The Long-Term Tracking

The TLD (PN) Long-Term Tracker

includes:

- adaptive tracker(s) (FOT)
- object detector(s)
- P and N event recognizers for unsupervised learning generating (*possibly incorrectly)* labelled samples
- an (online) supervised method that updates the detector(s)

Operation:

1. Train **Detector on** the first patch
2. Runs **TRACKER** and **DETECTOR** in parallel
3. Update the object **DETECTOR using P-N learning**

TLD a.k.a. PN Tracker a.k.a. “The Predator”

http://kahlan.eps.surrey.ac.uk/featurespace/tld/

Z. Kalal, K.Mikolajczyk, J. Matas: Tracking-Learning-Detection. IEEE T PAMI 34(7): 1409-1422 (2012)

Implementation of TLD

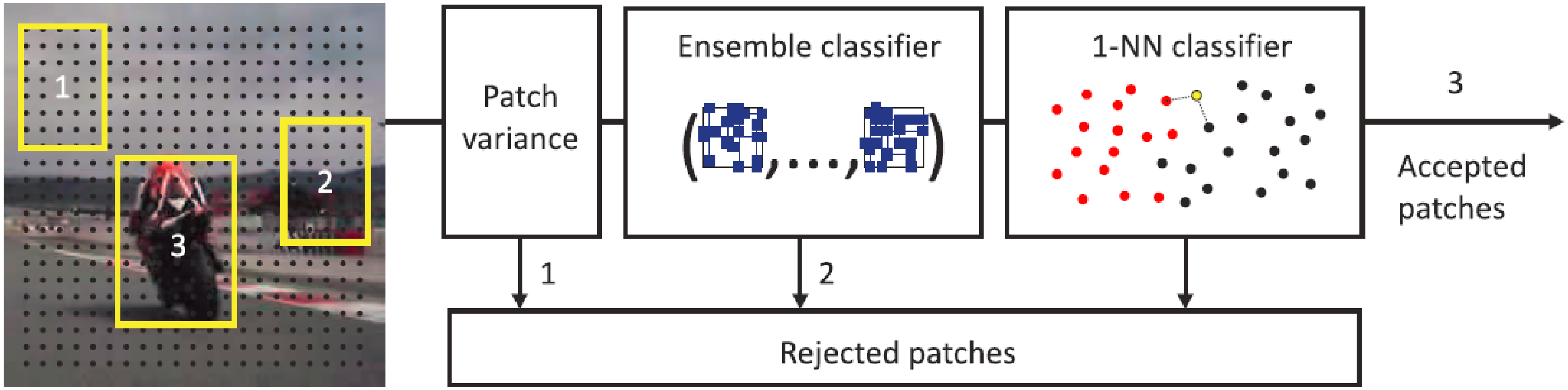
- Object instance
  - Bounding box, fixed aspect ratio (location and scale)
  - Represented by a patch (normalized to 15x15 px)
- Similarity measured by Normalized Cross Correlation
- Object Model
  - Set of positive and set of negative patches
  - Model is updated on-line

- Integrator
  - Tracker and Detector run in parallel
  - Output is the patch with maximum confidence
  - The object in not visible state (if both tracker and detector provides a bounding box)

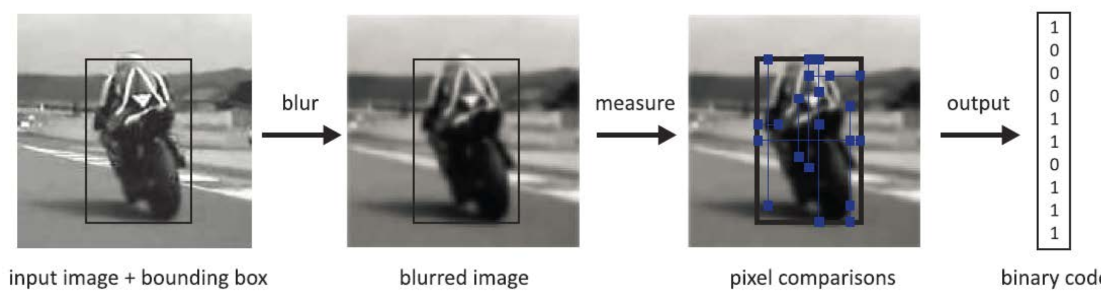
- Initialization
  - Positive patches generated by “data augmentation” of the input bounding-box (shifts, small in-plane rotations, scales)
  - Negative patches - samples from the surrounding background

Object Detector

- Scanning window process
- Sequential decision process (3 stages)



