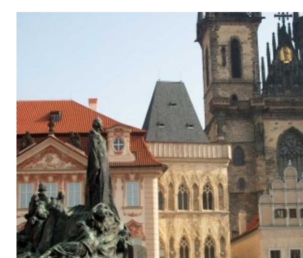
Visual Tracking

Jiri Matas

Center for Machine Perception
Department of Cybernetics,
Faculty of Electrical Engineering
Czech Technical University,

Prague, Czech Republic



Outline of the Lecture

1. Visual tracking: not one, but many problems.

2. The KLT tracker
3. The Mean-Shift tracker
4. The Flock of Trackers (FoT) -
   a robust short-term tracker example
5. The TLD tracker -
   a robust long-term tracker example
6. Tracking by detection (STRUCT), correlation (KCF)
7. How to evaluate a tracker?

8. Conclusions

Application domains of Visual Tracking

- monitoring, assistance, surveillance, control, defense
- robotics, autonomous car driving, rescue
- measurements: medicine, sport, biology, meteorology
- **human computer interaction**
- **augmented reality**
- **management of video content: indexing, search**
- **film production and postproduction: motion capture, editing**
- **action and activity recognition**
- image stabilization
- mobile applications
- camera “tracking”

Applications, applications, applications, …

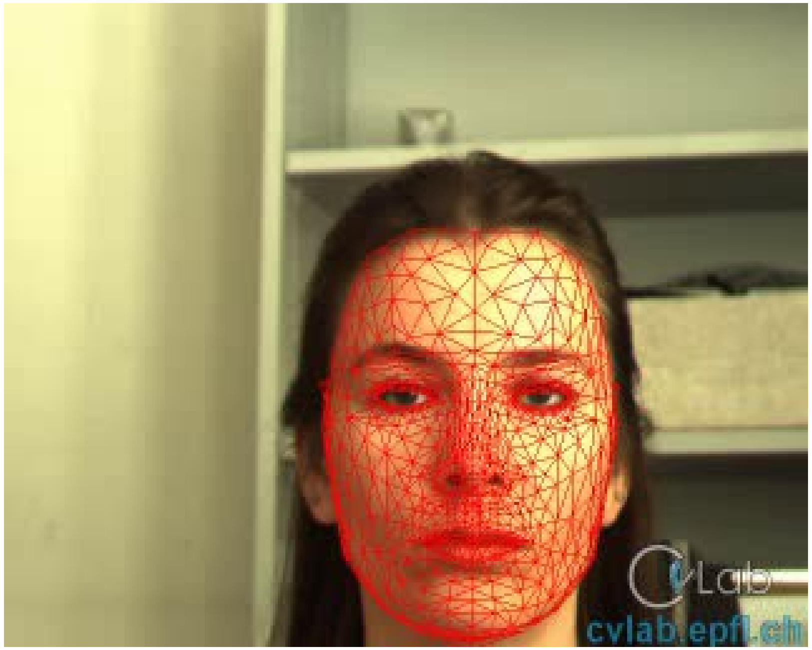
Tracking Applications ….

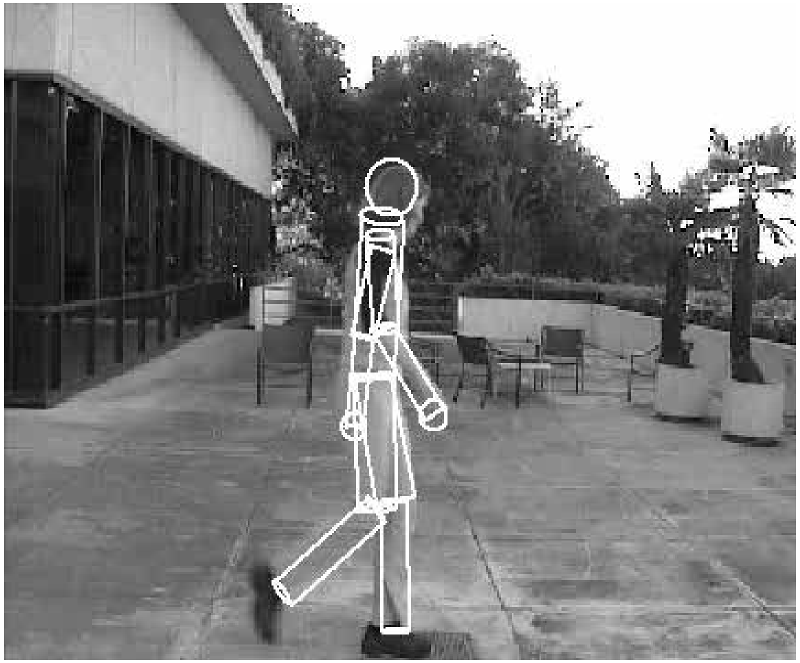
– Team sports: game analysis, player statistics, video annotation, …

Sport examples

http://cvlab.epfl.ch/~lepetit/
http://www.dartfish.com/en/media-gallery/videos/index.htm

Tracking people and faces





http://cvlab.epfl.ch/research/completed/realtime_tracking/

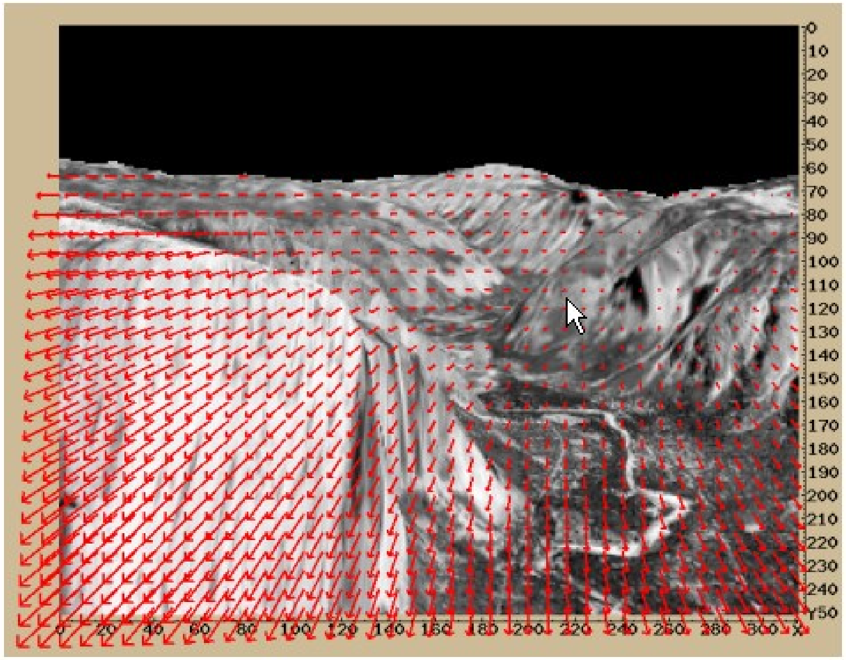
http://www.cs.brown.edu/~black/3Dtracking.html

Slide Credit: Patrick Perez

We know what tracking is?

video credit: Helmut Grabner

Tracking and Motion Estimation.

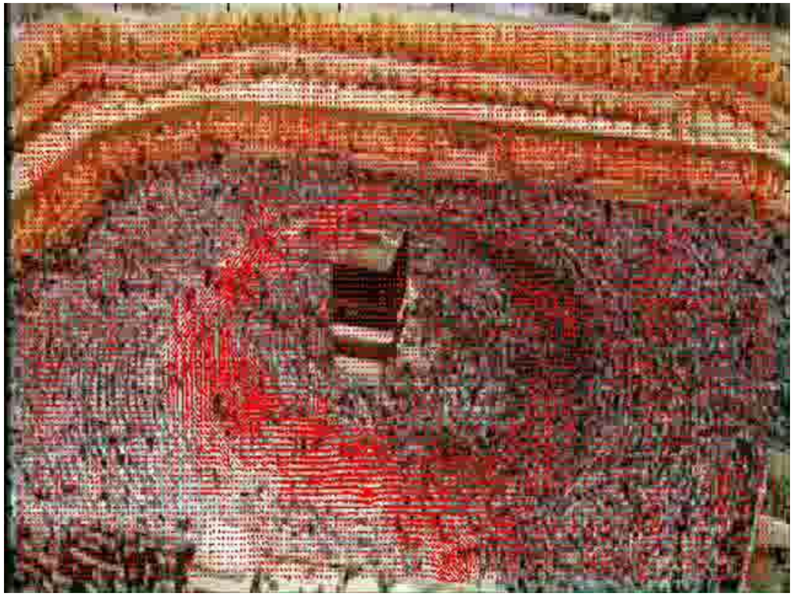


Yosemite sequence real flow

ie. if a perfect optic flow algorithm was available, tracking would be solved?

