Robust Short- and Long-Term Visual Tracking

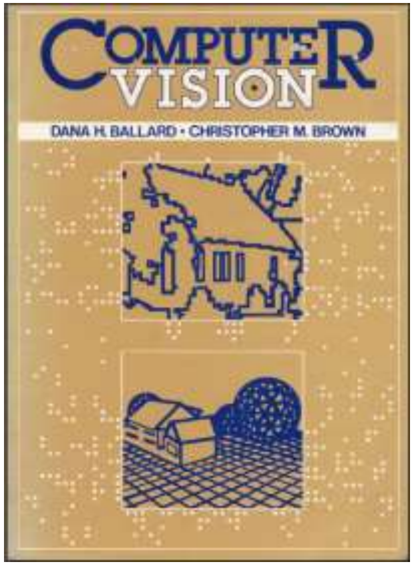
Jiri Matas

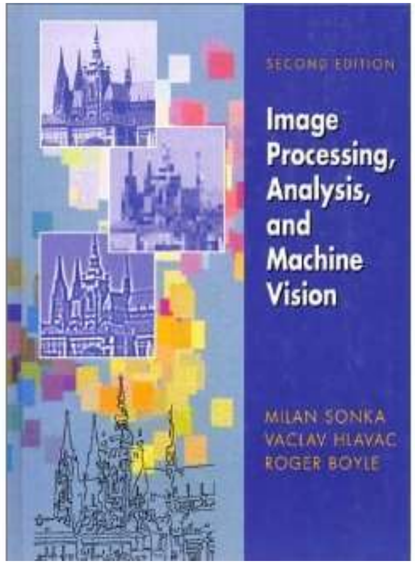
Center for Machine Perception
Department of Cybernetics,

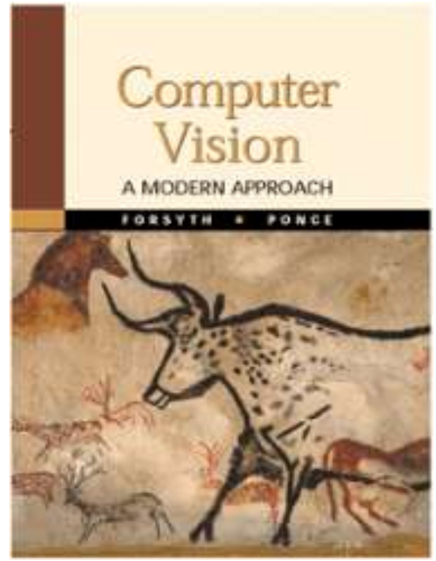
Faculty of Electrical Engineering
Czech Technical University,

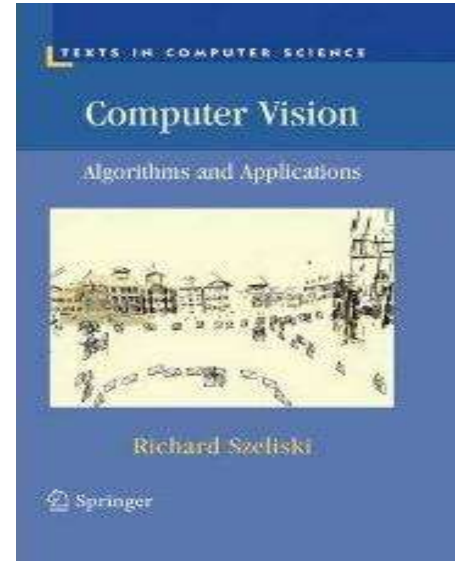
Prague, Czech Republic

Tracking: Definition - Literature









Surprisingly little is said about tracking in standard textbooks.
Limited to optic flow, plus some basic trackers, e.g. Lucas-Kanade.

Definition (0):
[Forsyth and Ponce, *Computer Vision: A modern approach*, 2003]

> *"Tracking is the problem of generating an inference about the motion of an object given a sequence of images.*
> *Good solutions of this problem have a variety of applications…"*

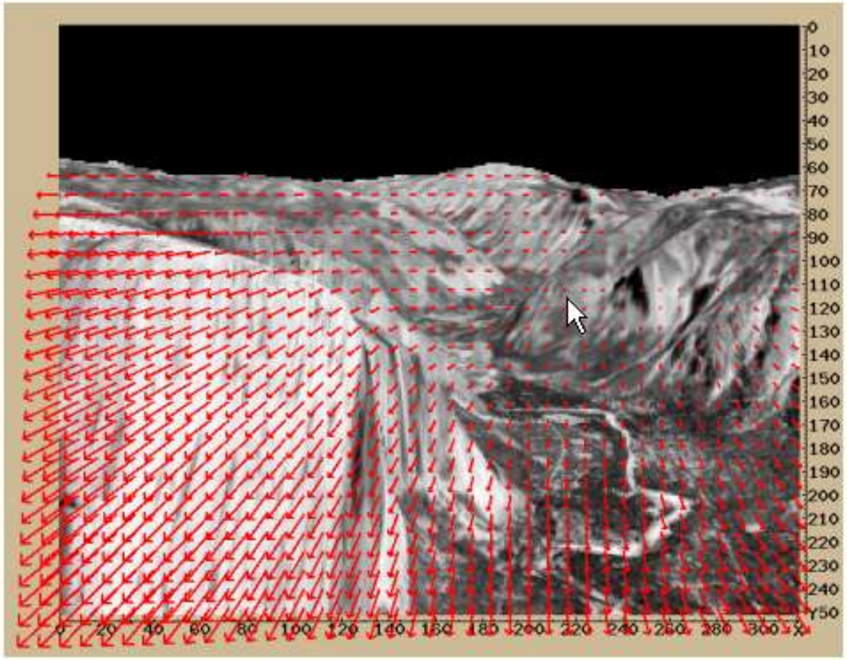
Optic Flow v. Tracking

- At every pixel, 2D displacement is estimated (dense result)
- Problem 1: occlusion, pixels visible in one image only
  - in the standard formulation, “no” is not an answer
- Problem 2: is the ground truth ever known?
  - performance evaluation problematic (synthetic sequences ..)
- Problem 3: requires regularization (smoothing)
- Problem 4: failure not easy to detect
- Problem 5: historically, very slow

However:

- Recent surge in interest, real-time on GPU, some robustness achieved
- Applications: time-to-contact, ego-motion

Tracking v. Optic Flow, Motion Estimation



Yosemite sequence real flow

Definition (1a): Tracking

Establishing point-to-point correspondences

in consecutive frames of an image sequence

Notes:

- The concept of an "object" in F&P definition disappeared.
- If an algorithm correctly established such correspondences, would that be a perfect tracker?
- tracking = motion estimation?

Definition (1a): Tracking

Establishing point-to-point correspondences

in consecutive frames of an image sequence

Notes:

- The concept of an “object” in F&P definition disappeared.
- If an algorithm correctly established such correspondences, would that be a perfect tracker?
- tracking = motion estimation?

Consider this sequence:

Definition (1b): Tracking

Establishing point-to-point correspondences

between all pairs frames in an image sequences

- If an algorithm correctly established such correspondences, would that be a perfect tracker?

Definition (1b): Tracking

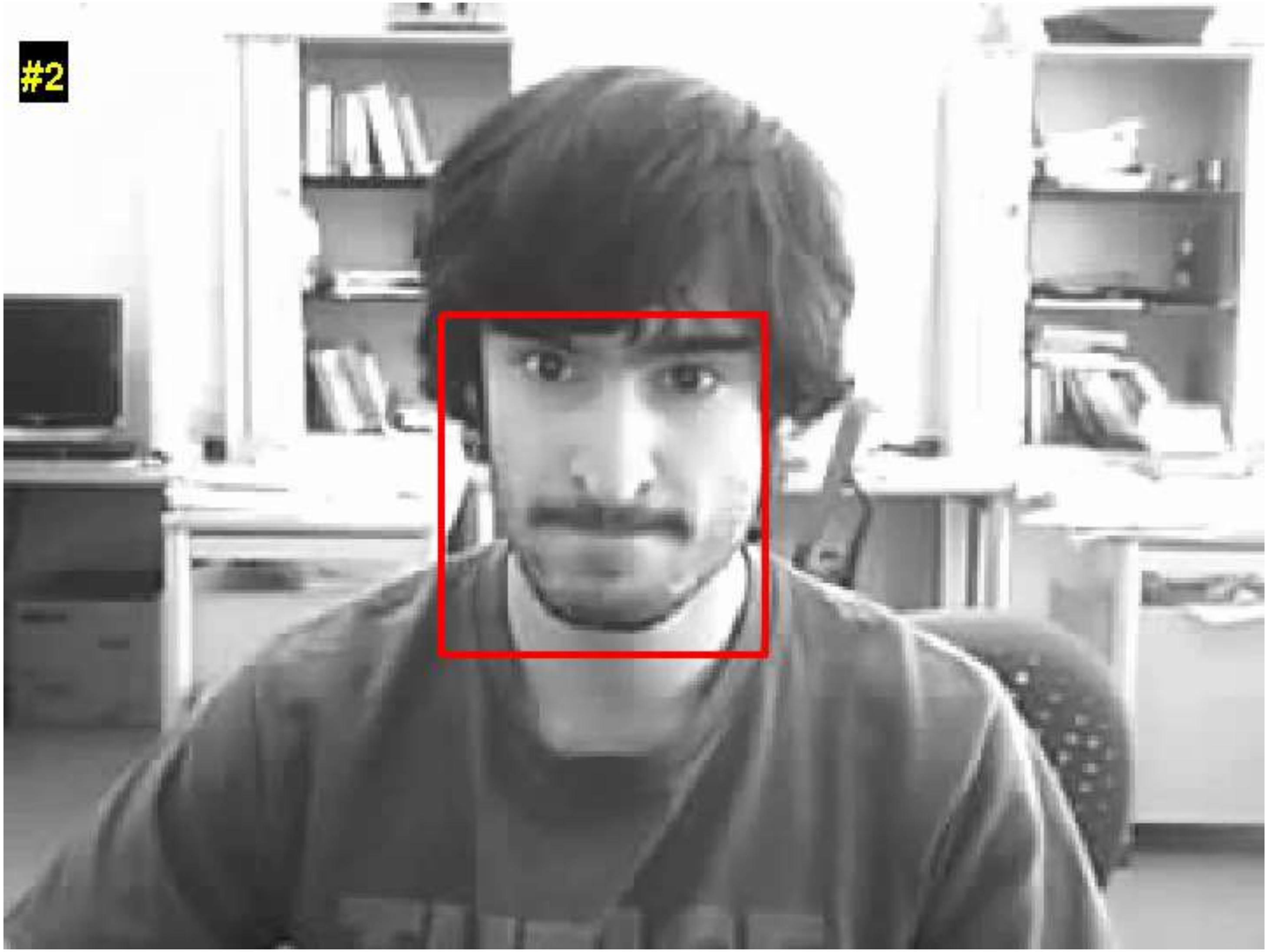
Establishing point-to-point correspondences

between all pairs frames in an image sequences

Notes:

- If an algorithm correctly established such correspondences, would that be a perfect tracker?
- rather full off-line video analysis than tracking …

A "standard" CV tracking method output



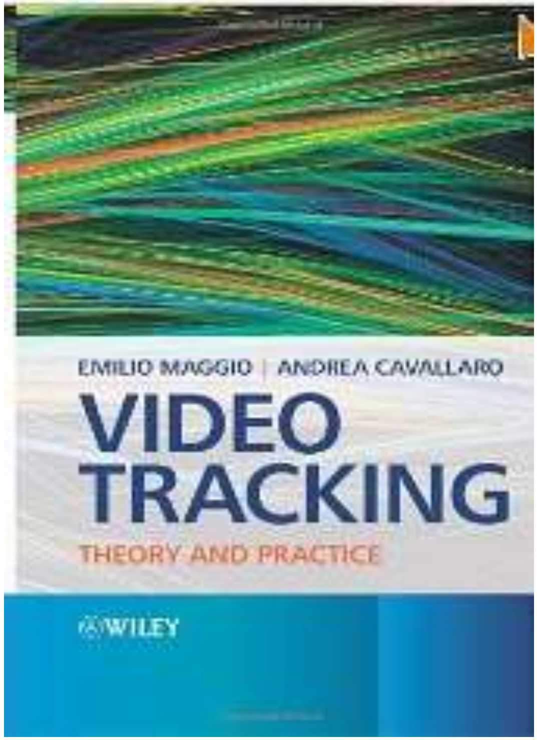
Definition (2): Tracking

Given an initial estimate of its position,
locate X in a sequence of images,

Where X may mean:

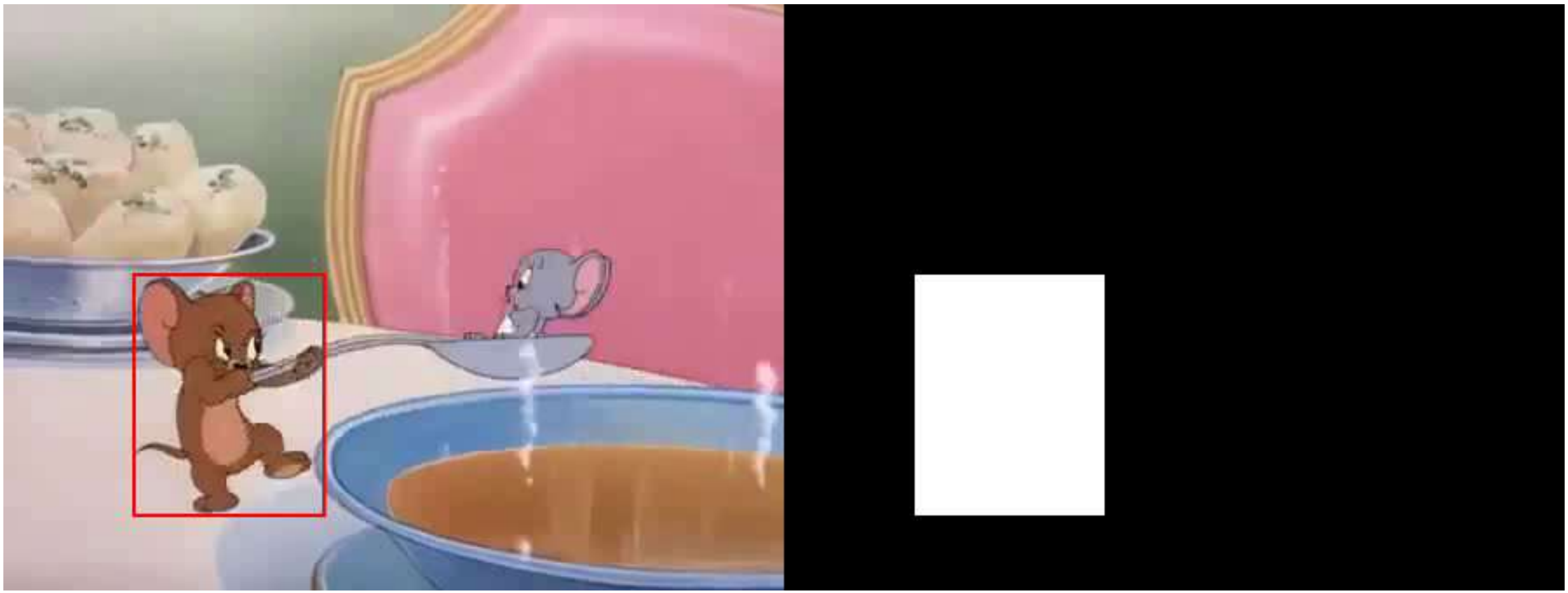
- A (rectangular) region
- An "interest point" and its neighbourhood
- An "object"

This definition is adopted e.g. in a recent book by Maggio and Cavallaro, *Video Tracking*, 2011



Smeulders T-PAMI13:
Tracking is the analysis of video sequences for the purpose of establishing the location of the target over a sequence of frames (time) starting from the bounding box given in the first frame.

Tracking as Segmentation



J. Fan et al. Closed-Loop Adaptation for Robust Tracking, ECCV 2010

Tracking-Learning-Detection (TLD)

Definition (3): Tracking

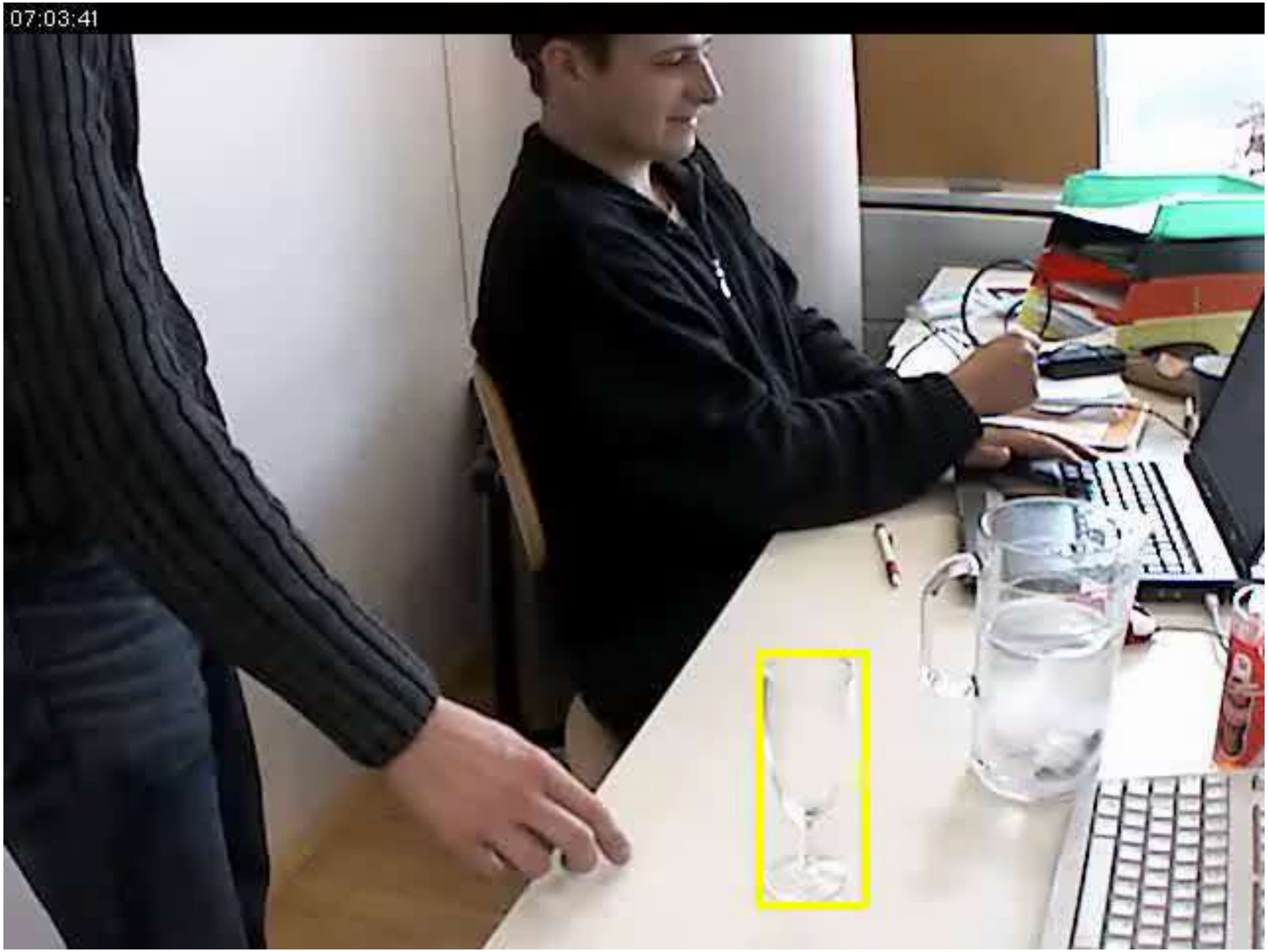
Given an initial estimate of the pose and state of X :

In all images in a sequence, (in a causal manner)

1. *estimate the pose and state of X*
2. *(optionally) update the model of X*

- Pose: any geometric parameter (position, scale, …)
- State: appearance, shape/segmentation, visibility, articulations
- Model update: essentially a semi-supervised learning problem
  - a priori information (appearance, shape, dynamics, …)
  - labeled data ("track this") + unlabeled data = the sequences
- Causal: for estimation at T, use information from time $t \leq T$

