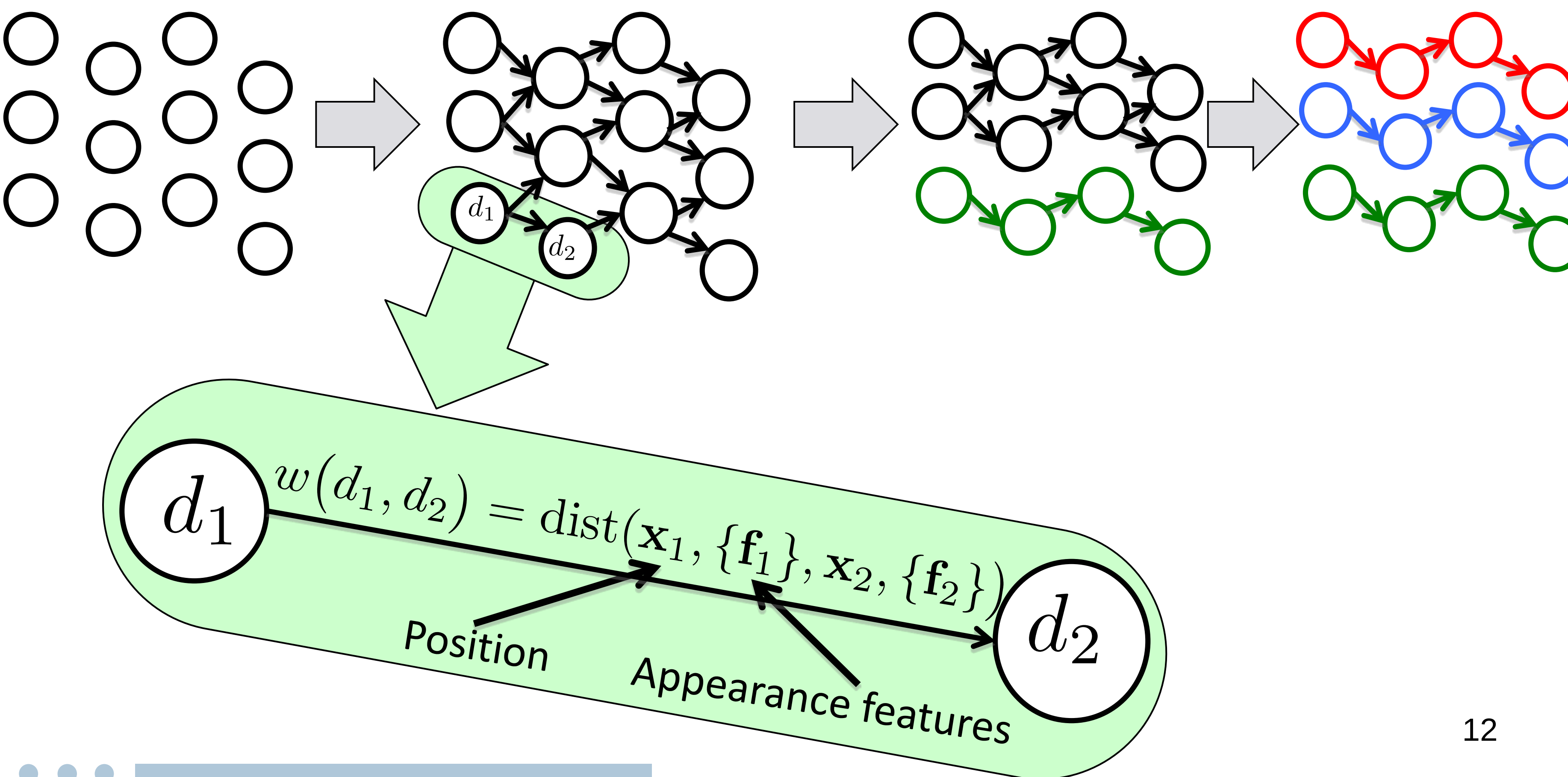


- K-shortest paths, i.e. find the K paths between the source and the sink nodes such that the total cost (= sum of edges costs) of the paths is minimum.