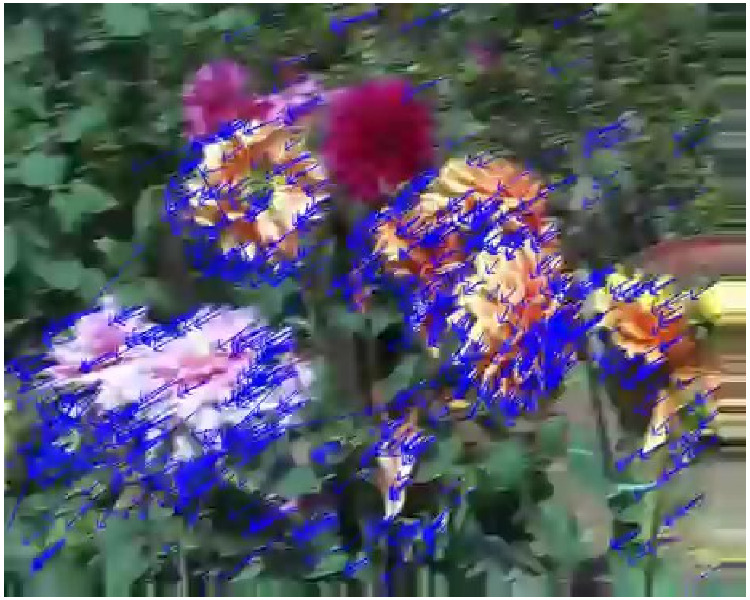
Motion field examples

Dense motion field



http://www.cs.cmu.edu/~saada/Projects/CrowdSegmentation/

Sparse motion field



http://www.youtube.com/watch?v=ckVQrwYIjAs

Optic Flow

Standard formulation:

- At every pixel, 2D displacement is estimated between consecutive frames

Missing:

- occlusion – disocclusion handling: pixels visible in one image only
  - in the standard formulation, “don’t know” is not an answer
- considering the 3D nature of the world

Practical issues hindering progress in optic flow:

- is the ground truth ever known?
  - learning and performance evaluation problematic (synthetic sequences ..)
- requires generic regularization (smoothing)
- failure (assumption validity) not easy to detect

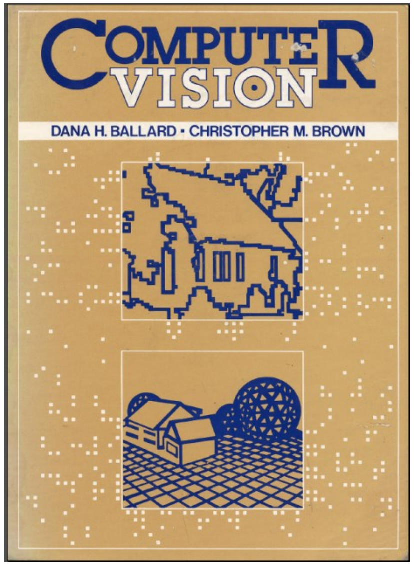
In certain applications, tracking is motion estimation on part of the image with specific constraints: augmented reality, sports analysis

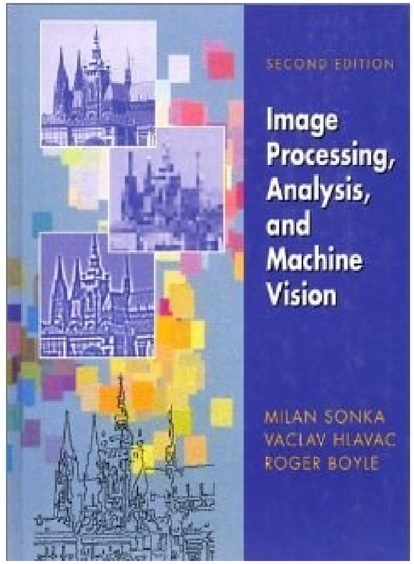
Definition (1): Tracking

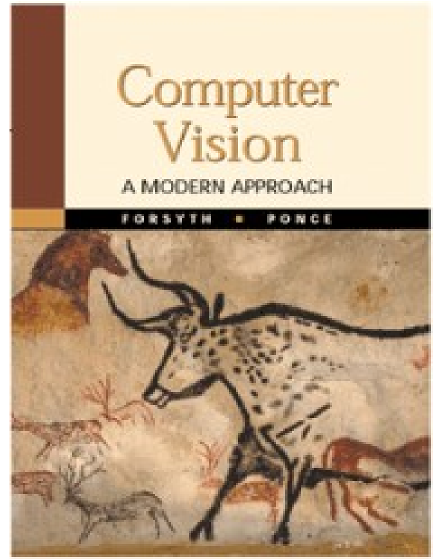
Establishing point-to-point correspondences

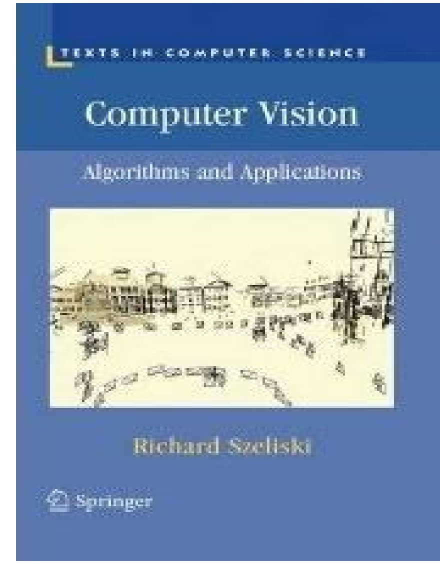
in image sequences

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 <u>inference</u> about the <u>motion</u> of an <u>object</u> given a <u>sequence of images</u>.*
> *Good solutions of this problem have a variety of applications…*"

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

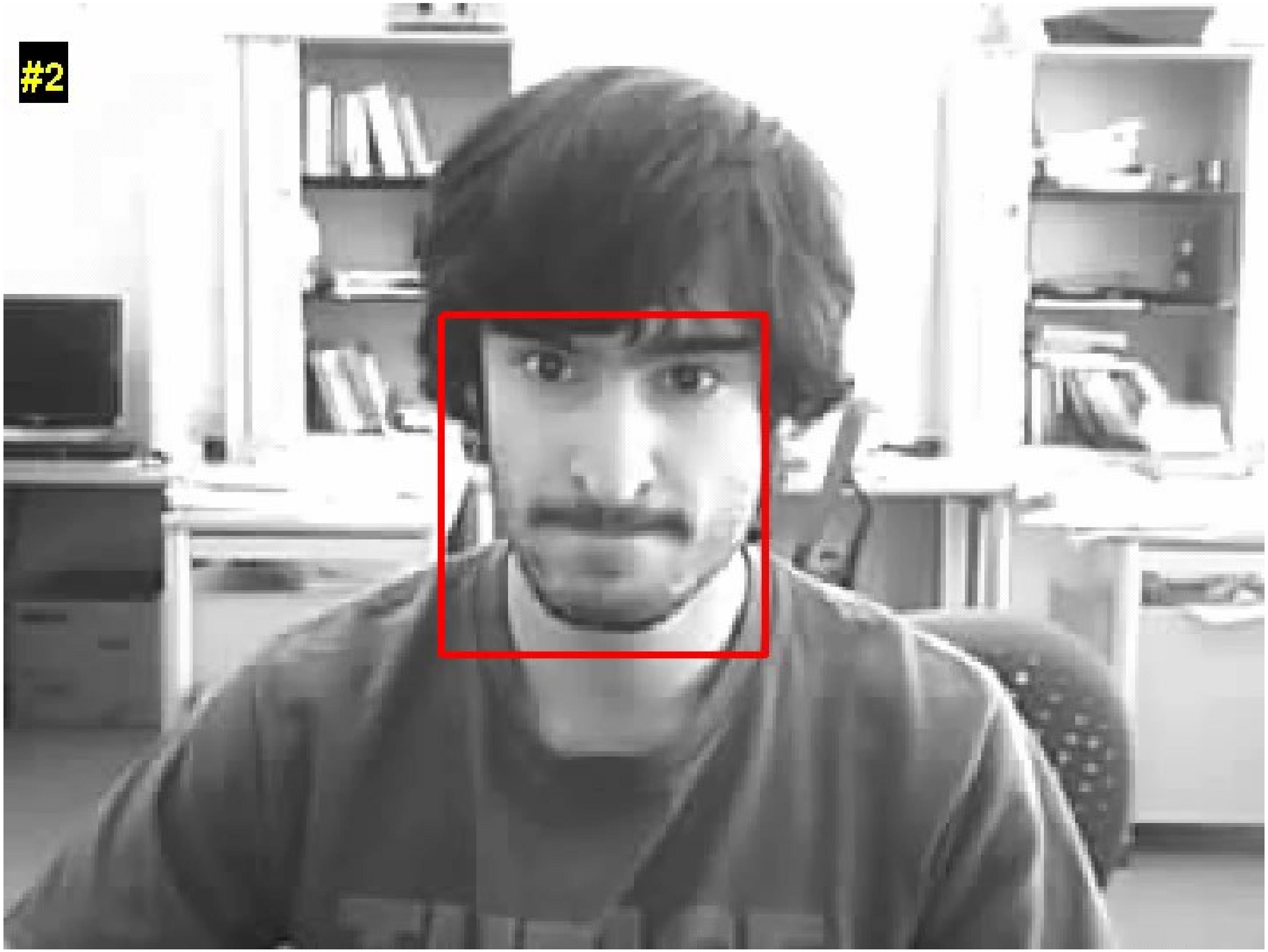
Establishing point-to-point correspondences

between all pairs frames in an image sequences

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

Tracking as detection and pose est. in 3D

A “standard” CV tracking method output



Definition (2): Tracking

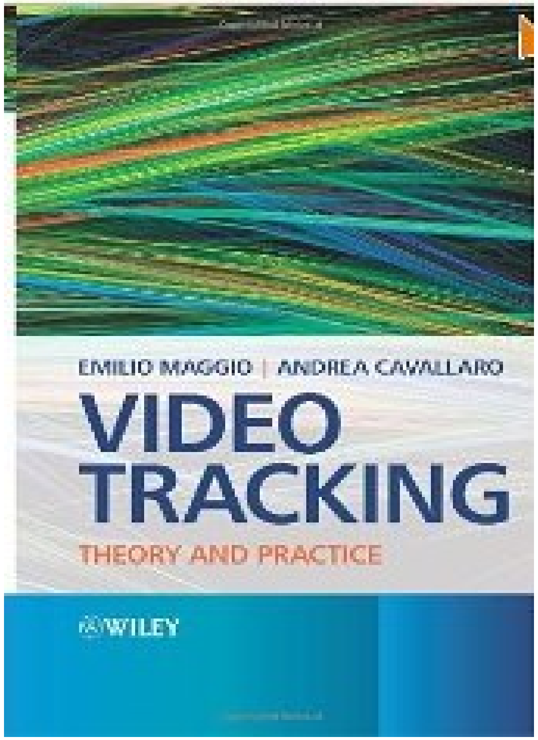
Given an initial estimate of its position,

locate X in a sequence of images,

Where X may mean:

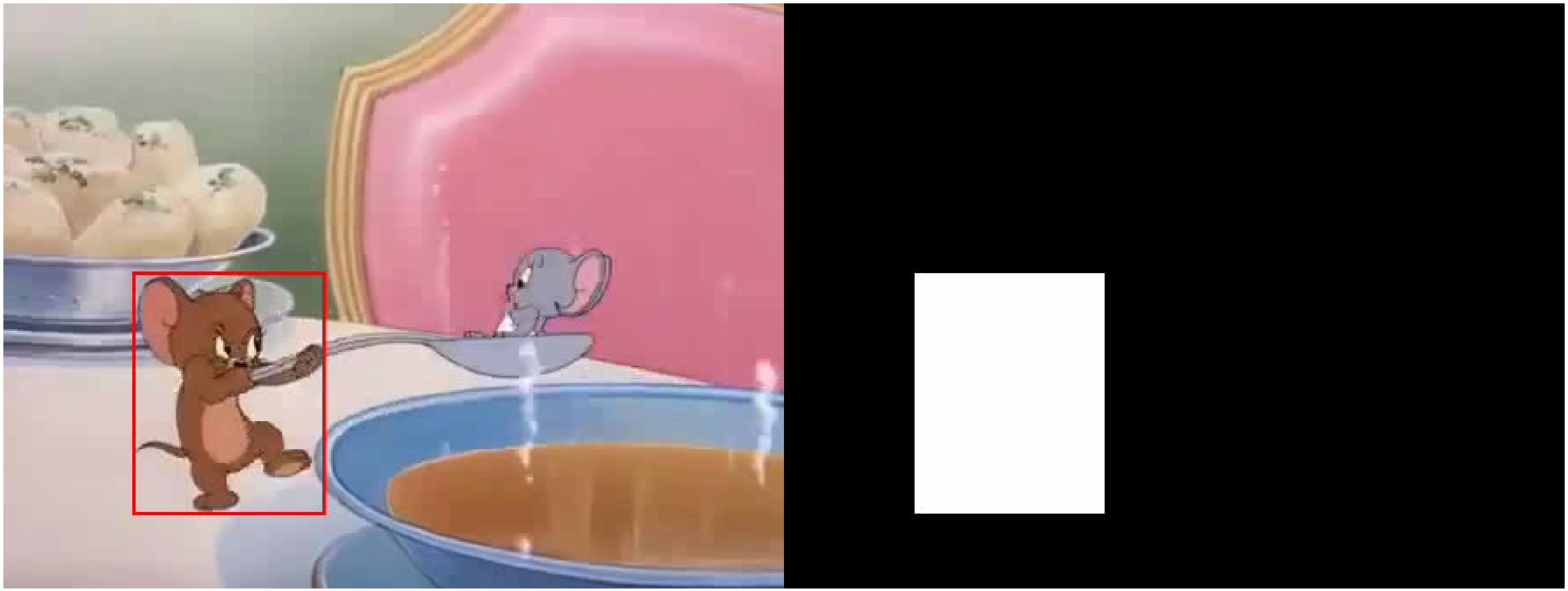
- A (rectangular) region
- An “interest point” and its neighbourhood
- An “object”

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

Definition (3): Tracking

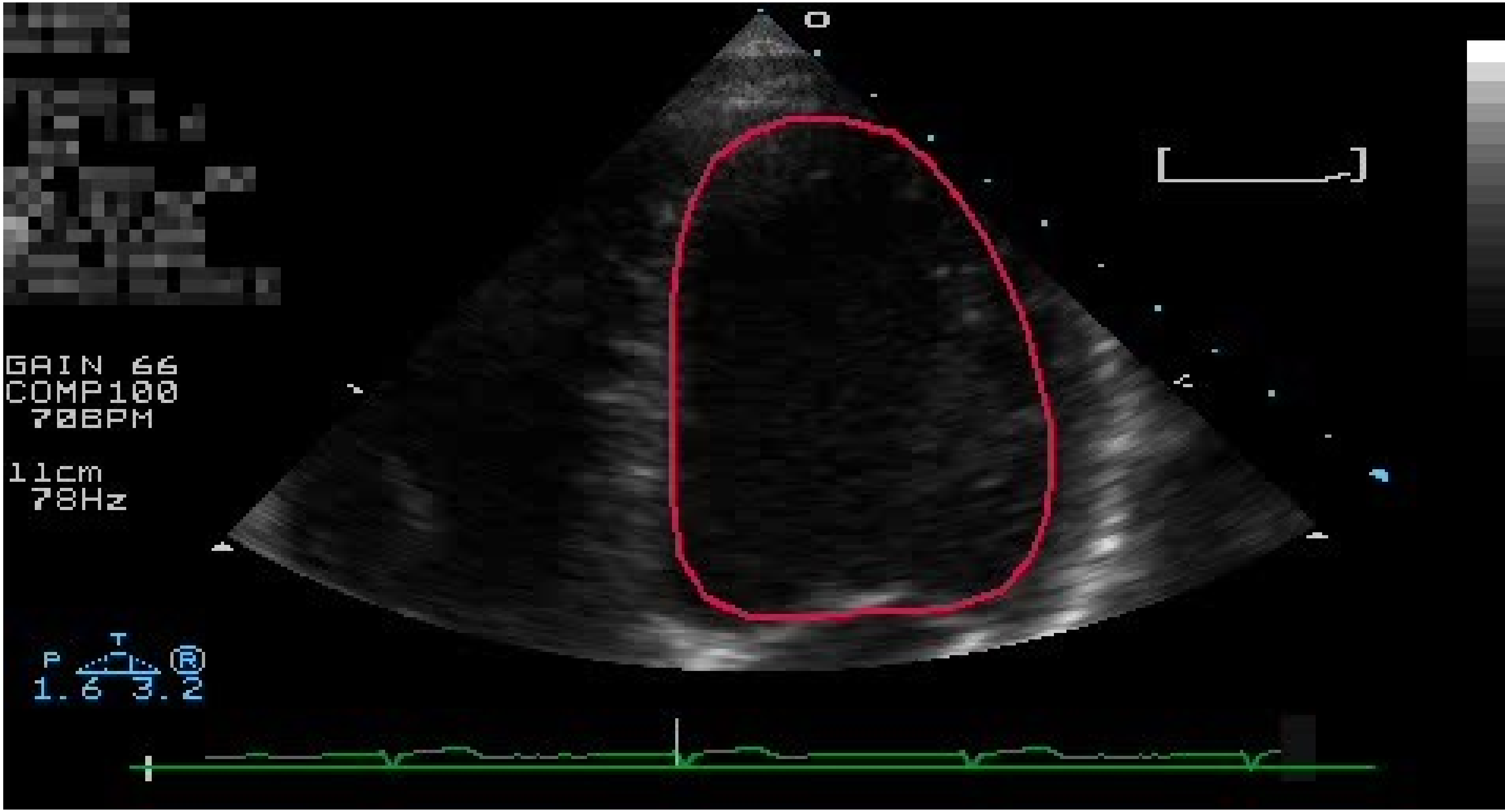
Given an initial estimate of the pose and state of X :

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

1. *estimate the pose and <u>state</u> 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 $t \leq T$