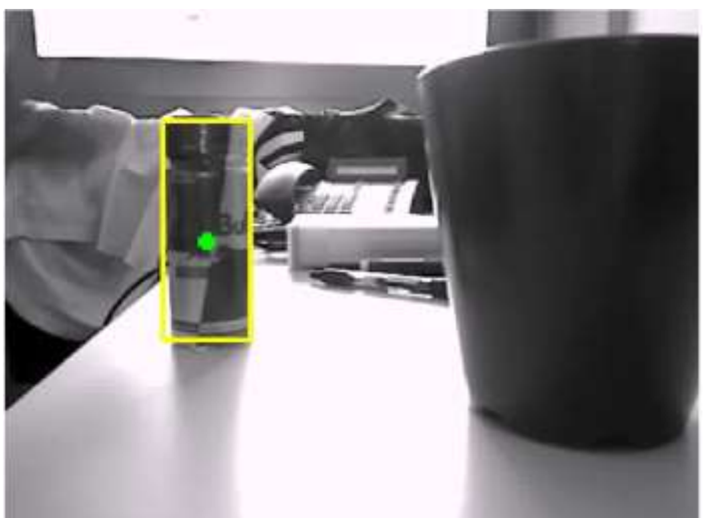
A “miracle”: Tracking a Transparent Object

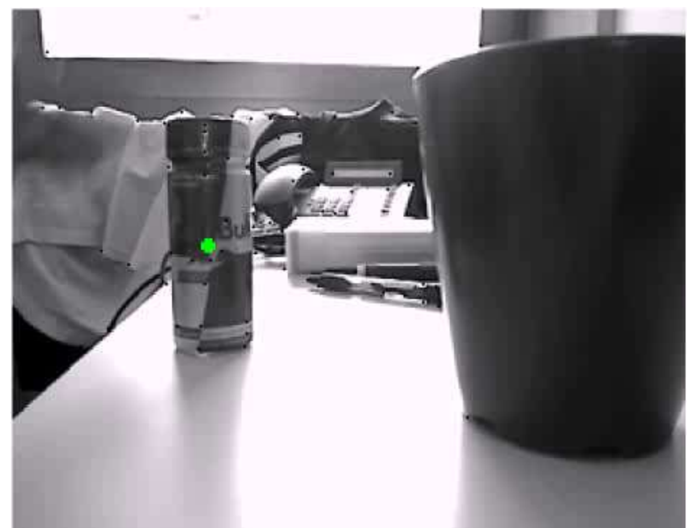


video credit: Helmut Grabner

H. Grabner, H. Bischof, On-line boosting and vision, CVPR, 2006.

Tracking the “Invisible”





H. Grabner, J. Matas, L. Gool, P. Cattin,Tracking the invisible: learning where the object might be, CVPR 2010.

Definition (4): Tracking

Given an estimate of the pose (and state) of X in “key” images

(and a priori information about X),

In all images in a sequence, (in a causal manner):

1. *estimate the pose and state of X*
2. *(optionally) estimate the state of the scene [e.g. “supporters”]*
3. *(optionally) update the model of X*

<u>Out</u>: a sequence of poses (and states),(and/or the learned model of X)

Notes:

- Often, not all parameters of pose/state are of interest, and the state is estimated as a side-effect.
- If model acquisition is the desired output, the pose/state estimation is a side-effect.
- The model may include: relational constraints and dynamics, appearance change as a function as pose and state

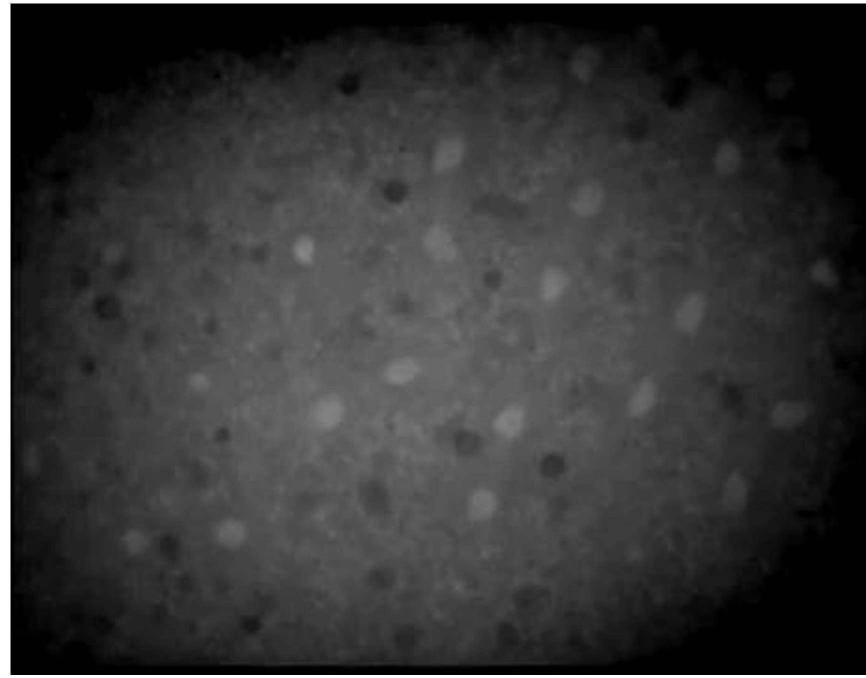
Definition (k): Tracking

http://server.cs.ucf.edu/~vision/projects/sali/CrowdTracking/index.html

…… multiple object tracking …..

Definition (n): Tracking

Cell division.
http://www.youtube.com/watch?v=rgLJrvoX_qo



Three rounds of cell division in Drosophila Melanogaster.
http://www.youtube.com/watch?v=YFKA647w4Jg

splitting and merging events ….

Short-term v. Long-term Tracking v. OF

Short-term Trackers:

- primary objective: “where is X?” = precise estimation of pose
- secondary: be fast; don’t lose track
- evaluation methodology: frame number where failure occurred
- examples: Lucas Kanade tracker, mean-shift tracker

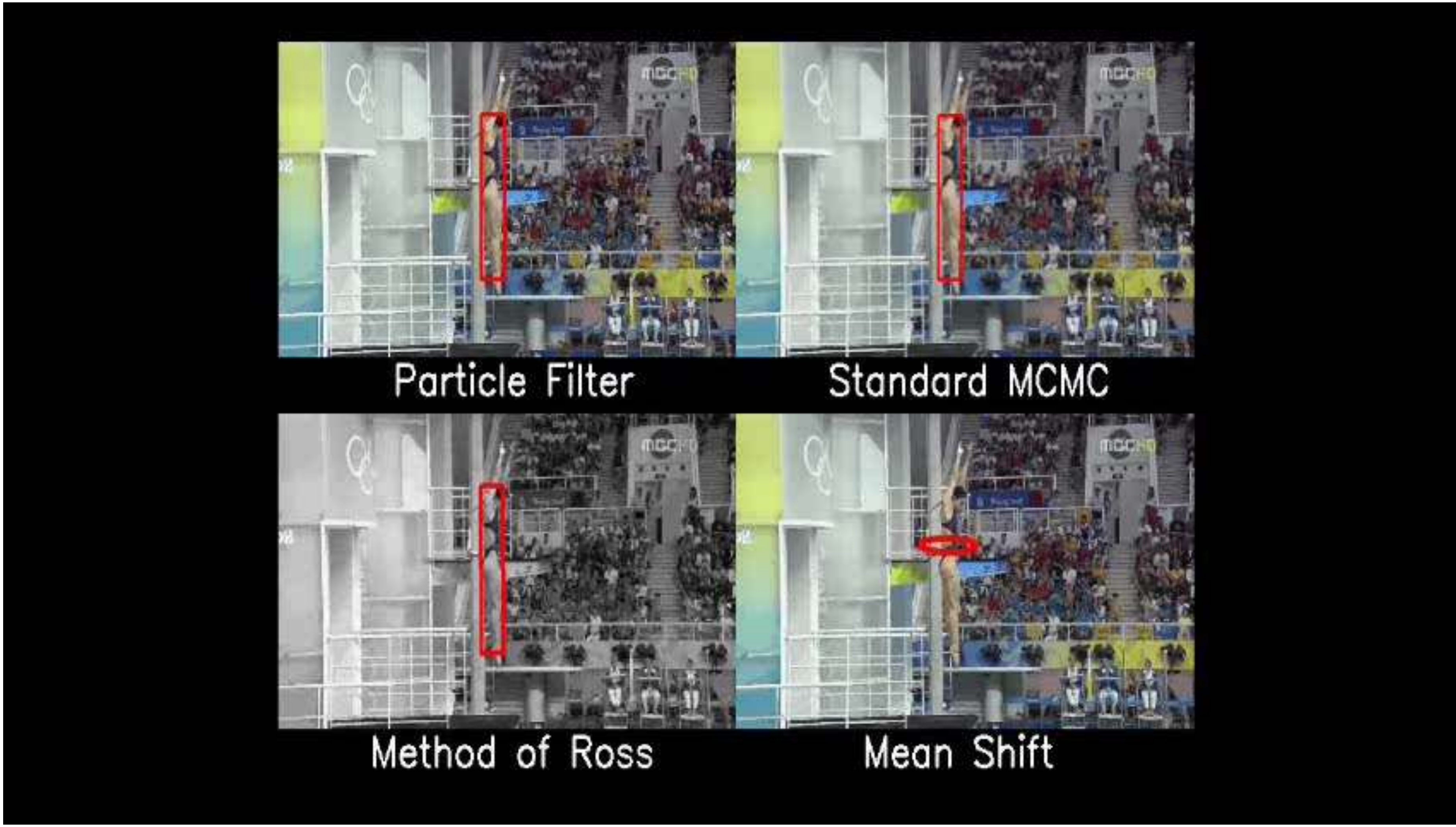
Long-term Tracker-Detectors:

- primary objective: unsupervised learning of a detector, since *every (short-term) tracker fails, sooner or later* (disappearance from the field of view, full occlusion)
- avoid the “*first failure means lost forever*” problem
- close to online-learned detector, but assumes and exploits the fact that a sequence with temporal pose/state dependence is available
- evaluation methodology: precision/recall, false positive/negative rates (i.e. like detectors)
- note: the detector part may help even for short-term tracking problems, provides robustness to fast, unpredictable motions.

