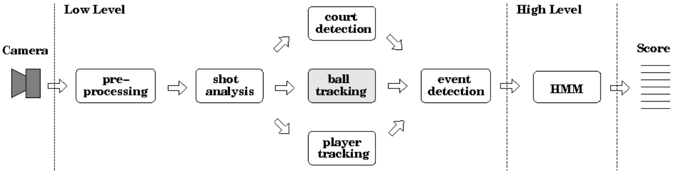
Tennis Ball Tracking

Fei Yan

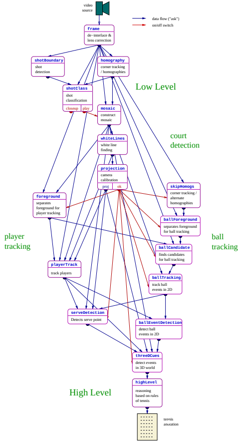
ACASVA Project Meeting
June 24, 2010
University of Surrey, Guildford

Tennis understanding system

- A system that "understands" tennis
- High level reasoning relies on low level processing:
  - court line detection / court model fitting
  - ball tracking
  - player tracking / player action recognition



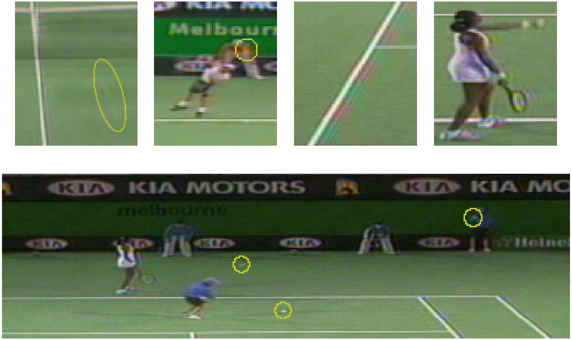
Tennis understanding system



Difficulties of tennis ball tracking

- Small size, high velocity
- Motion deformation, motion blur
- Occlusion in successive frames
- Abrupt change of motion direction
- Camera pan, tilt and zoom (PTZ)
- Monocular sequences

Difficulties of tennis ball tracking



Overall strategy

- Track after detection (TAD) and track before detection (TBD)
    - TAD: small objects with simple representations
    - TBD: large objects with complex representations
- A TAD approach can be decomposed into:
    - candidate detection
    - data association
- Data association is the key in tennis ball tracking

Ball candidate detection

- De-interlace, geometric distortion correction
- Homography computation and motion conpensation
- Motion-compensated frame differencing for motion segmentation
- An example sequence
- Motion segmentation results on the example sequence

Layered data association - candidate level association

- Candidate level association: an example
- Fitting a constant acceleration dynamic model:

$$\mathbf{v}_1 = \frac{\mathbf{z}_2 - \mathbf{z}_1}{\Delta k_{21}} - \frac{\Delta k_{21} \times \mathbf{a}}{2} \quad (1)$$

$$\mathbf{a} = 2 \times \frac{\Delta k_{21} \times (\mathbf{z}_3 - \mathbf{z}_2) - \Delta k_{32} \times (\mathbf{z}_2 - \mathbf{z}_1)}{\Delta k_{21} \times \Delta k_{32} \times (\Delta k_{21} + \Delta k_{32})} \quad (2)$$

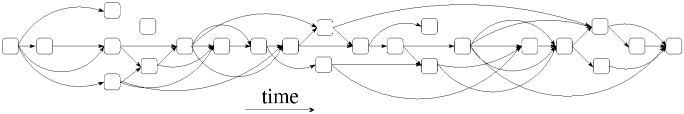
- Evaluating the quality of a model

$$C = \sum_{k=i-V}^{i+V} \sum_{j} \rho(\mathbf{z}_k^j) \quad (3)$$

$$\rho(\mathbf{z}_k^j) = \begin{cases} d^2(\hat{\mathbf{z}}_k, \mathbf{z}_k^j) & \text{if } d(\hat{\mathbf{z}}_k, \mathbf{z}_k^j) < d_{th} \\ d_{th}^2 & \text{if } d(\hat{\mathbf{z}}_k, \mathbf{z}_k^j) \geq d_{th} \end{cases} \quad (4)$$