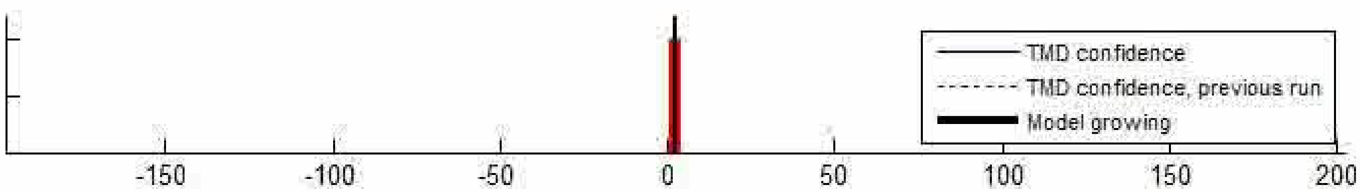
Tracking as segmentation



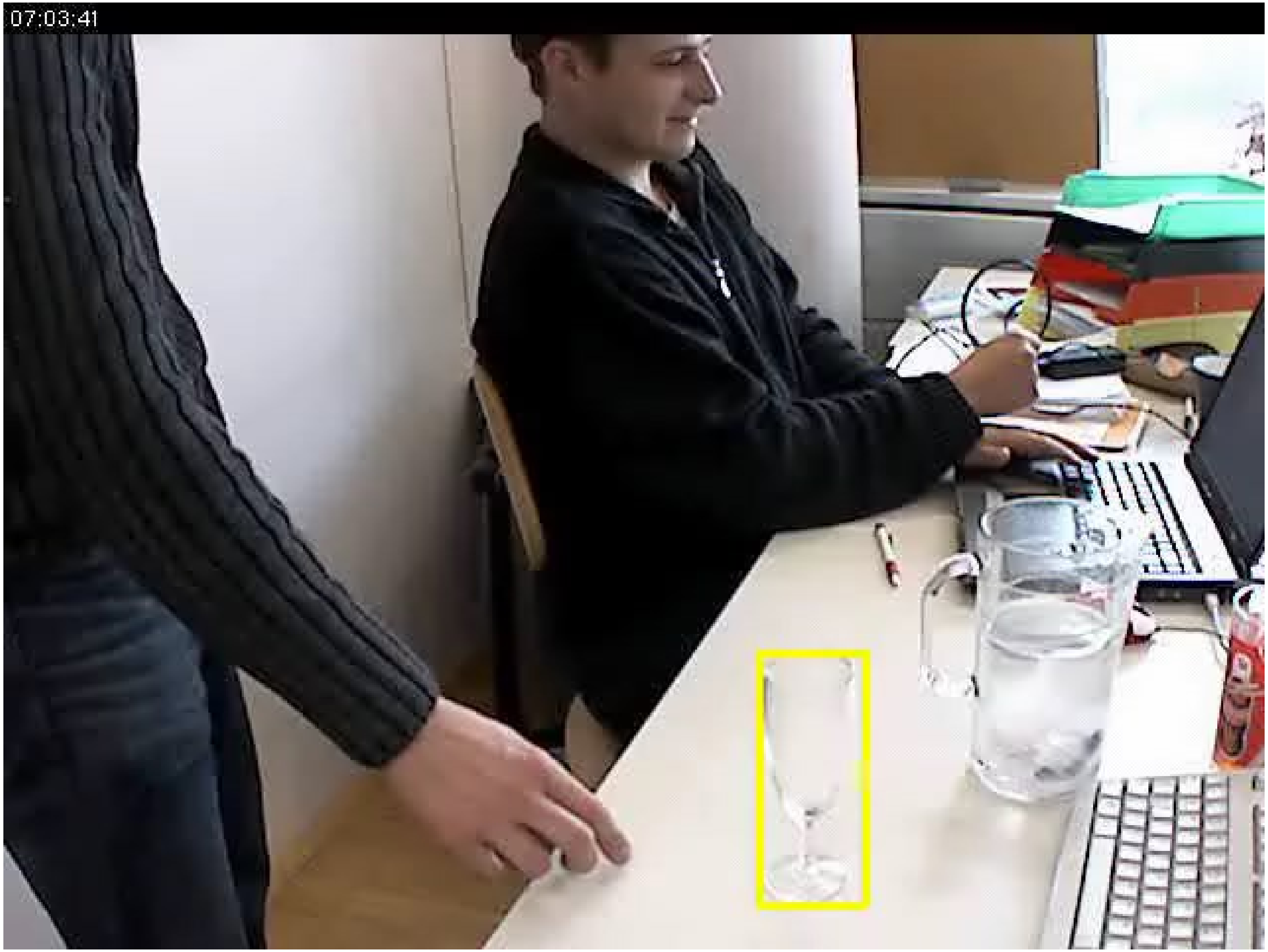
- heart

http://vision.ucsd.edu/~kbranson/research/cvpr2005.html

Tracking-Learning-Detection (TLD)



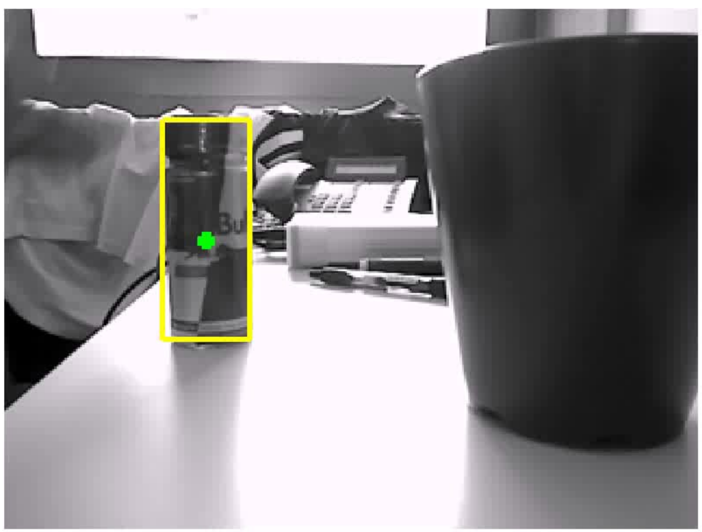
A "miracle": Tracking a Transparent Object



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

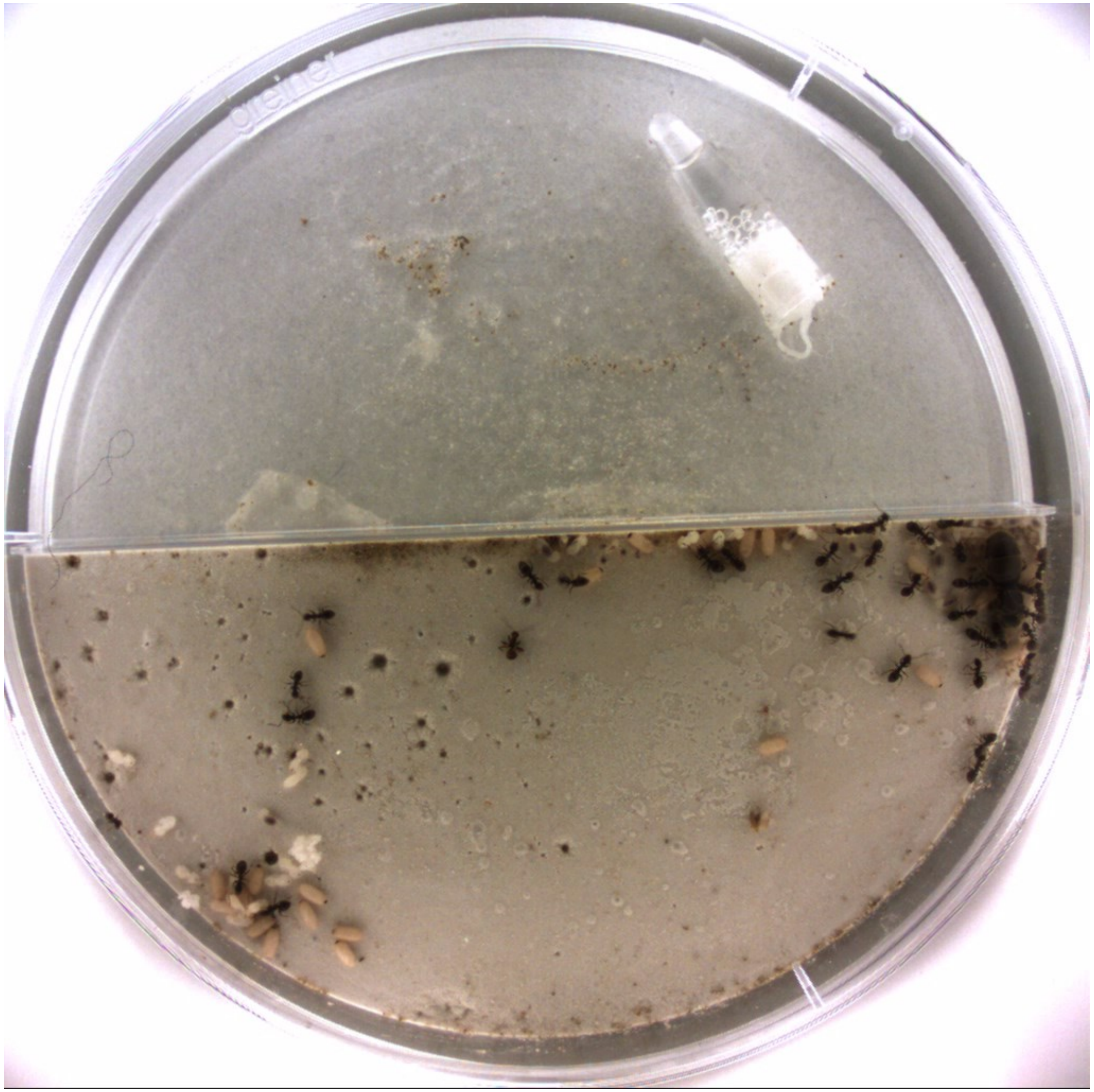
Other Tracking Problems:

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

...... multiple object tracking ....