  Note: the problem has been well-studied in network optimization.
  More details about tracking specificities (e.g. only one path per node, directed acyclic graph) in:
  Berclaz and al., 'Multiple Object Tracking Using K-Shortest Paths Optimization', IEEE TPAMI 2011.

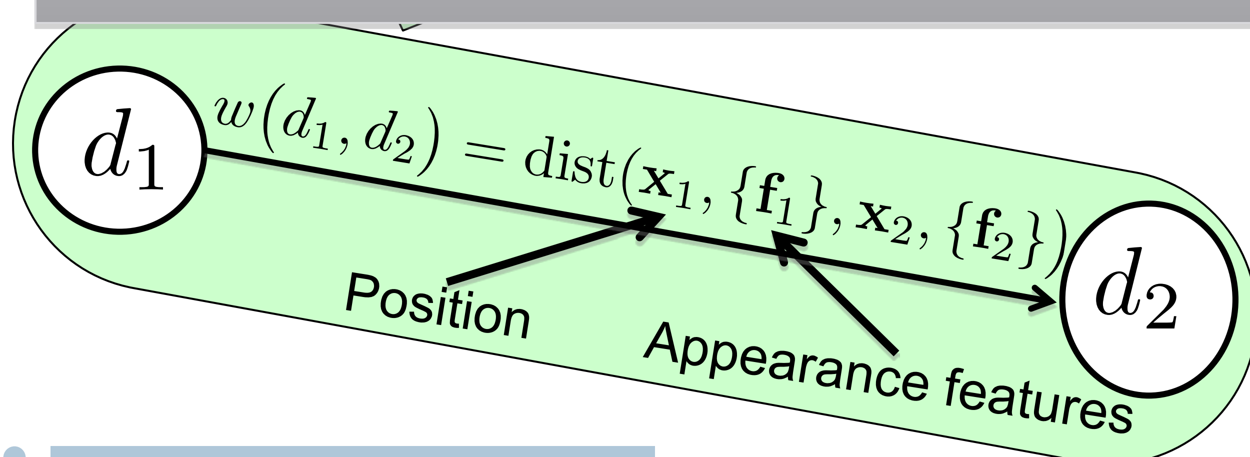
Fundamentally: shortest paths accumulates local distances

Detections | Construct graph | Shortest paths (Iteratively or jointly) | Final trajectories