Optic Flow, Motion estimation: establish all correspondences a sequence

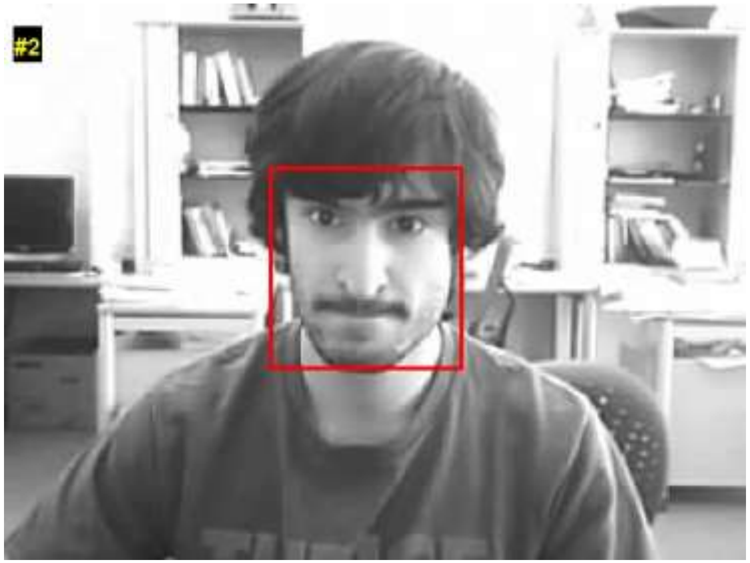
Tracking: Which methods work?

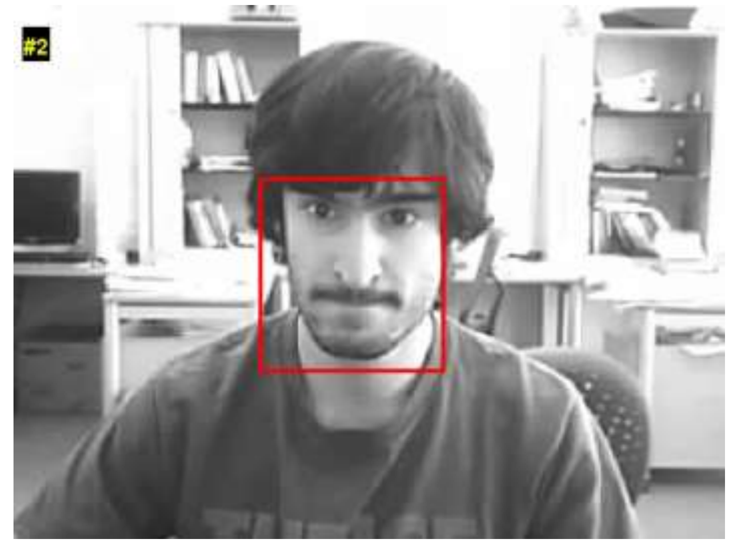
Tracking: Which methods work?

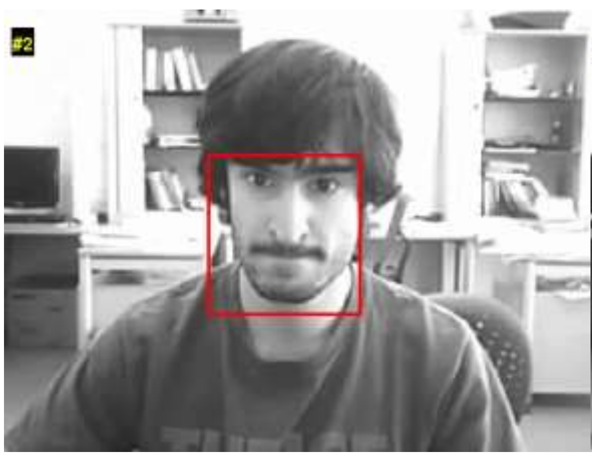


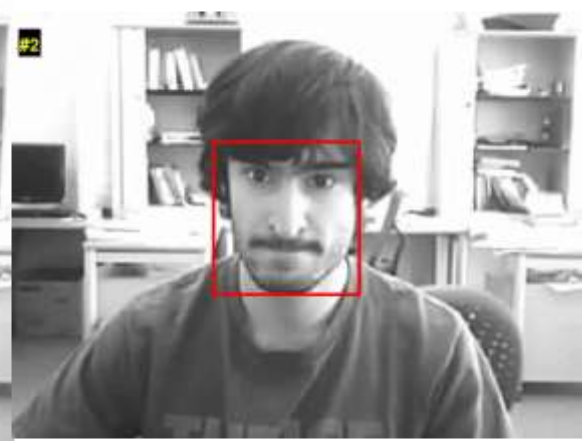
What works? “The zero-order tracker” ☺

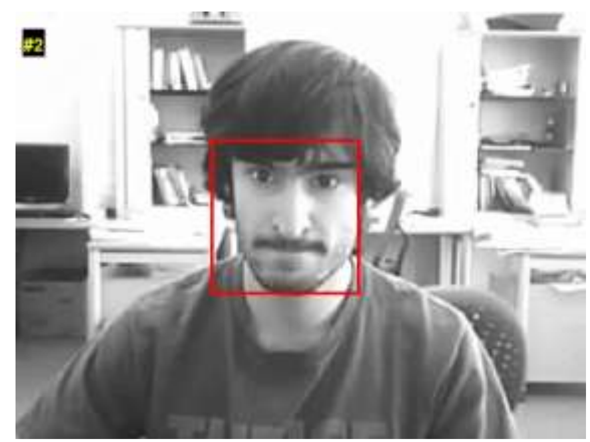
Compressive Tracker (ECCV'12). Different runs.











The Flock of Trackers – FOT

The Flock of Trackers

- A n x m grid (say 10x10) of Lucas-Kanade / ZSP trackers

- Tracker initialised on a regular grid

- Robust estimation of global, either "median" direction/scale or RANSAC (up to homography)

- Each tracker has a *failure predictor*

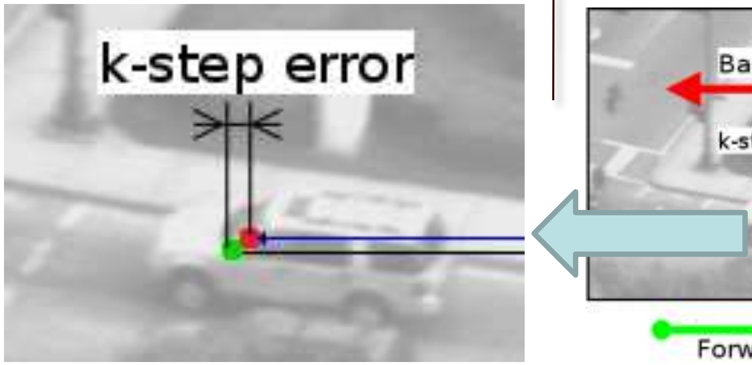
Two classical Failure Predictors

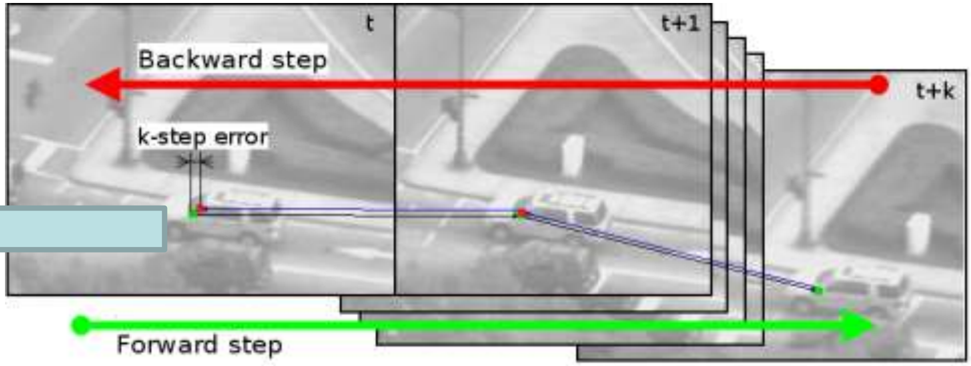
Normalized Cross-correlation

- Compute normalized cross-correlation between local tracker patch in time *t* and *t+1*
- Sort local trackers according to NCC response
- Filter out bottom 50% (Median)

Forward-Backward[1]

- Compute correspondences of local trackers from time *t* to *t+k* and *t+k* to *t* and measure the k-step error
- Sort local trackers according to the k-step error
- Filter out bottom 50% (Median)





[1] Z. Kalal, K. Mikolajczyk, and J. Matas.
Forward-Backward Error: Automatic Detection of Tracking Failures. ICPR, 2010

Failure Predictor: Neighbourhood Consistency

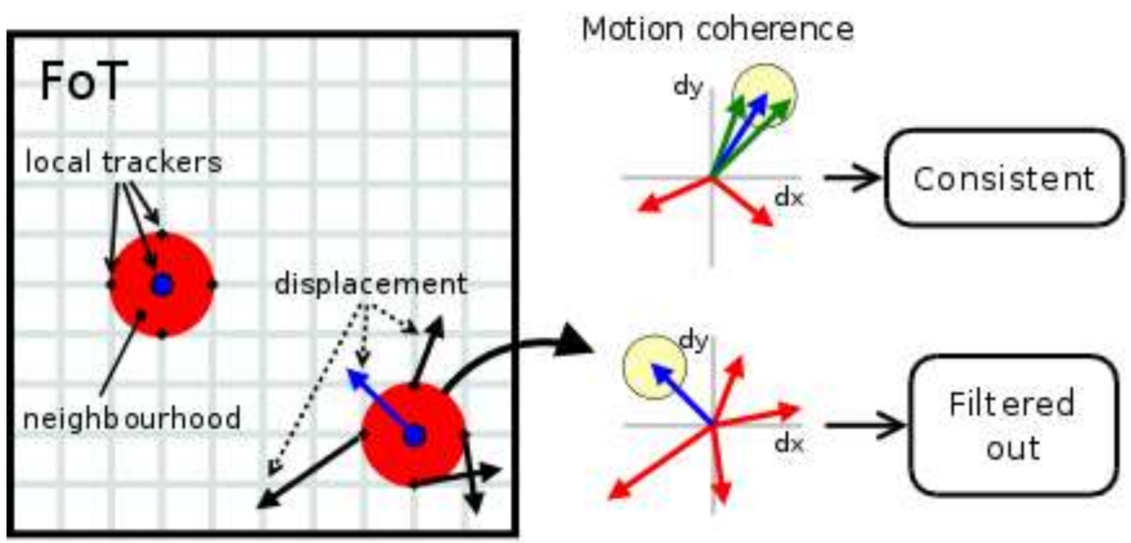
- For each local tracker *i* is computed neighbourhood consistency score as follows :

$$S_i^{Nh} = \sum_{j \in N_i} [\| \Delta_j - \Delta_i \|^2 < \varepsilon] \, , \quad \text{where} \quad [expression] = \begin{cases} 1 & \text{if } expression \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