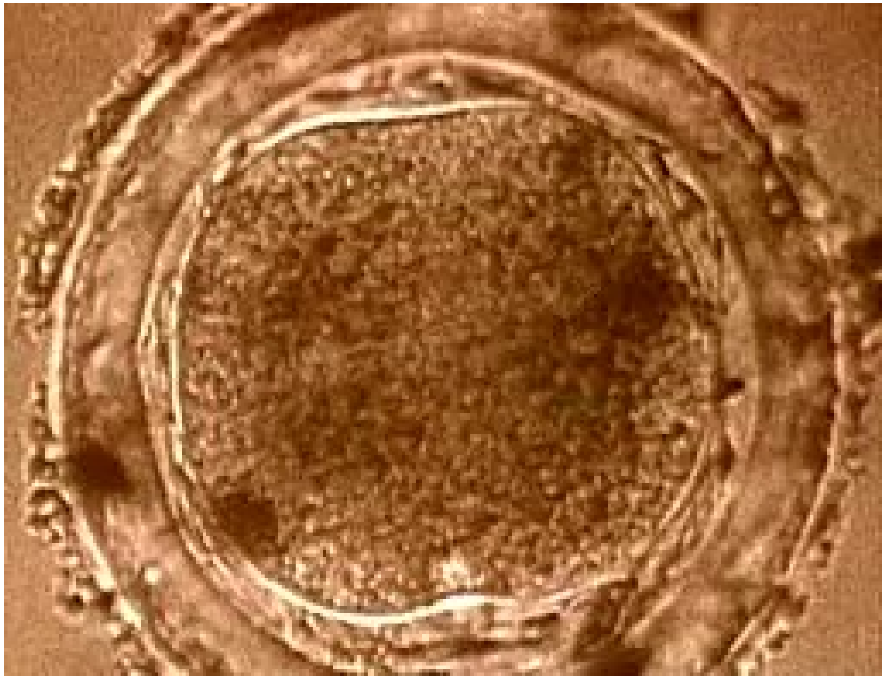
another example, example2

Tracking as detection and identification

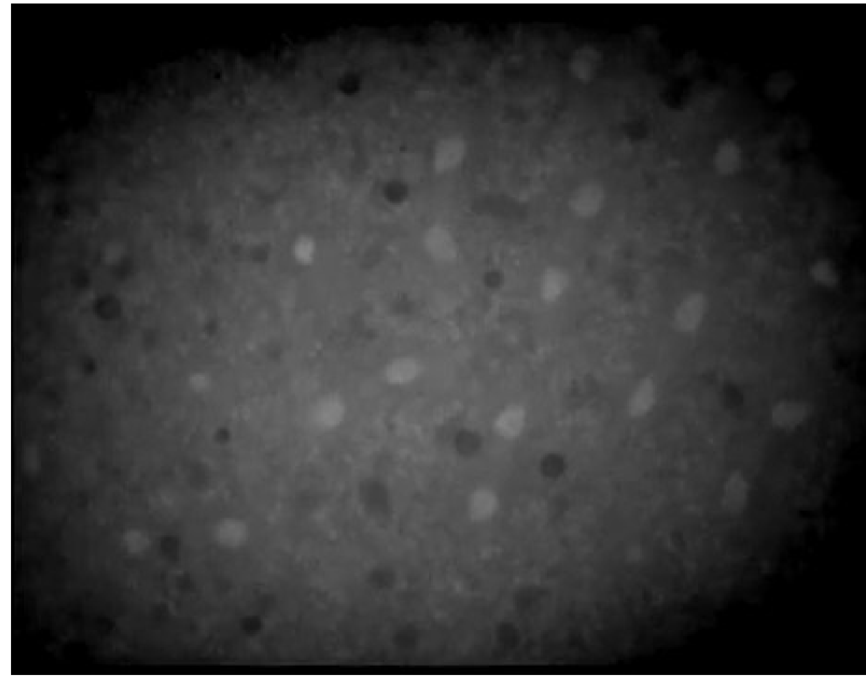


- ant tracking 1
- result 1

Other Tracking Problems:



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 ….

Motion Estimation from a Single Image

Other tracking problems:

- multiple cameras
- RGBD sensors
- combination of sensors (accelerometer + visual)
- ….

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.

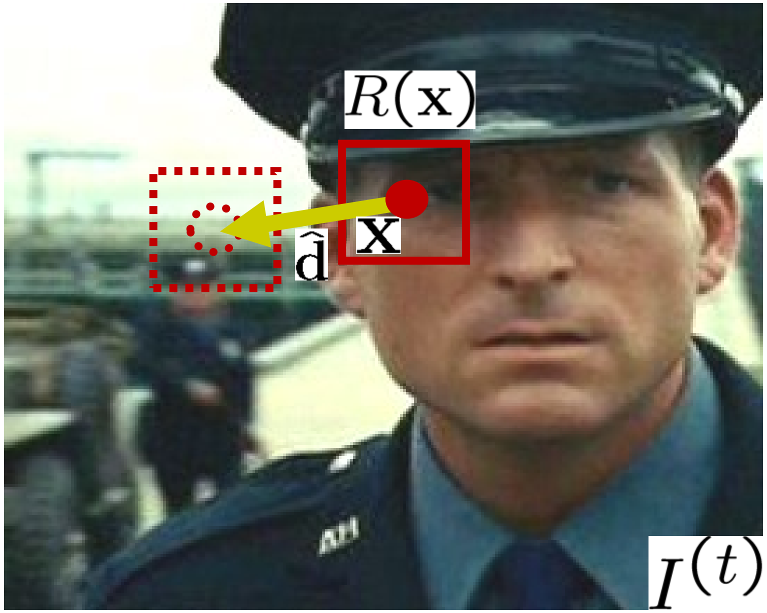
Optic Flow, Motion estimation: establish all correspondences a sequence

The KLT tracker

Fragment tracking

– Problem: tracking "key points" (SIFT, SURF, STAR, RIFF, FAST), or random image patches, as long as possible
  - Input: detected/chosen patches
  - Output: *tracklets* of various life-spans

slide credit: Patrick Perez

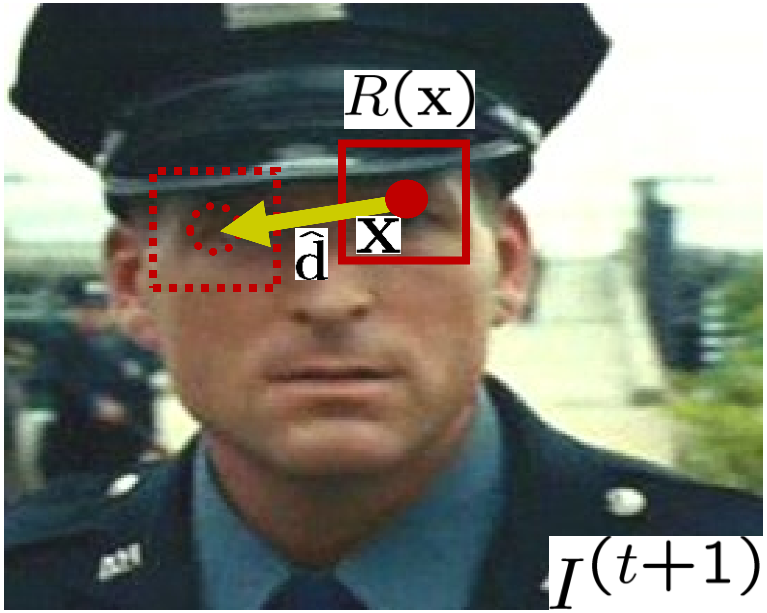


$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \underbrace{\sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2}_{\text{SSD}}$$

Fragment tracking

- Problem: tracking “key points” (SIFT, SURF, STAR, RIFF, FAST), or random image patches, as long as possible
  - Input: detected/chosen patches
  - Output: *tracklets* of various life-spans

slide credit: Patrick Perez



$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \underbrace{\sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2}_{\text{SSD}}$$

Good Points to Track. Lucas - Kanade (1981)

- for tracking, the sum of square differences (SSD) is an acceptable similarity measure, as illumination rarely changes between consecutive frames:

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \underbrace{\sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2}_{\text{SSD}}$$

- Displacements are small, use 1st-order Taylor expansion inside SSD:

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}) + \nabla I^{(t+1)}(\mathbf{p})^{\mathsf{T}}\mathbf{d} - I^{(t)}(\mathbf{p})|^2$$

$$\hat{\mathbf{d}} = -\underbrace{\left(\sum_{\mathbf{p}\in R(\mathbf{x})} \nabla I(\mathbf{p})\nabla I(\mathbf{p})^{\mathsf{T}}\right)}_{\mathbf{A}}^{-1} \sum_{\mathbf{p}\in R(\mathbf{x})} \nabla I(\mathbf{p}) I_t(\mathbf{p})$$

- For good conditioning, and for the Moravec condition to hold, the matrix **A** must have no eigenvalue ≈ 0

Multi-resolution Lucas-Kanade

– First assuming small displacement: 1st-order Taylor expansion inside SSD

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}) + \nabla I^{(t+1)}(\mathbf{p})^{\mathsf{T}}\mathbf{d} - I^{(t)}(\mathbf{p})|^2$$

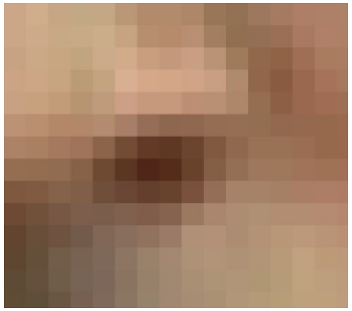
slide credit: Patrick Perez

$$\hat{\mathbf{d}} = -\left(\sum_{\mathbf{p}\in R(\mathbf{x})} \nabla I(\mathbf{p})\nabla I(\mathbf{p})^{\mathsf{T}}\right)^{-1} \sum_{\mathbf{p}\in R(\mathbf{x})} \nabla I(\mathbf{p}) I_t(\mathbf{p})$$

For good conditioning, patch must be textured/structured enough:

- Uniform patch: no information
- Contour element: aperture problem (one dimensional information)
- Corners, blobs and texture: best estimate



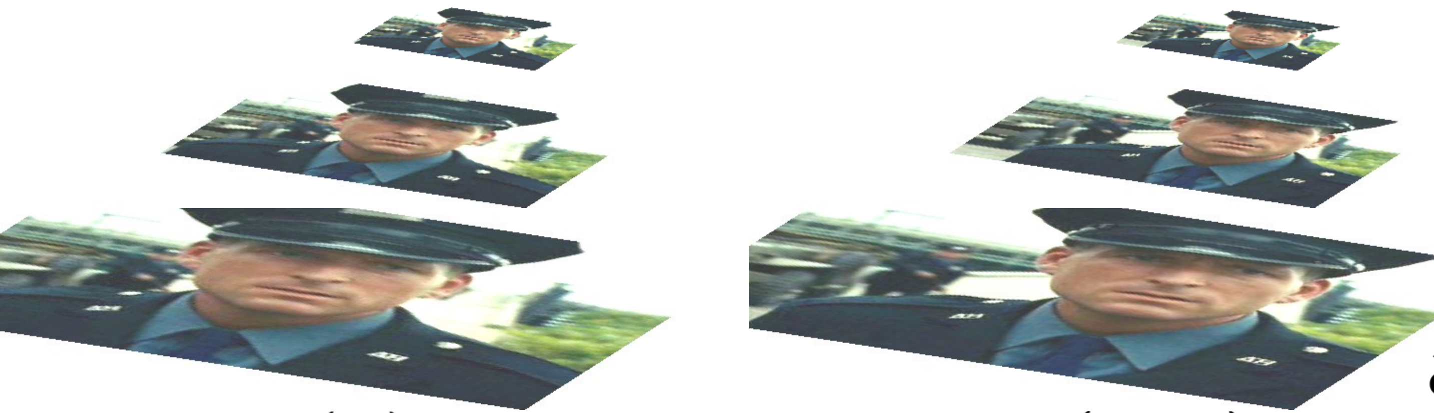



[Lucas and Kanda 1981][Tomasi and Shi, CVPR'94]