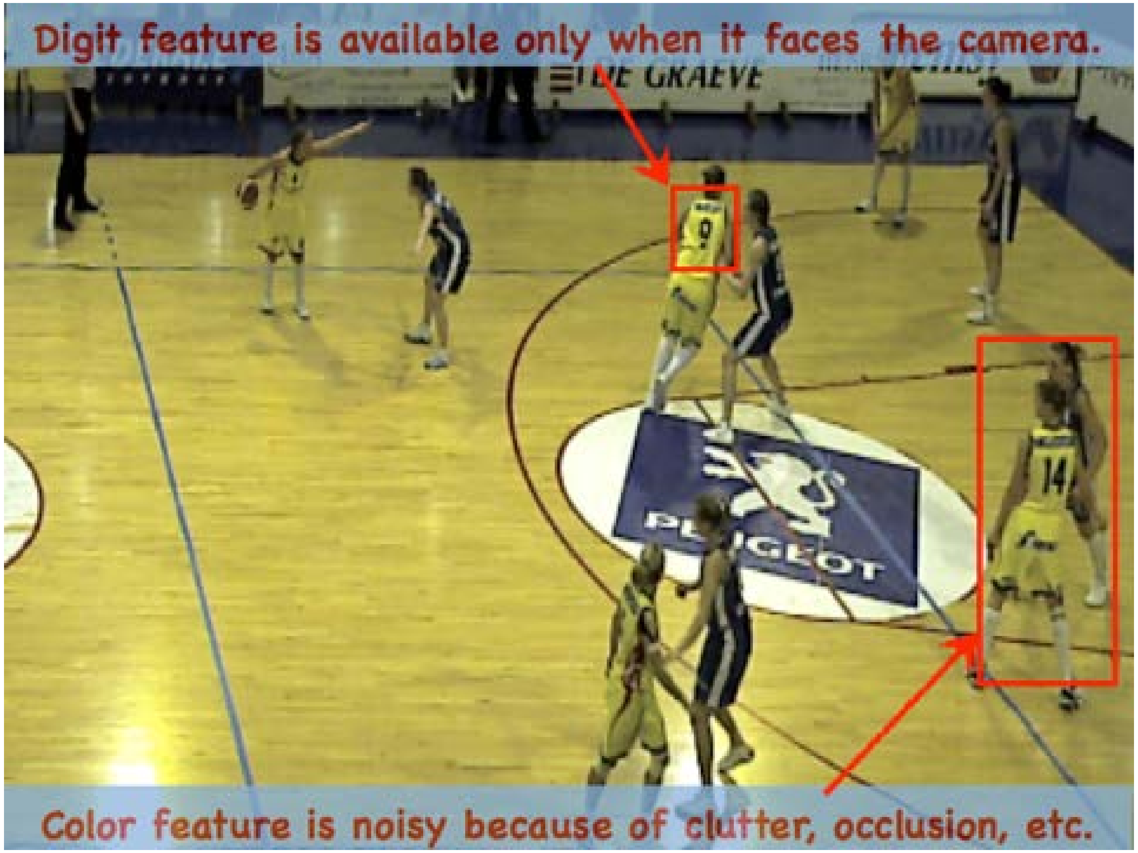


Detections | Construct graph | Shortest paths (Iteratively or jointly) | Final trajectories

Main problem: Accumulation of distance between consecutive nodes works ONLY if the features are available in every node with similar reliability.



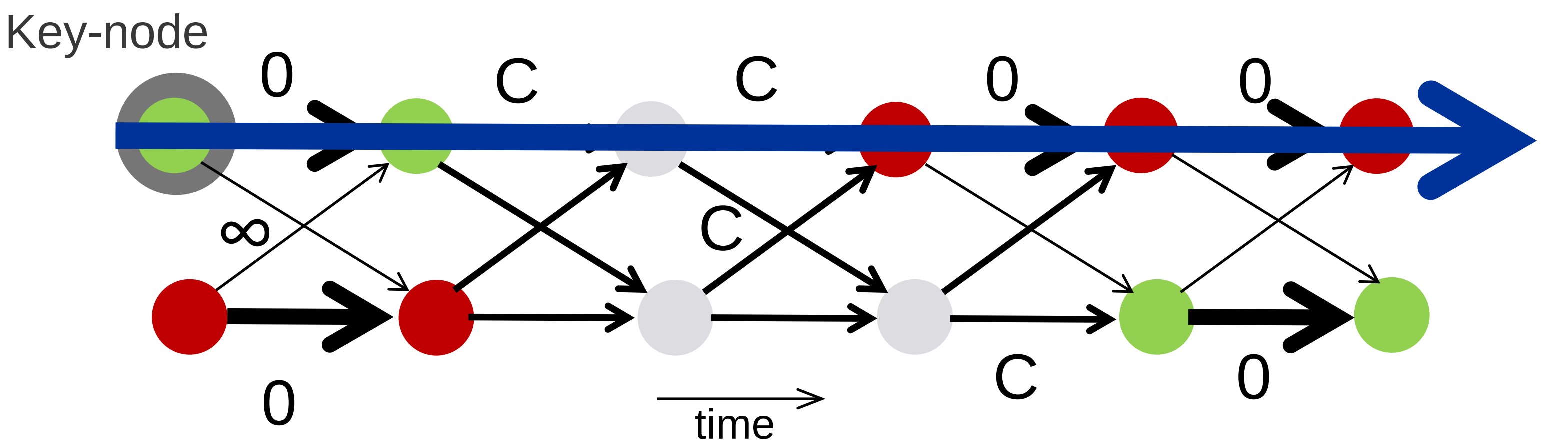
In practice, appearance features are noisy and/or sporadic