- Variance filter
  - Implemented by integral image Var X = EX^2 – (EX)^2
  - Typically rejects 50% of patches
- Ensemble classifier
  - Random Ferns

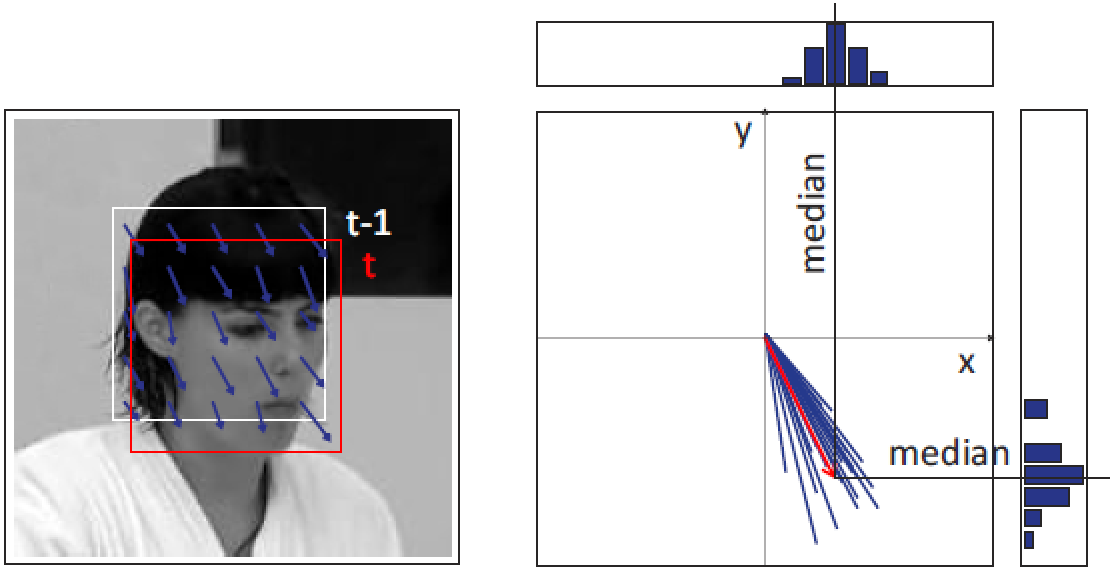


- NN classifier
  - Typically decides only 50 remaining patches

Tracker

- Median Flow

  Median-shift tracker - tracker of a rectangle, based on the Lucas-Kanade tracker, robust to partial occlusions. Estimates translation and scale.



Sparse motion field

Object displacement

- Failure detection based on robust dispersion of individual LK tracks

  median |di – d| > 10 px*

Learning

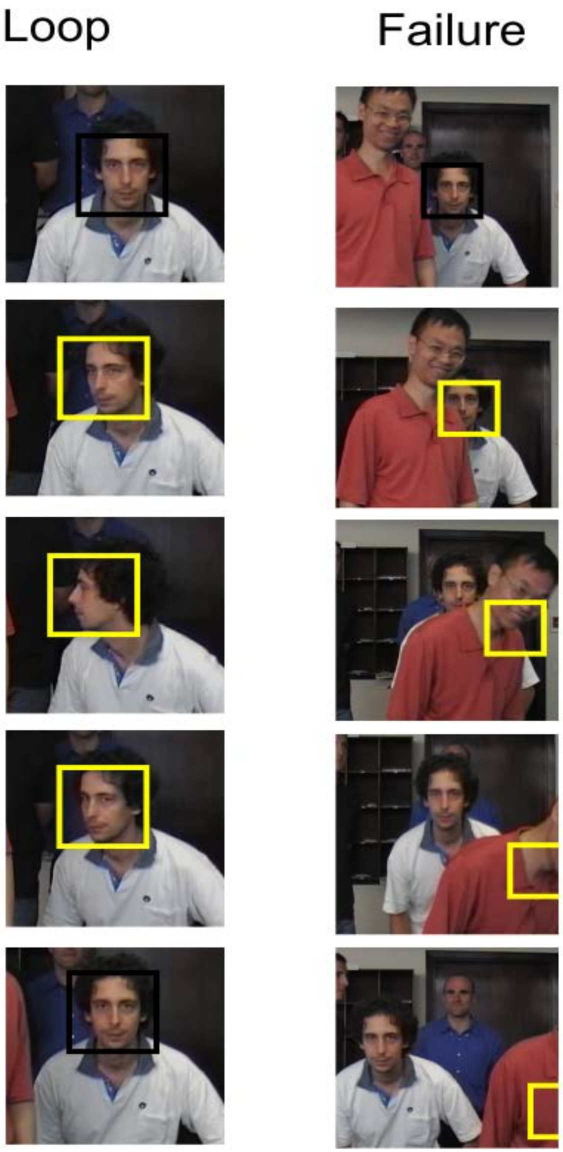
- The object model is updated using P and N experts
- P-expert
  - adds positive examples (detects false negative)
- N-expert
  - adds negative examples (detects false positive)