Multi-resolution Lucas-Kanade

- Arbitrary displacement
  - Multi-resolution approach: Gauss-Newton like approximation down image pyramid

slide credit:
Patrick Perez



$\{I^{(\ell,t)}\}$ $\{I^{(\ell,t+1)}\}$

$$\hat{\mathbf{d}}^{(\ell)}$$

$$\hat{\mathbf{d}}^{(\ell+1)} \overset{?}{\leftarrow} \hat{\mathbf{v}} + 2\hat{\mathbf{d}}^{(\ell)}$$

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \sum_{\mathbf{p}\in R^{\ell}(\mathbf{x})} |I^{(\ell,t+1)}(\mathbf{p}+2\mathbf{d}^{(\ell)}) + \nabla I^{(\ell,t+1)}(\mathbf{p}+2\mathbf{d}^{(\ell)})^{\mathsf{T}}\mathbf{v} - I^{(\ell,t)}(\mathbf{p})|^2$$

$$\hat{\mathbf{v}} = -\left(\sum_{\mathbf{p}\in R^{\ell}(\mathbf{x})} \nabla \tilde{I}^{(\ell,t)}(\mathbf{p}) \nabla \tilde{I}^{(\ell,t+1)}(\mathbf{p})^{\mathsf{T}}\right)^{-1} \sum_{\mathbf{p}\in R(\mathbf{x})} \nabla \tilde{I}^{(\ell,t+1)}(\mathbf{p}) \tilde{I}_t^{(\ell,t+1)}(\mathbf{p})$$

Monitoring quality

– Translation is usually sufficient for small fragments, but:
  - Perspective transforms and occlusions cause drift and loss
– Two complementary options
  - Kill tracklets when minimum SSD too large
  - Compare as well with *initial patch under affine transform (warp)* assumption

slide credit:
Patrick Perez

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \sum_{\mathbf{p}\in R_t} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2$$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{\mathbf{p}\in R_0} |I^{(t+1)}(\mathbf{w}[\mathbf{p}]) - I^{(0)}(\mathbf{p})|^2$$

Good Points to Track. The history.

H. Moravec (1980) observed:

to be able to track a (region around) a point, the region must (at least) be different from all regions in its neighbouhood, i. e.

good points to track must have low self-similarity everywhere in their neighbourhood

H. Moravec, Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. *Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute*.

The Mean-shift Tracker
(colour-based tracking)

Color-based tracking

- Global description of tracked region: color histogram
- Reference histogram with *B* bins

  $$\mathbf{q}^* = (q_u^*)_{u=1\cdots B}$$

  set at track initialization
- Candidate histogram at current instant

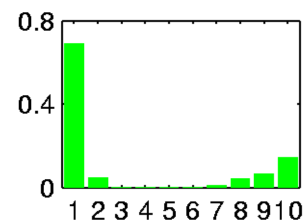
  $$\mathbf{q}(\mathbf{x}) = (q_u(\mathbf{x}))_{u=1\cdots B}$$

  gathered in region $R(\mathbf{x})$ of current image.
- At each instant

  $$\widehat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}} \operatorname{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))$$

  - searched around $\widehat{\mathbf{x}}_t$
  - iterative search initialized with $\widehat{\mathbf{x}}_t$: meanshift-like iteration

slide credit: Patrick Perez



Color-based tracking

slide credit:
Patrick Perez

- Global description of tracked region: color histogram
- Reference histogram with *B* bins

$$\mathbf{q}^* = (q_u^*)_{u=1\cdots B}$$

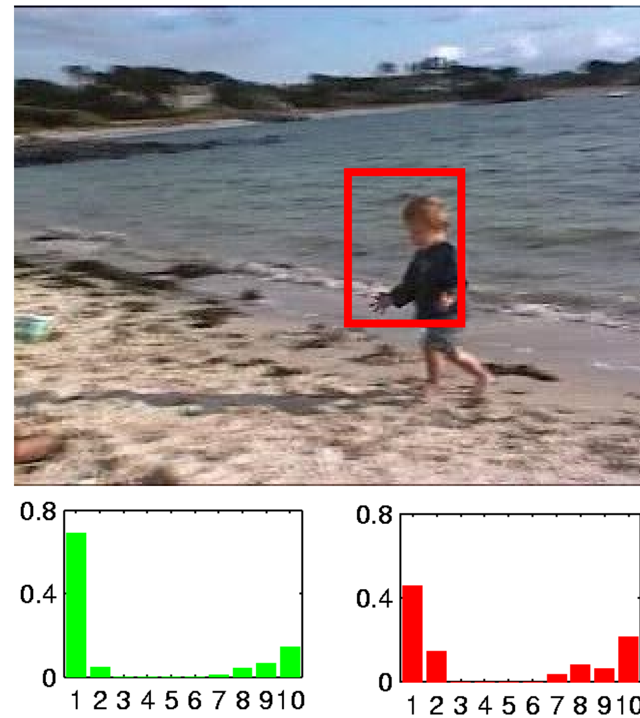
  set at track initialization
- Candidate histogram at current instant

$$\mathbf{q}(\mathbf{x}) = (q_u(\mathbf{x}))_{u=1\cdots B}$$

  gathered in region $R(\mathbf{x})$ of current image.
- At each instant

$$\widehat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}} \mathrm{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))$$

  - searched around $\widehat{\mathbf{x}}_t$
  - iterative search initialized with $\widehat{\mathbf{x}}_t$: meanshift-like iteration



Color-based tracking

- Global description of tracked region: color histogram
- Reference histogram with *B* bins

$$\mathbf{q}^* = (q_u^*)_{u=1\cdots B}$$

  set at track initialization

- Candidate histogram at current instant

$$\mathbf{q}(\mathbf{x}) = (q_u(\mathbf{x}))_{u=1\cdots B}$$

  gathered in region $R(\mathbf{x})$ of current image.

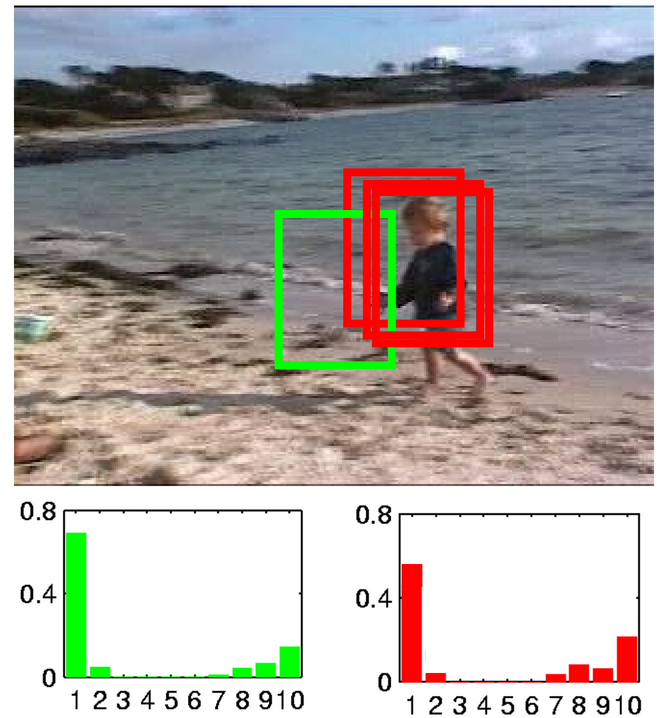
- At each instant

$$\hat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}} \mathrm{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))$$

  - searched around $\hat{\mathbf{x}}_t$
  - iterative search initialized with $\hat{\mathbf{x}}_t$: meanshift-like iteration

$\mathbf{I}_t$

slide credit: Patrick Perez



Color distributions and similarity

– Color histogram weighted by a kernel
  - Kernel elliptic support sits on the object
  - Central pixels contribute more
  - Makes differentiation possible

$$q_u(\mathbf{x}) \propto \sum_{\mathbf{p}_i \in R(\mathbf{x})} k\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right) \mathbf{1}[I(\mathbf{p}_i) \in b_u]$$

  - *H*: "bandwidth" sym. def. pos. matrix, related to bounding box dimensions
  - *k*: "profile" of kernel (Gaussian or Epanechnikov)

– Histogram dissimilarity measure
  - Battacharyya measure $\text{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))^2 = 1 - \sum_u \sqrt{q_u^* q_u(\mathbf{x})} = 1 - \rho[\mathbf{q}^*, \mathbf{q}(\mathbf{x})]$
  - Symmetric, bounded, null only for equality
  - 1 - dot product on positive quadrant of unitary hyper-sphere

slide credit:
Patrick Perez

Color distributions and similarity

- Color histogram weighted by a kernel
  - Kernel elliptic support sits on the object
  - Central pixels contribute more
  - Makes differentiation possible

$$q_u(\mathbf{x}) \propto \sum_{\mathbf{p}_i \in R(\mathbf{x})} k\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right) \mathbf{1}[I(\mathbf{p}_i) \in b_u]$$

  - *H*: “bandwidth” sym. def. pos. matrix, related to bounding box dimensions
  - *k*: “profile” of kernel (Gaussian or Epanechnikov)
- Histogram dissimilarity measure
  - Battacharyya measure $\text{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))^2 = 1 - \sum_u \sqrt{q^*_u q_u(\mathbf{x})} = 1 - \rho[\mathbf{q}^*, \mathbf{q}(\mathbf{x})]$
  - Symmetric, bounded, null only for equality
  - 1 - dot product on positive quadrant of unitary hyper-sphere

slide credit: Patrick Perez

Iterative ascent

$$\hat{\mathbf{x}}_{t+1} = \arg\max_{\mathbf{x}} \sum_u \sqrt{q_u^* q_u(\mathbf{x})}$$

$$q_u(\mathbf{x}) \propto \sum_{\mathbf{p}_i} k\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right) \mathbf{1}[I(\mathbf{p}_i) \in b_u]$$