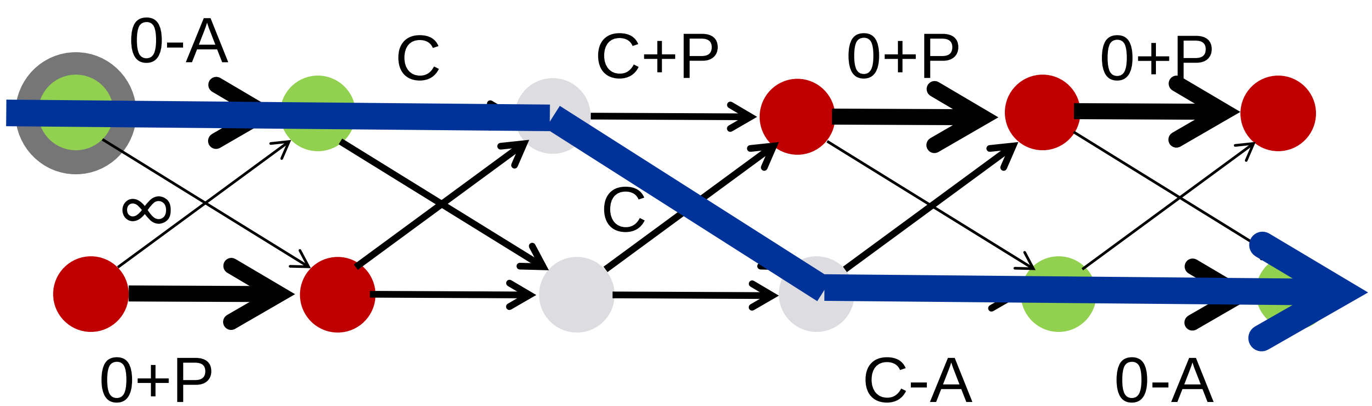
Despite they are only sporadically available/reliable, color and digit features are however crucial in preventing incompatible associations.

C. Verleysen and C. De Vleeschouwer, « Recognition of sport players' numbers using fast color segmentation », IS&T | SPIE Electronic Imaging 2012, California, USA, January 2012

Why is it important to exploit sporadic features?



Solution ? Assume that target appearance = Key-node appearance



Outline

- (Detection-based) tracking motivation
- Graph-based tracking formalisms
  - Shortest path
  - K-shortest path
  - Iterative hypothesis testing
  - Label propagation