- Both Tracker and Detector are not error free
  - The system can recover from errors to a large extent
  - Possible errors are are mutually compensated

P-event: “Loop”

- exploits **temporal** structure
- turns drift of adaptive trackers into advantage
- **Assumption:**
  If an adaptive tracker fails, it is unlikely to recover.
- **Rule:**
  Patches from a track starting and ending in the current model (black), ie. are validated by the detector, are added to the model

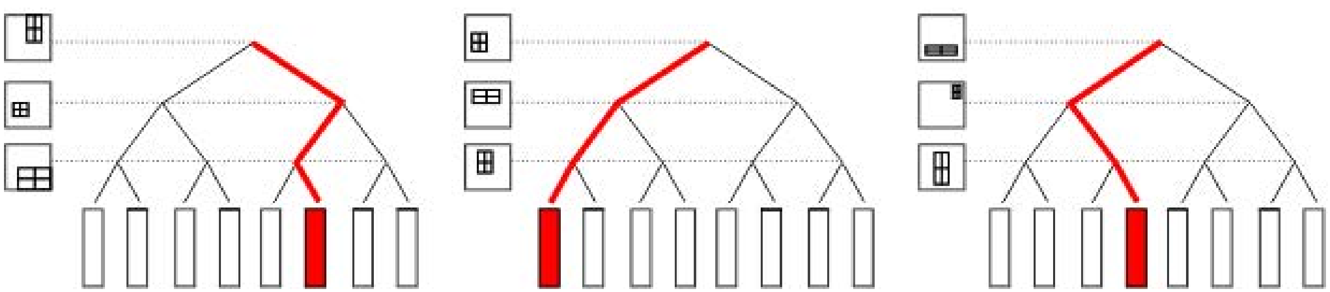