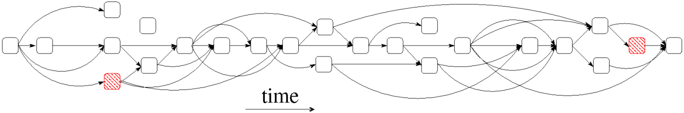
Layered data association - tracklet level association

- All tracklets in the example sequence
- Tracklet level association: shortest path problem in a directed and edge-weighted graph
  - Each node is a tracklet
  - Edge weight reflects the "compatibility" of the two nodes
  - Desired data association contained in the shortest path
- Final tracking results after event detection



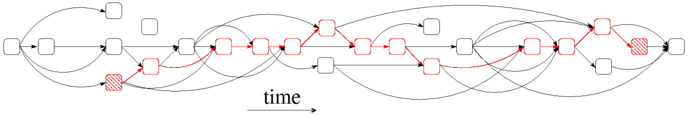
Layered data association - tracklet level association

- All tracklets in the example sequence
- Tracklet level association: shortest path problem in a directed and edge-weighted graph
  - Each node is a tracklet
  - Edge weight reflects the "compatibility" of the two nodes
  - Desired data association contained in the shortest path
- Final tracking results after event detection



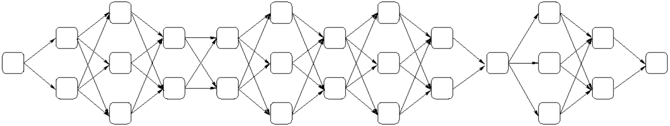
Layered data association - tracklet level association

- All tracklets in the example sequence
- Tracklet level association: shortest path problem in a directed and edge-weighted graph
  - Each node is a tracklet
  - Edge weight reflects the "compatibility" of the two nodes
  - Desired data association contained in the shortest path
- Final tracking results after event detection



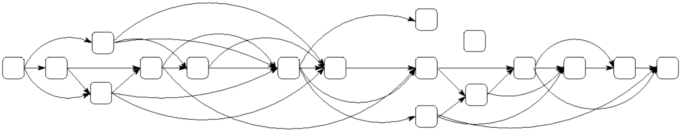
Layered data association - an efficient APSP algorithm

- General-purpose APSP algorithms:
  - Applying Dijkstra's algorithm repeatedly: $O(N^3)$
  - The Floyd-Warshall algorithm: $O(N^3)$
  - Johnson's algorithm: $O(N^2 \log N + NE)$
- Trellis graph: $\forall e_{u,v}^{p,q} \in \mathcal{E},\ u+1=v$
- $\mathcal{G}$ has a special topological property: $\forall e_{u,v}^{p,q} \in \mathcal{E},\ u<v$

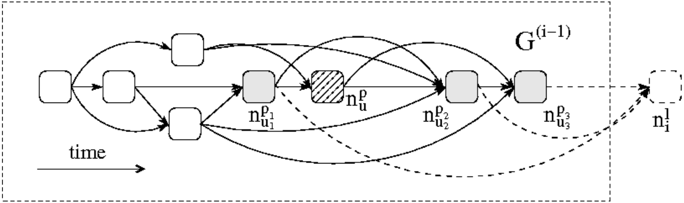


Layered data association - an efficient APSP algorithm

- General-purpose APSP algorithms:
  - Applying Dijkstra's algorithm repeatedly: $O(N^3)$
  - The Floyd-Warshall algorithm: $O(N^3)$
  - Johnson's algorithm: $O(N^2 \log N + NE)$
- Trellis graph: $\forall e_{u,v}^{p,q} \in \mathcal{E},\ u + 1 = v$
- $\mathcal{G}$ has a special topological property: $\forall e_{u,v}^{p,q} \in \mathcal{E},\ u < v$



Layered data association - an efficient APSP algorithm



- Based on dynamic programming
  - Assume APSP in $\mathcal{G}^{(i-1)}$ is solved
  - Solve APSP in $\mathcal{G}^{(i)}$ using the APSP results in $\mathcal{G}^{(i-1)}$
- Complexity: $O(N^2)$

Future work

- Doubles game?
  - More abrupt motion change and more clutter
  - Improve the robustness of the current tracker
- Other ball games?
  - Badminton? Table tennis?