N_i is four neighbourhood of local tracker *i*, Δ is displacement and ε is displacement error threshold

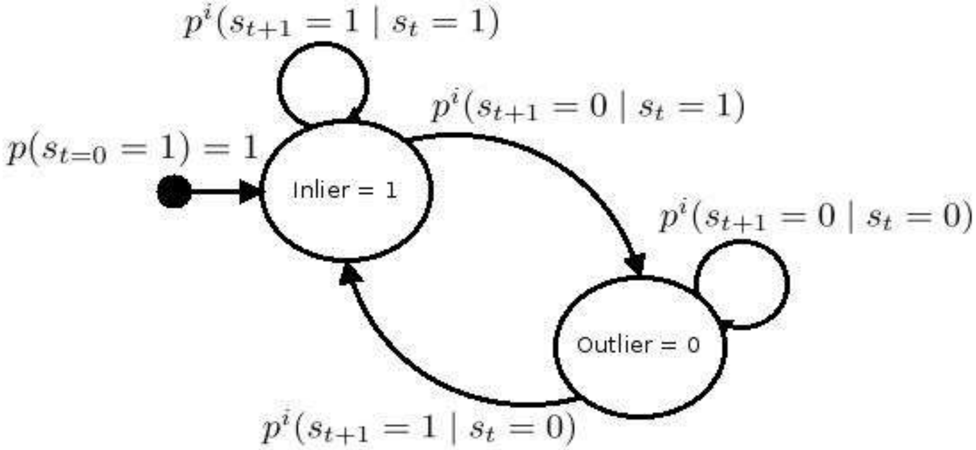
- Local trackers with $S_i^{Nh} < \Theta_{Nh}$ are filtered out

- Setting:
  $\varepsilon = 0.5$px
  $\Theta_{Nh} = 1$



Failure Predictors: Temporal consistency

- Markov Model predictor (MMp) models local trackers as two states (i.e. inlier, outlier) probabilistic automaton with transition probabilities $p^i(s_{t+1} \mid s_t)$



- MMp estimates the probability of being an inlier for all local trackers $\Rightarrow$ filter by
  1) Static threshold Θ_s
  2) Dynamic threshold Θ_r

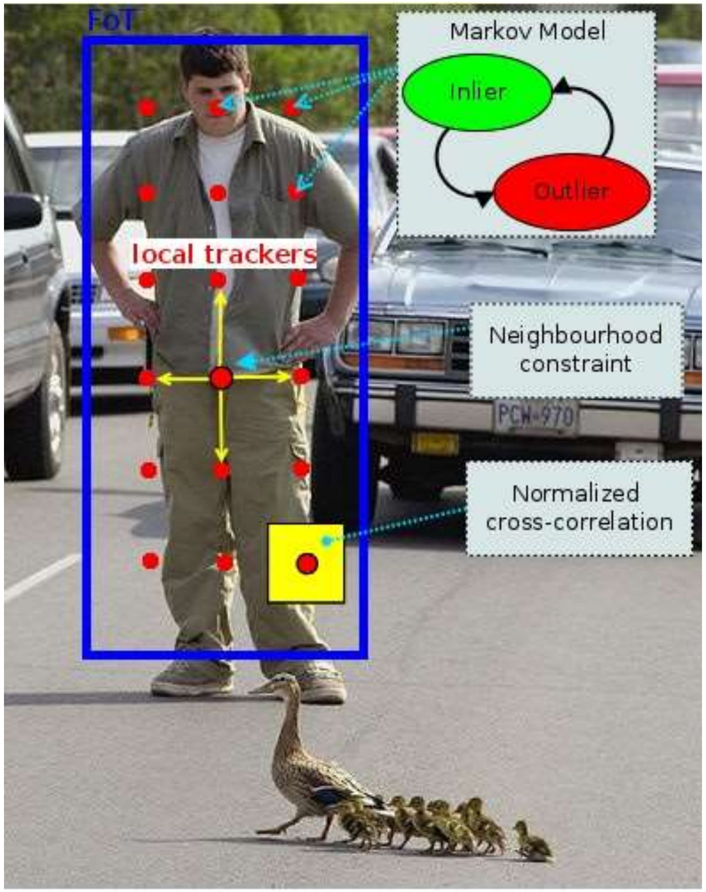
- Learning is done incrementally (learns are the transition probabilities between states)

- Can be extended by "forgetting", which allows faster response to object appearance change

The combined outlier filter Σ

Combining three indicators of failure:

- Local appearance (NCC)
- **Neighbourhood consistency (Nh)** (similar to *smoothness assumption* used in optic flow estimation)
- **Temporal consistency using a Markov Model predictor (MMp)**