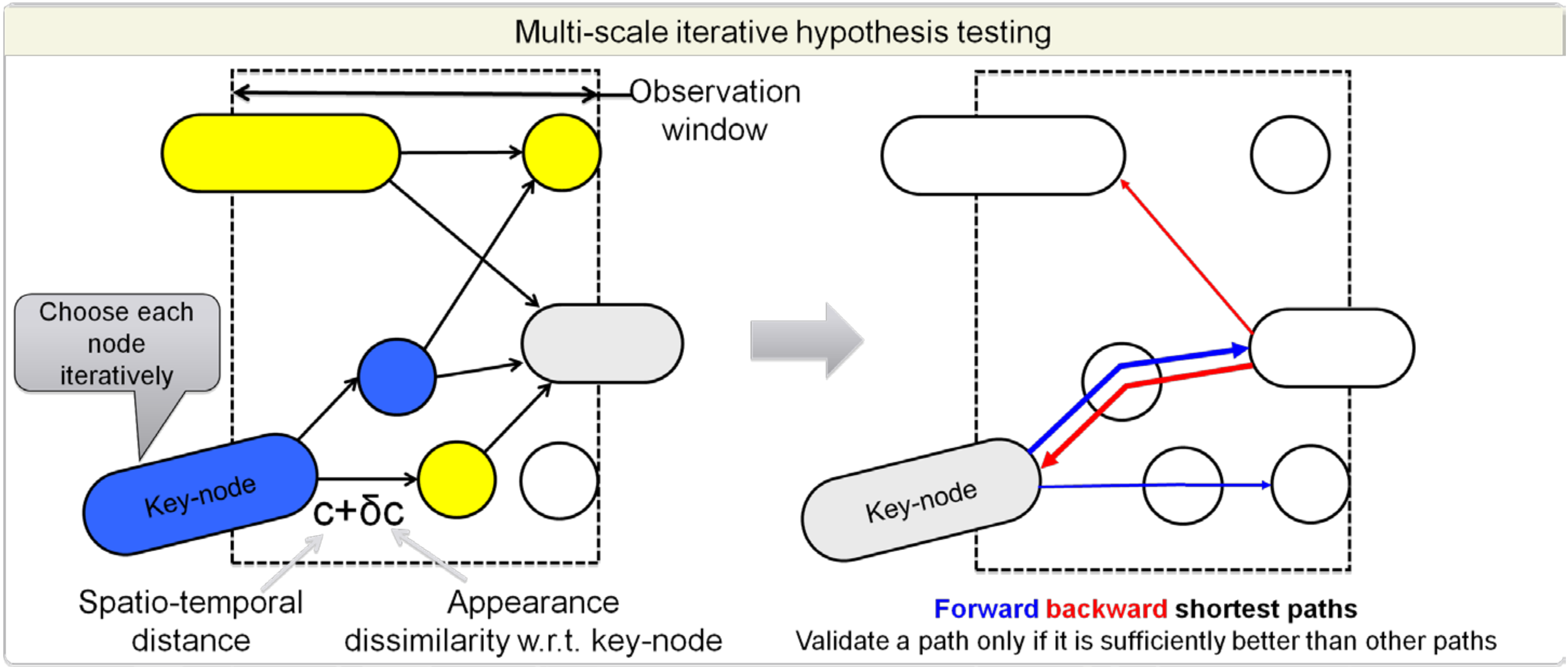
Iterative hypothesis testing paradigm(1/2)

- Investigate aggregation opportunities **in an observation window around a key-node**
  - Key-node are selected through random graph scanning;
  - Hypothesis: Target appearance = Key-node appearance;
  - Testing: compute shortest paths to/from the key-node to/from the extremities of some observation window, given the target appearance assumption;
  - To **avoid sensitivity to key-node selection order**: Aggregate the shortest path into a single node, named *tracklet*, **only if** alternative paths are sufficiently worse. Note:

Amit Kumar K.C., and al., "Aggregation of Local Shortest Paths for Multiple Object Tracking with Noisy/Missing Appearance Features", Asian Conference on Computer Vision (ACCV), 2012

Iterative hypothesis testing paradigm (2/2)

Hypothesis: Target appearance = Key-node appearance



- Relax the validation criterion along iterations (strict → soft). **Why?**
- Multiscale: make the window proportional to the size of the key-node.
- Give more credit to appearance features when less noisy or key-node gets bigger.

Amit Kumar K.C., and al., "Aggregation of Local Shortest Paths for Multiple Object Tracking with Noisy/Missing Appearance Features", Asian Conference on Computer Vision (ACCV), 2012

Results in a basketball players tracking context

Exploiting sporadic features makes a difference:

	IHT	K-short
MOTA	0.912	0.761
ID Switch	0.019	0.093

Misses False positives Mismatches

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

Outline

- (Detection-based) tracking motivation
- Graph-based tracking formalisms
  - Shortest path
  - K-shortest path
  - Iterative hypothesis testing
  - Label propagation