– Non quadratic minimization: iterative ascent with linearizations

u_i bin index of pixel i: $I(\mathbf{p}_i) \in b_{u_i}$

$$\nabla \sum_u \sqrt{q_u^* q_u(\mathbf{x})} \propto H^{-1} \sum_{\mathbf{p}_i} \sqrt{\frac{q_{u_i}^*}{q_{u_i}(\mathbf{x})}} k'\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right)(\mathbf{x} - \mathbf{p}_i)$$

– Setting move to (*g*=-*h*')

$$\frac{\sum_{\mathbf{p}_i} \sqrt{\frac{q_{u_i}^*}{q_{u_i}(\mathbf{x})}} g\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right)(\mathbf{p}_i - \mathbf{x})}{\sum_{\mathbf{p}_i} \sqrt{\frac{q_{u_i}^*}{q_{u_i}(\mathbf{x})}} g\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right)} = \mathsf{MeanShift}(\mathbf{x}) - \mathbf{x}$$

yields a simple algorithm…

Meanshift tracker

- In frame t+1
  - Start search at $\mathbf{y}^{(0)} = \hat{\mathbf{x}}_t$
  - Until stop
    - Compute candidate histogram $\mathbf{q}(\mathbf{y}^{(n)})$
    - Weight pixels inside kernel support

$$\forall \mathbf{p}_i \in R(\mathbf{y}^{(n)}),\ w_i \propto \sqrt{\frac{q^*_{u_i}}{q_{u_i}(\mathbf{y}^{(n)})}} g\left(\|\mathbf{p}_i - \mathbf{y}^{(n)}\|^2_{H^{-1}}\right),\ \sum_i w_i = 1$$

    - Move kernel

$$\mathbf{y}^{(n+1)} = \mathbf{y}^{(n)} + [\text{MeanShift}(\mathbf{y}^{(n)}) - \mathbf{y}^{(n)}] = \sum_{\mathbf{p}_i \in R(\mathbf{y}^{(n)})} w_i \mathbf{p}_i$$

    - Check overshooting until $\rho[\mathbf{q}^*, \mathbf{p}(\mathbf{y}^{(n+1)})] < \rho[\mathbf{q}^*, \mathbf{p}(\mathbf{y}^{(n)})],\ \mathbf{y}^{(n+1)} \leftarrow \frac{\mathbf{y}^{(n)} + \mathbf{y}^{(n+1)}}{2}$
    - If $\|\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}\|^2 < \varepsilon$ stop, else $n \leftarrow n + 1$
  - $\hat{\mathbf{x}}_{t+1} = \mathbf{y}^{(n+1)}$

slide credit: Patrick Perez

Mean Shift tracking example

Feature space: 16×16×16 quantized RGB

Target: manually selected on 1st frame

Average mean-shift iterations: 4

Mean Shift tracking example

D. Comaniciu, V. Ramesh, P. Meer: *Kernel-Based Object Tracking* TPAMI, 2003

Pros and cons

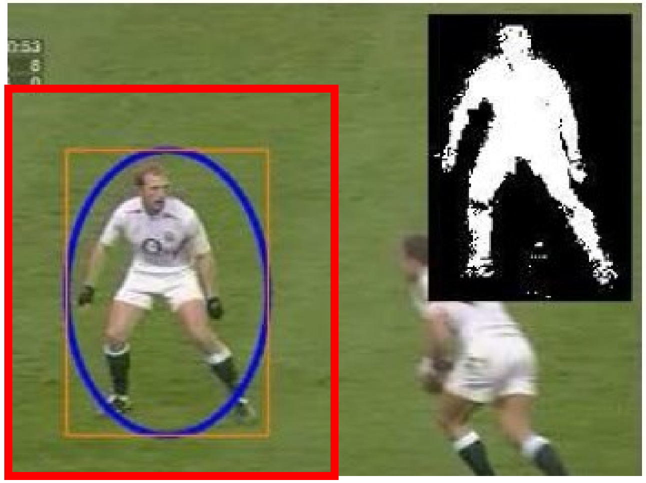
- Low computational cost (easily real-time)
- Surprisingly robust
  - Invariant to pose and viewpoint
  - Often no need to update reference color model

- Invariance comes at a price
  - Position estimate prone to fluctuation
  - Scale and orientation not well captured
  - Sensitive to color clutter (e.g., teamates in team sports)
- Deterministic local search challenged by
  - abrupt moves
  - occlusions

slide credit:
Patrick Perez

Variants

- Remove background corruption in reference
  - Simple segmentation based on surrounding color at initialization
  - Re-estimation of foreground model
  - Amounts to zero bins for colors more frequent in surrounding than in selection



- Scale/orientation estimation
  - Originally: greedy search around current scale/orientation
  - Afterwards: incorporate loose spatial layout (via multiple spatial kernels or spatial partionning with sub-models)
- Robustness to camera movement
  - Robust estimation of dominant apparent motion
  - Start search at previous position displaced according to dominant motion

slide credit:
Patrick Perez

Variants (2)

- Improved discrimination for improved robustness
  - Selection of best color space or color combinations to distinguish foreground from background
  - Alternative or complementary features (intensity, textures, co-occurrences)
- Improved accuracy
  - Coupling with more precise though more fragile tracking (fragment-based in particular)
- Smoother similarity measure
  - Kullback-Leibler
- Alternative search algorithms
  - Trust region

slide credit:
Patrick Perez

Further variants

- On-the-fly adaptation
  - Radical pose or illumination changes require adaptation
  - Usually linear mixing with exponential forgetting

$$\mathbf{q}^*_{t+1} = (1-\alpha)\mathbf{q}^*_t + \alpha\mathbf{q}(\widehat{\mathbf{x}}_t)$$

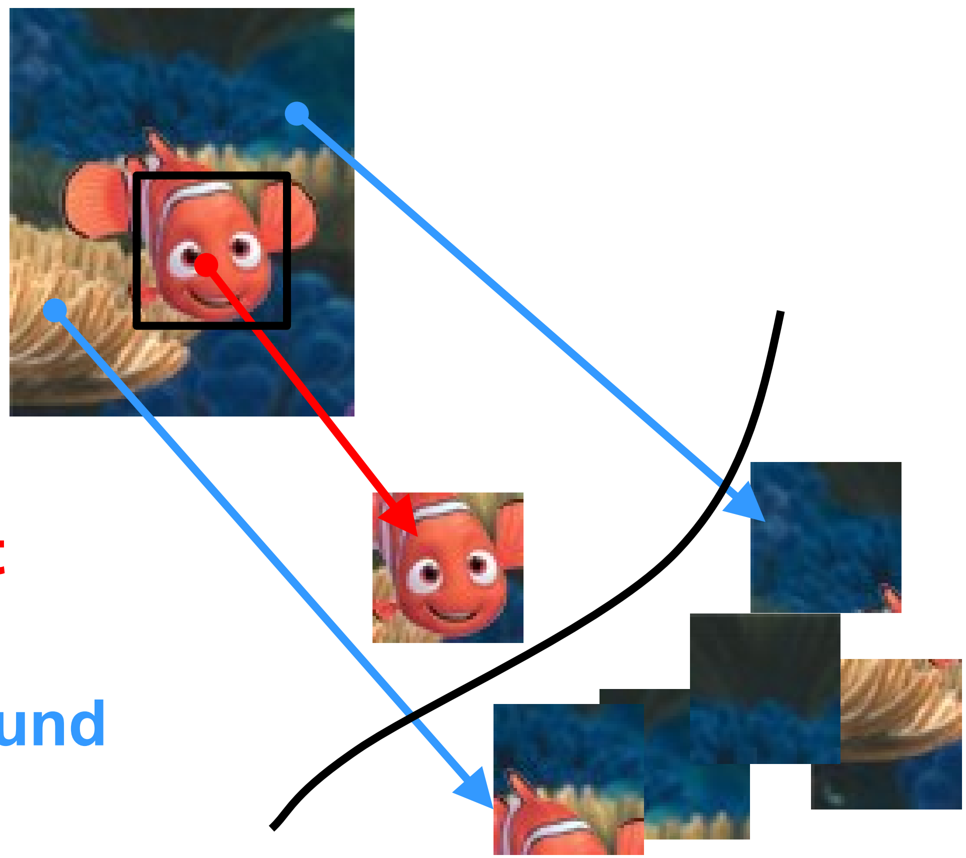
  - Still open dilemma: adapt when required, not during occlusions…
- Probabilistic version
  - Coupled with Kalman or particle filter for better handling of occlusions
- Automatic multi-objet tracking
  - Detection of a category of objects
  - Sequential tracking or batch “tracking”
  - Handling of multiple trackers (exclusion principle, multi-objet occlusion)

slide credit:
Patrick Perez

Tracking as classification

- Tracking as binary classification

S. Avidan. **Ensemble tracking**. CVPR 2005.
J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.



object

vs.