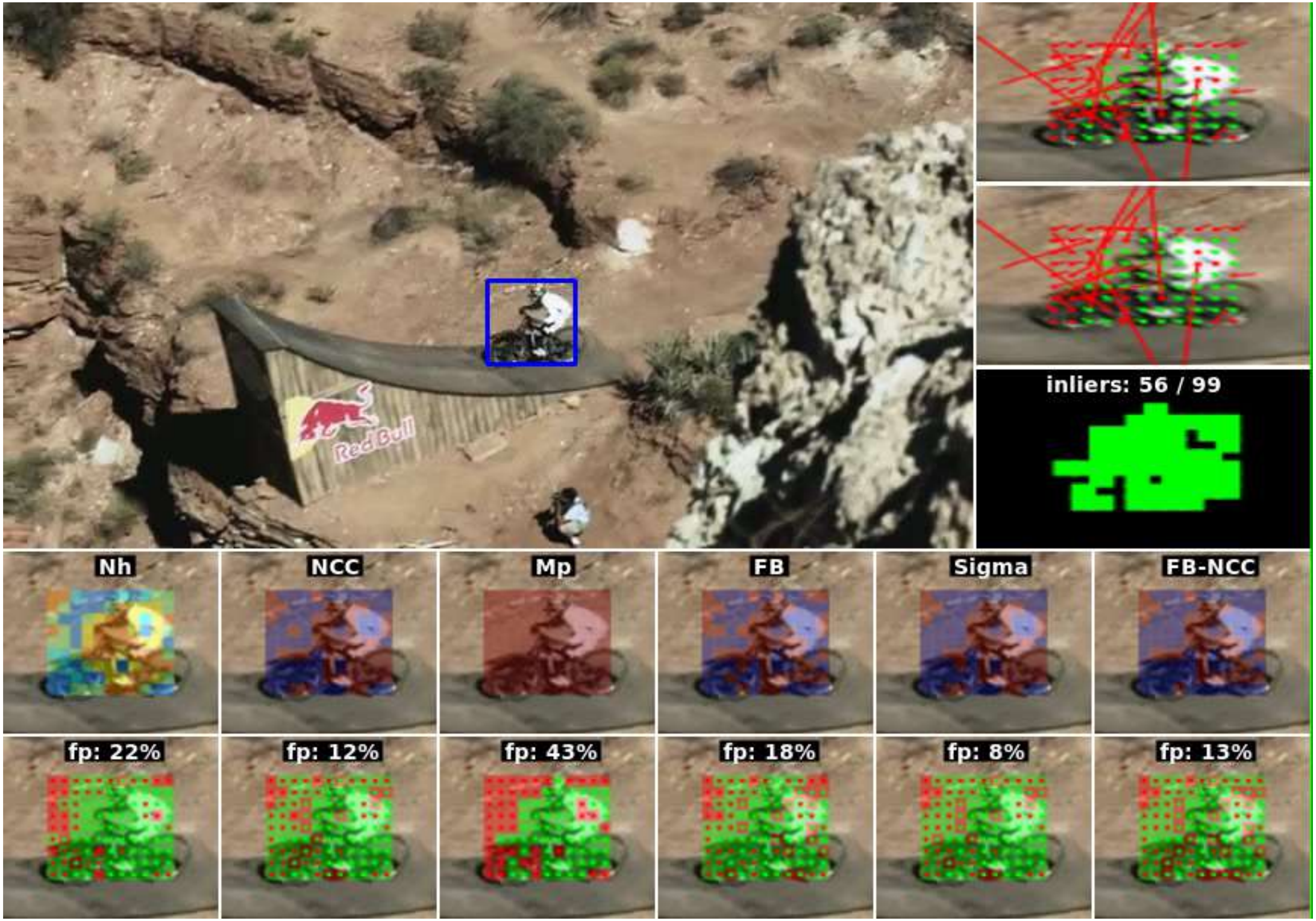


- Together form very a stronger predictor than the popular forward-backward

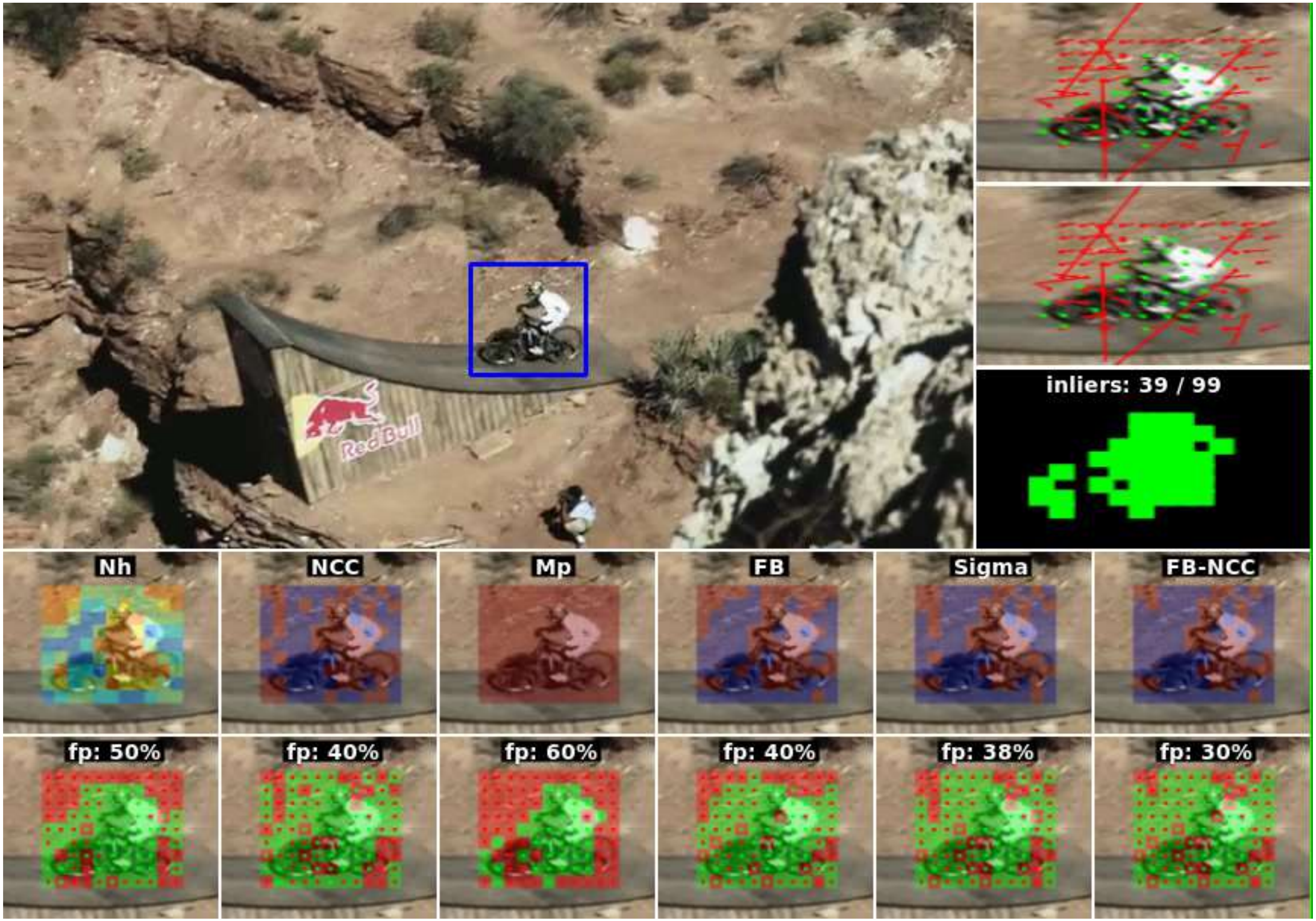
- Negligible computational cost (less than 10%)

T. Vojir and J. Matas. Robustifying the flock of trackers. CVWW '11,

FoT Error Prediction Bike tight box (ext. viewer)

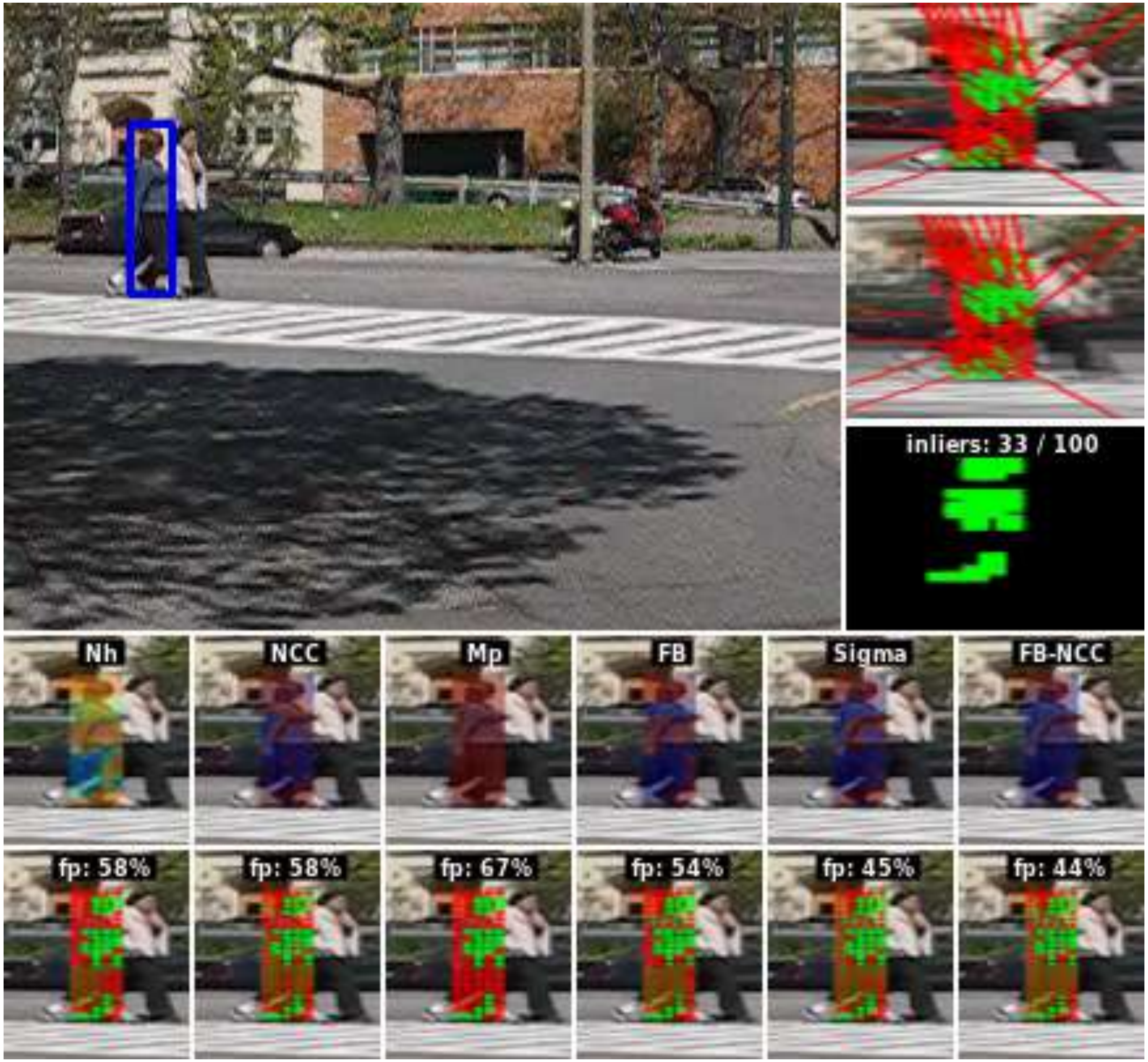


FoT Error Prediction Bike loose box (ext. viewer)



FoT Error Prediction

[(ext. viewer)]



The TLD (PN) Long-Term Tracker

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**

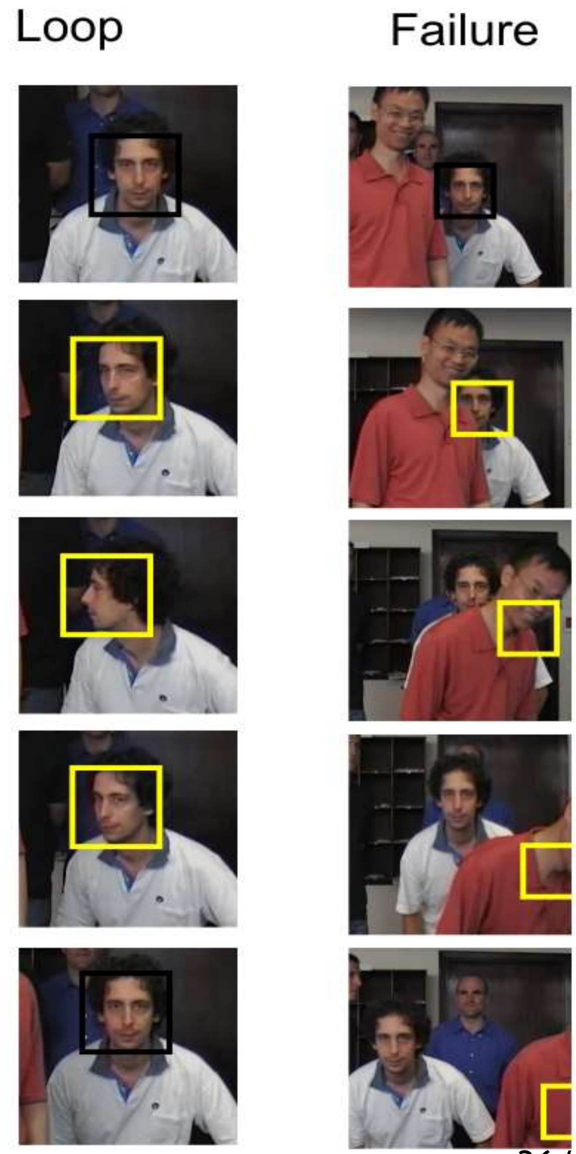
TLD a.k.a. PN Tracker a.k.a. “The Predator”

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

P-event: "Loop"