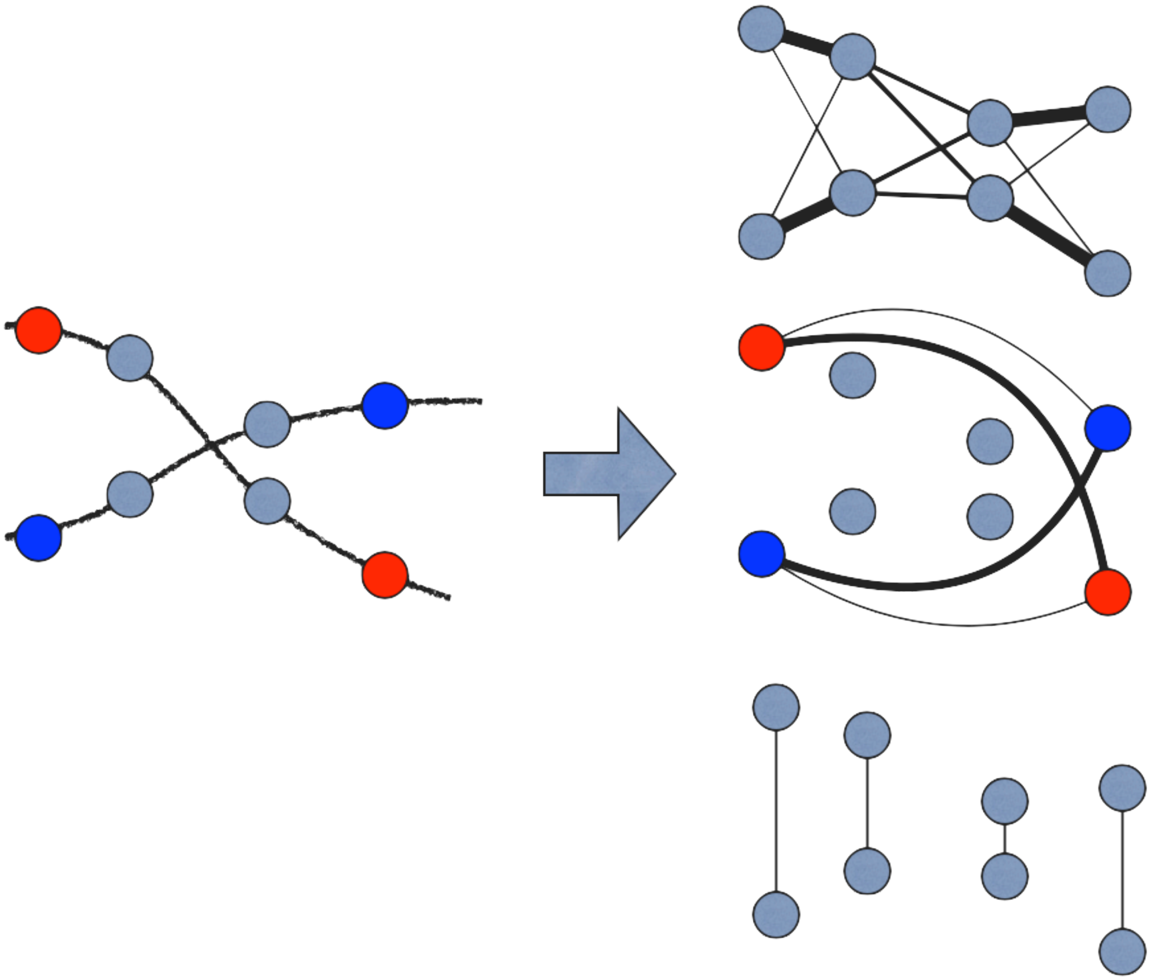
Formulating tracking as a label propagation problem (1/2)

- Given a graph $G = (V, E, W)$, we consider a label assignment $Y = (\boldsymbol{y}_1, ..., \boldsymbol{y}_{|V|})^\top$ that assigns a label distribution $\boldsymbol{y}_i \in [0, 1]^{|V|}$ to each node i. It should be noted that Y is a row-stochastic matrix, with each row summing to unity.

- The labelling error $\mathcal{E}_G(Y)$ is defined to measure the inconsistency between a label assignment Y and a graph G, namely

$$\mathcal{E}_G(Y) = \frac{1}{2}\sum_{i=1}^{|V|}\sum_{j \in \mathcal{N}_i} W_{ij} \parallel \boldsymbol{y}_i - \boldsymbol{y}_j \parallel^2$$

Graph construction for label propagation



Spatio-temporal graph G1:
- large weight to edges connecting nodes that are close in time and space.

Appearance graph G2:
- large weight to edges connecting nodes that have similar appearance.