Tracker responses



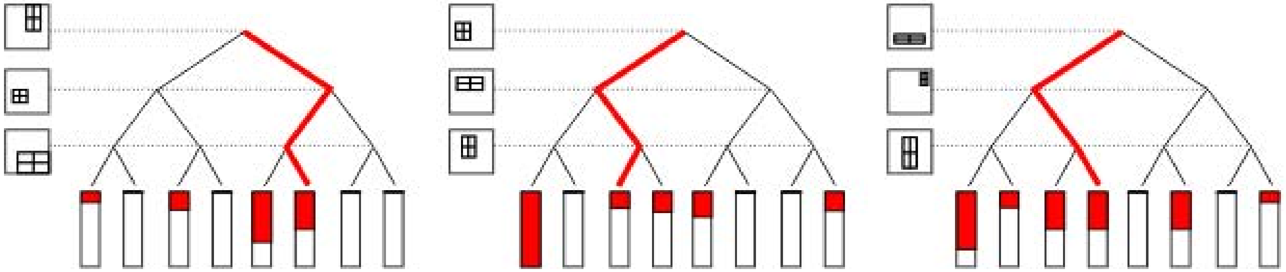
N-event: Uniqueness Enforcement

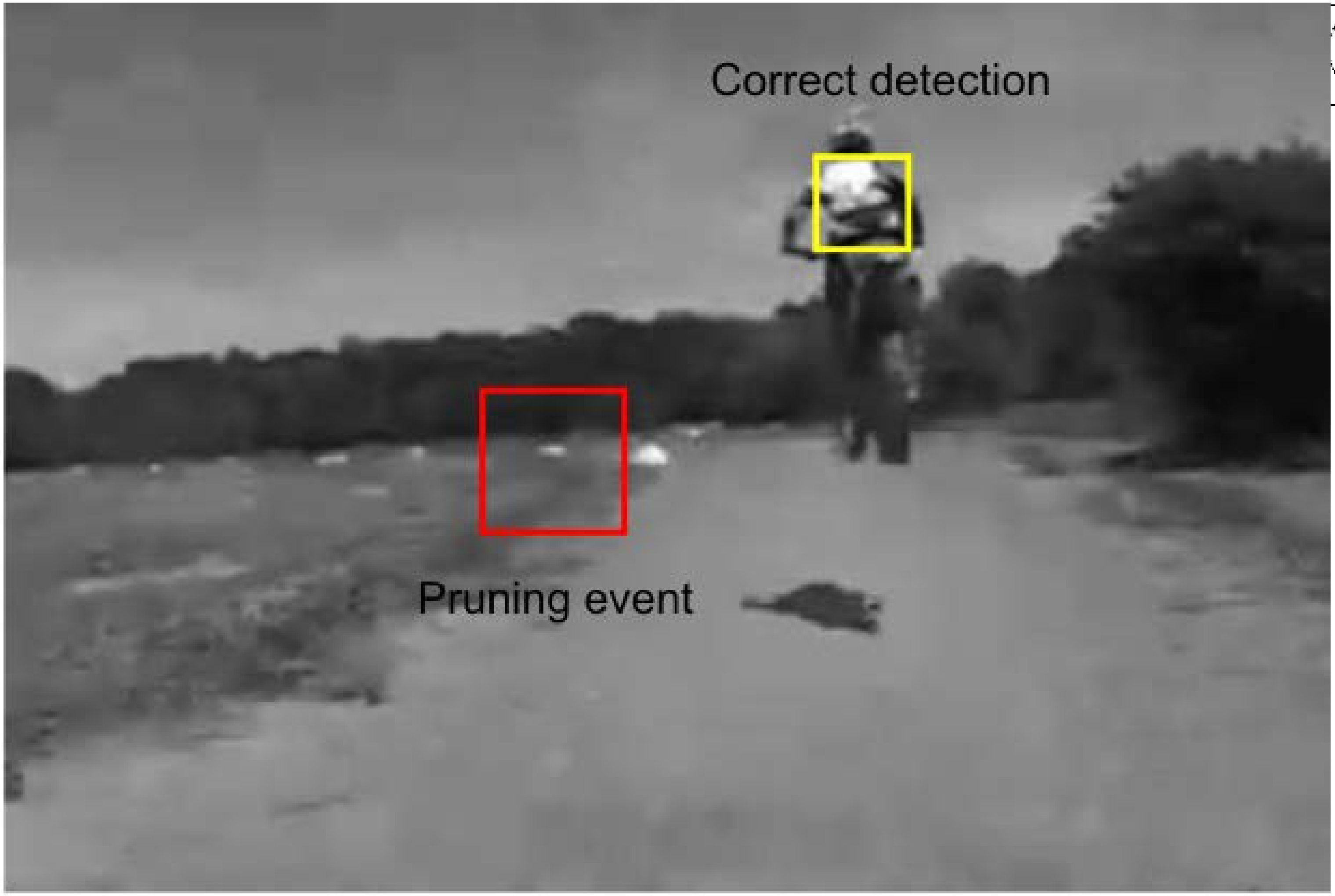
- exploits **spatial** structure
- **Assumption:**
  Object is unique in a single frame.
- **Rule:**
  If *the tracker is in model*, all other detections within the current frame (red) are assumed wrong $\rightarrow$ prune from the model

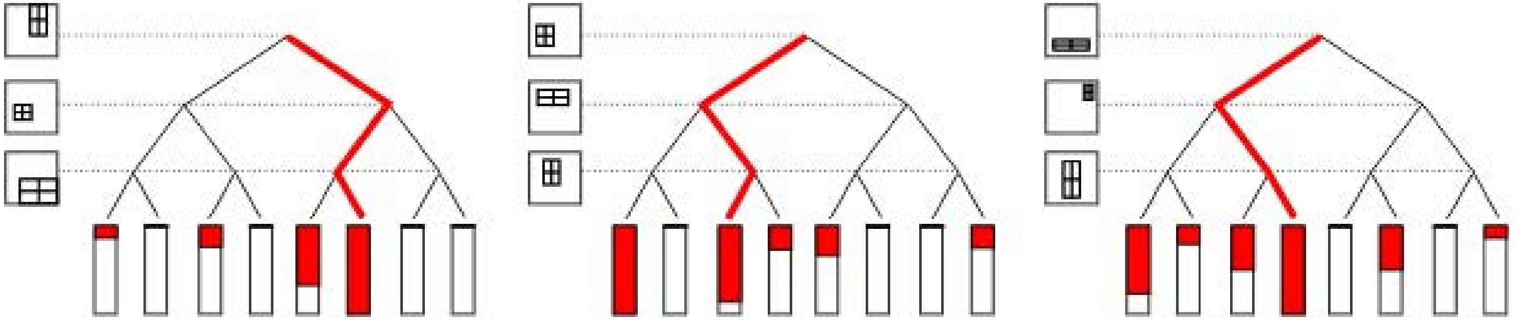

Growing event




Correct detection




Correct detection
Pruning event



2