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

Tracker responses



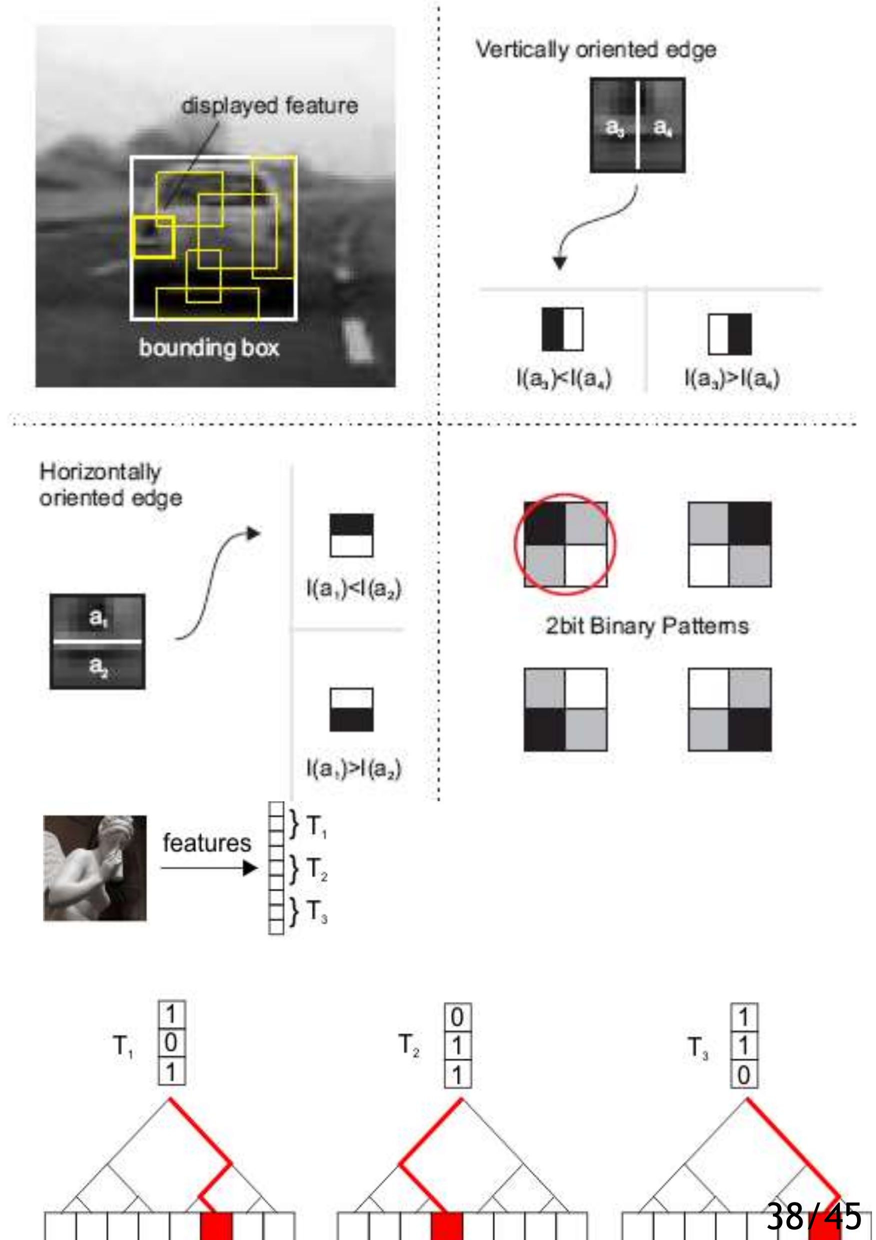
N-event: Uniqueness Enforcement

- 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 → prune from the model

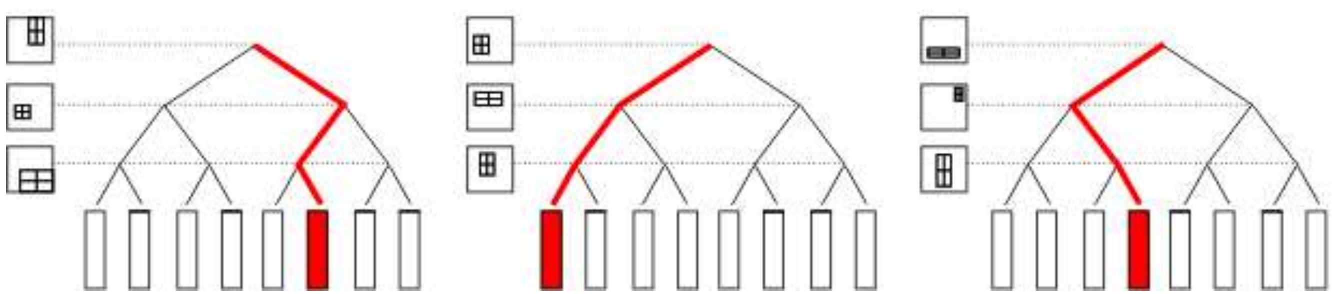
The Detector

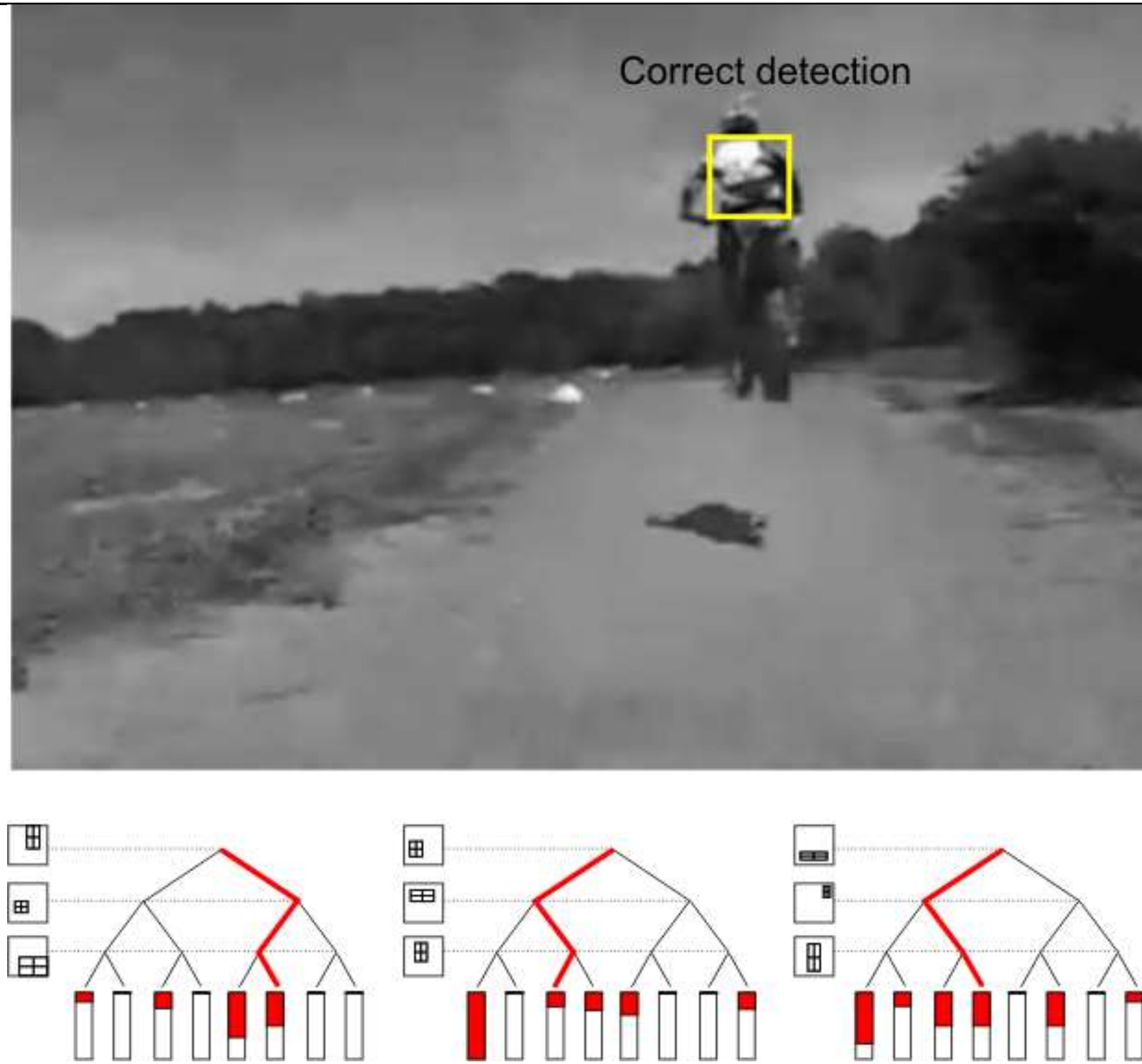
- Scanning window
- Randomized forest
- Trees implemented as ferns [Lepetit 2005]
- Real-time training/detection 20 fps on 320x240 image

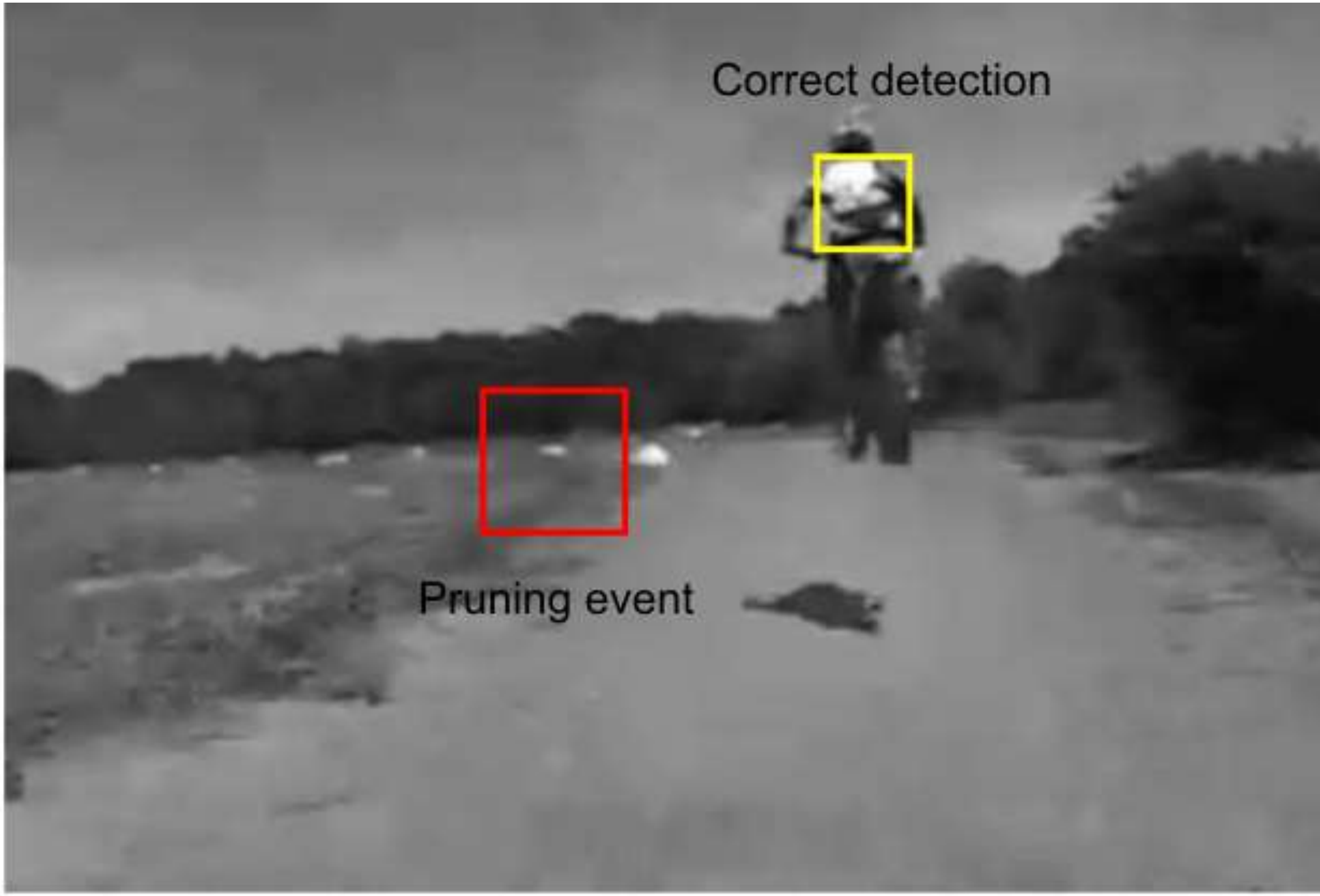
- High accuracy, 8 trees of depth 10
- 2bit Binary Patterns Combined Haar and LBP features
- Tree depth controls complexity & discriminability; currently not adaptive