Exclusion graph G3:
- large weight to edges connecting nodes that are detected at the same time, and thus do not correspond to the same target.

Formulating tracking as a label propagation problem (2/2)

- The optimal label assignement *Y** minimizes the labelling error associated to the spatio-temporal and the appearance graphs, while maximizing the error associated to exclusion graph. Hence,

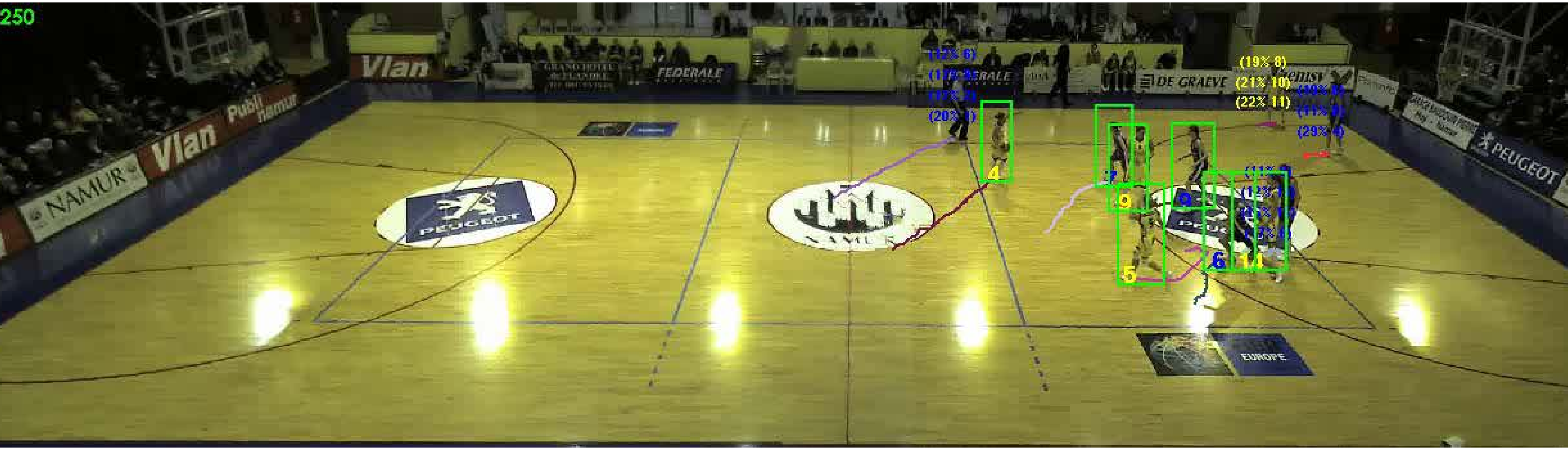
$$Y^* = \underset{Y}{\operatorname{argmin}}\, \underbrace{\varepsilon_{G1}(Y) + \varepsilon_{G2}(Y)}_{f(Y)} - \underbrace{\varepsilon_{G3}(Y)}_{g(Y)}$$

- The objective function is a difference of convex functions -> DC program, solved iteratively by linearizing *g*, and solving the resulting convex problem using the (generalized) gradient descent (= projected gradient method)

$$Y^{(k+1)} = \operatorname{argmin}_Y [\ f(Y) - \nabla g^\top \left(Y^{(k)}\right) Y]$$

Amit Kumar K.C. and al. Discriminative label propagation for multi-object tracking with sporadic appearance features. IEEE TPAMI 2017.

Evaluation



	IHT	DLP	K-short
MOTA	0.92	0.87	0.76
ID Switch	12	27	110

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

Conclusions

- Shortest path(s) computation sounds natural and intuitive. It results in efficient solution (Dijkstra), but requires a tricky hypothesis testing procedure to handle sporadic/noisy appearance features.

- Label propagation offers an elegant framework to exploit different kinds of cues, ranging from spatiotemporal to sporadic and noisy appearance features. However, it requires to optimize a difference of convex functions, and appears to be sensitive to false positive detections (because those false positives are erroneously used to replace missed ones).