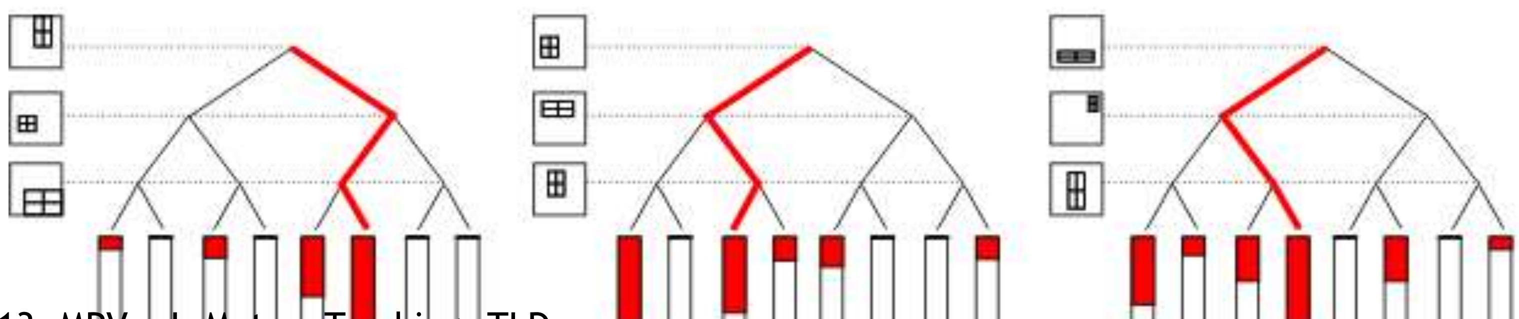



Growing event

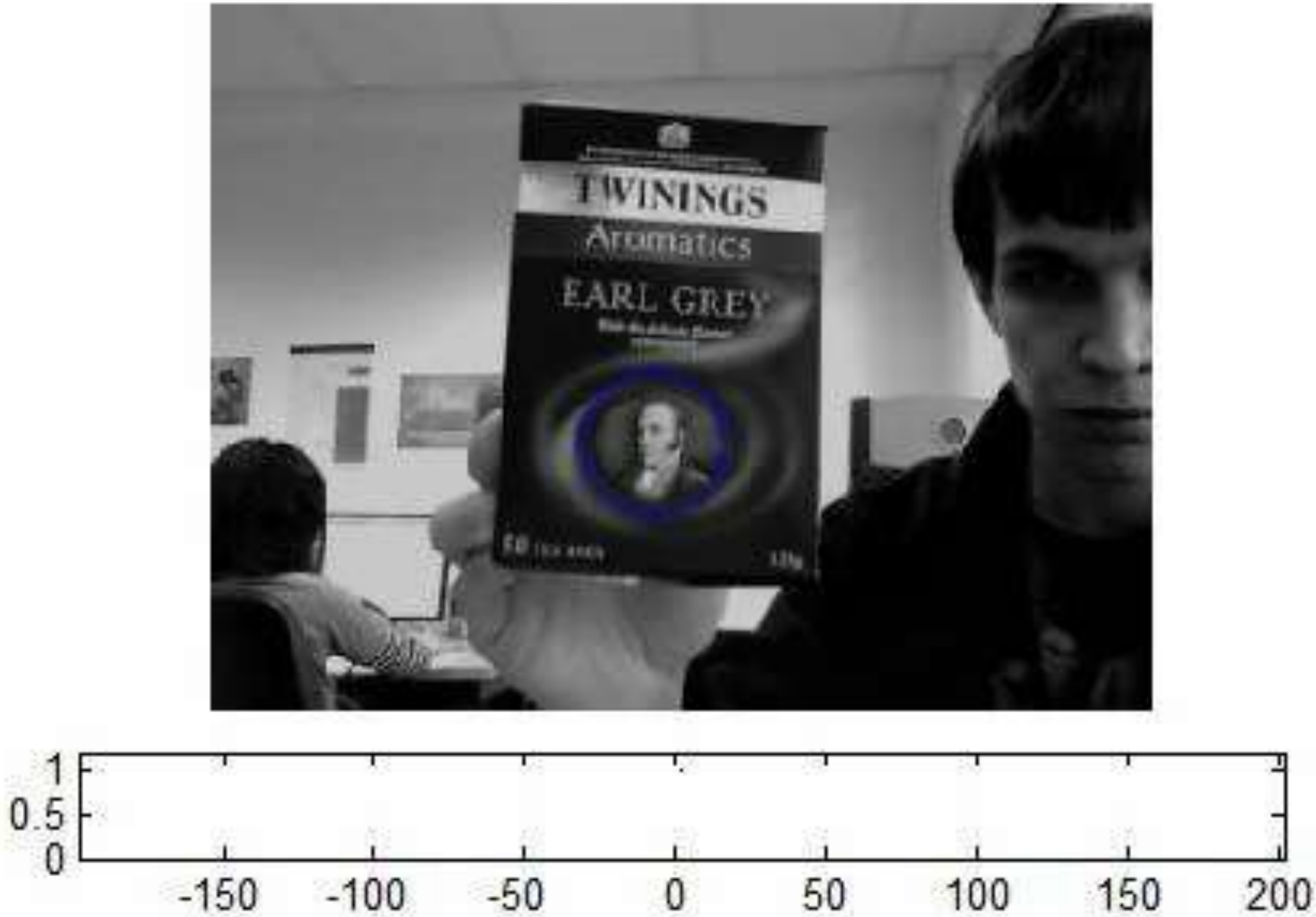


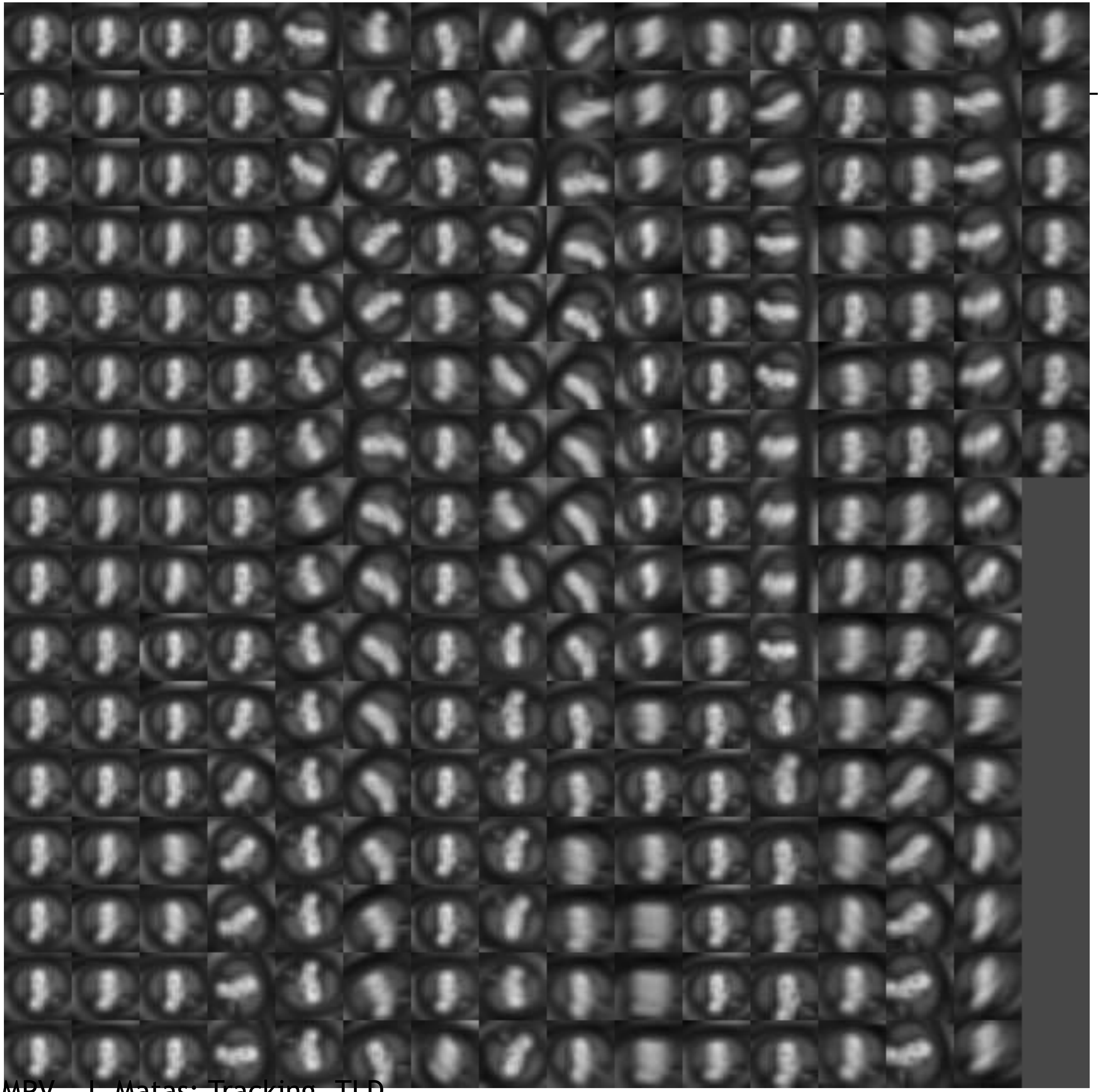

Correct detection


Correct detection
Pruning event



2





Summary

- "Visual Tracking" may refer to quite different problems:

- Robustness at all levels is the road to reliable performance
- Short-term tackers fail, sooner or later

- You cannot know for sure when making a mistake, but learn from contradictions!

- Long-term tracking includes learning and detection is interleaved and a detector learning plays a key role (might be even the output) is a promising direction.

THANK YOU.
Questions, please?