background

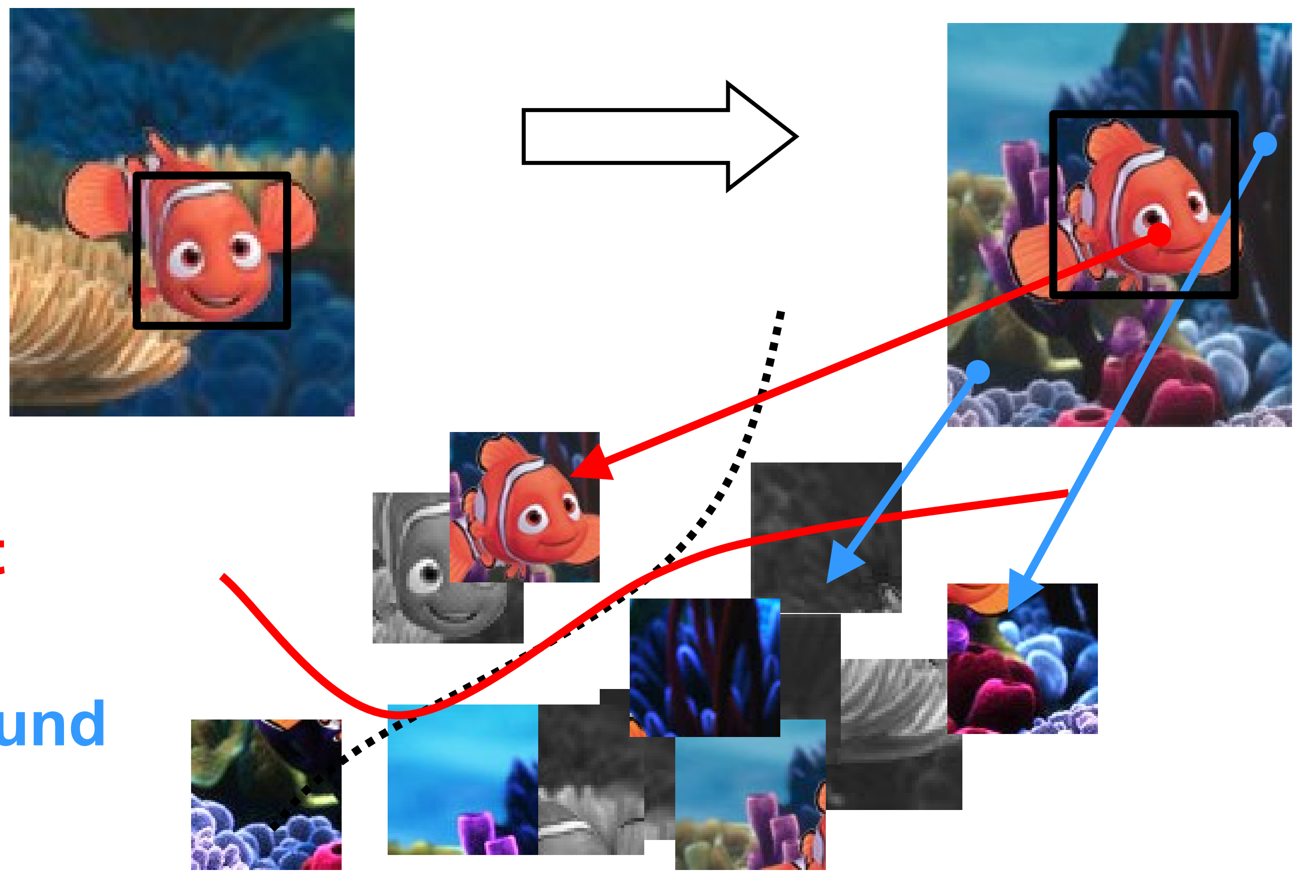
Slide credit: Helmut Grabner

Online discriminative tracking

- Tracking as binary classification

S. Avidan. **Ensemble tracking**. CVPR 2005.
J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.

- Object <u>and</u> background changes are robustly handled by on-line updating!



object

vs.

background

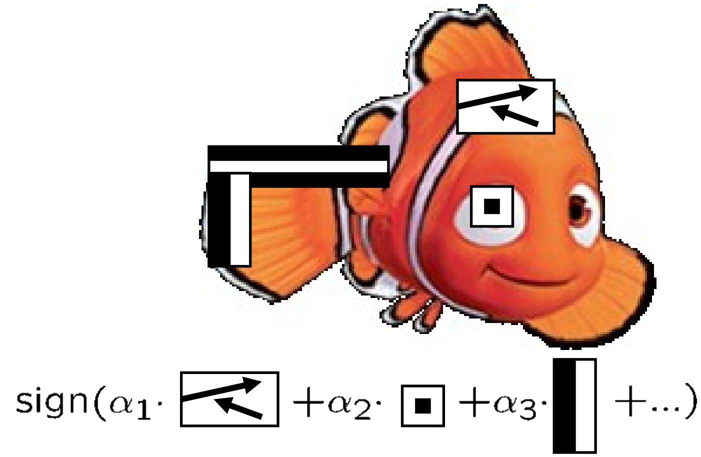
Slide credit: Helmut Grabner

Boosting for Feature Selection

Object Detector

P. Viola and M. Jones. **Rapid object detection using a boosted cascade of simple features.** CVPR 2001.

Fixed Training set
General object detector



Combination of simple image features using Boosting as Feature Selection

Object Tracker

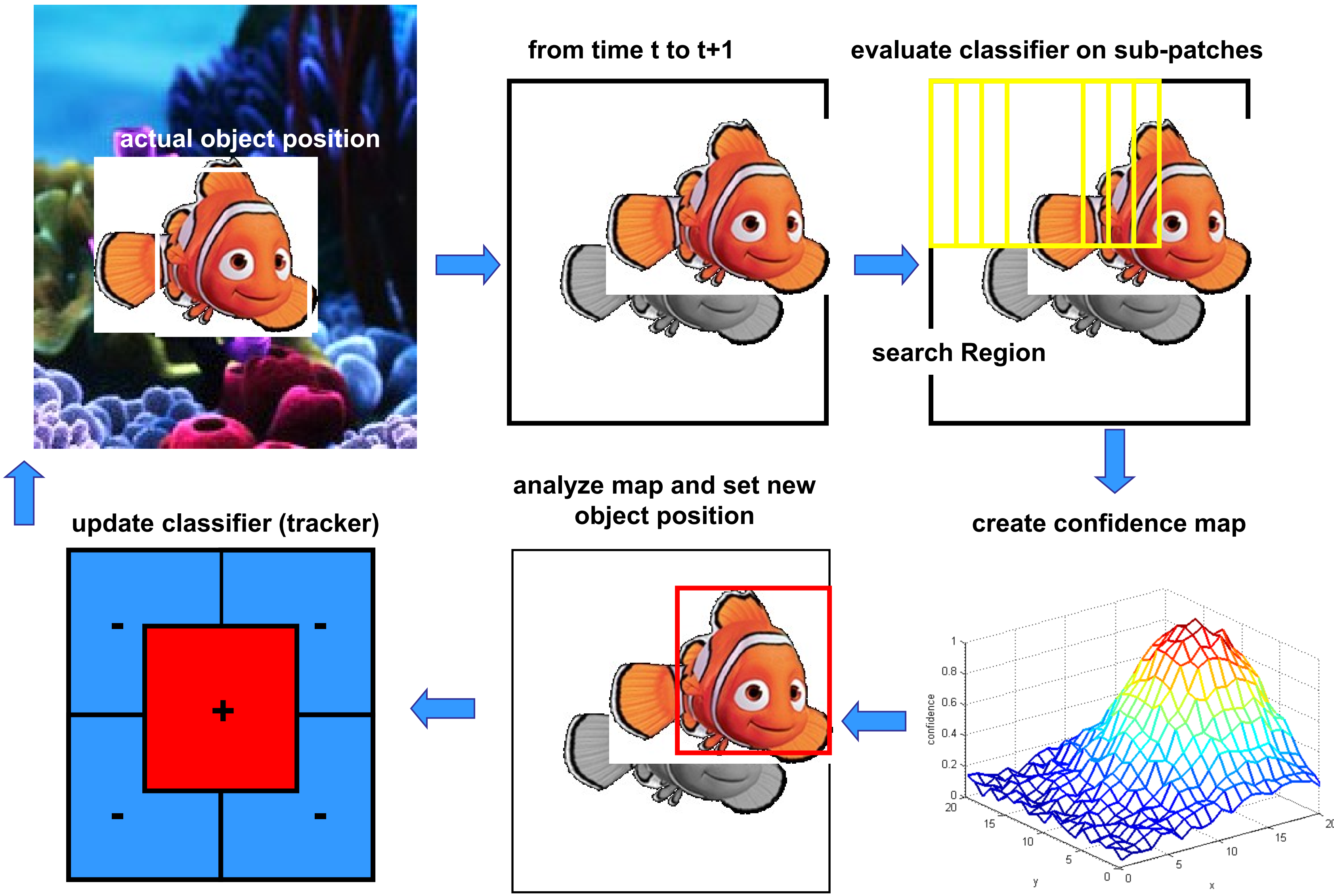
On-line update
Object vs. Background

On-Line Boosting for Feature Selection

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

Slide credit: Helmut Grabner

Tracking by online Adaboost



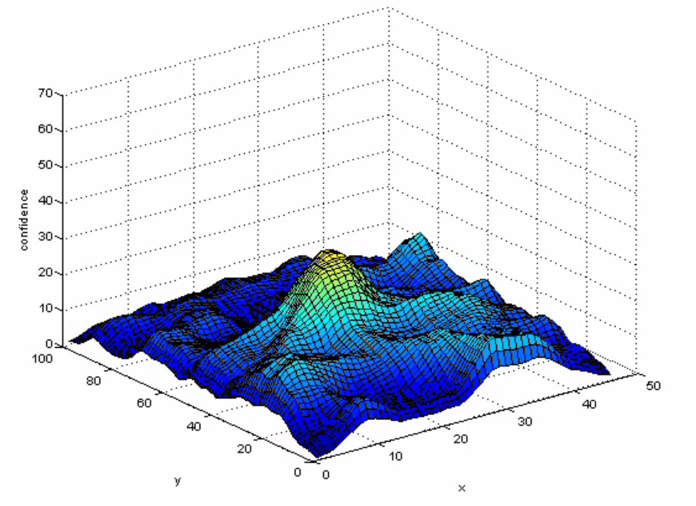
H. Grabner et. al., Real-Time Tracking via On-line Boosting . BMVC, 2006.

Slide credit: Helmut Grabner

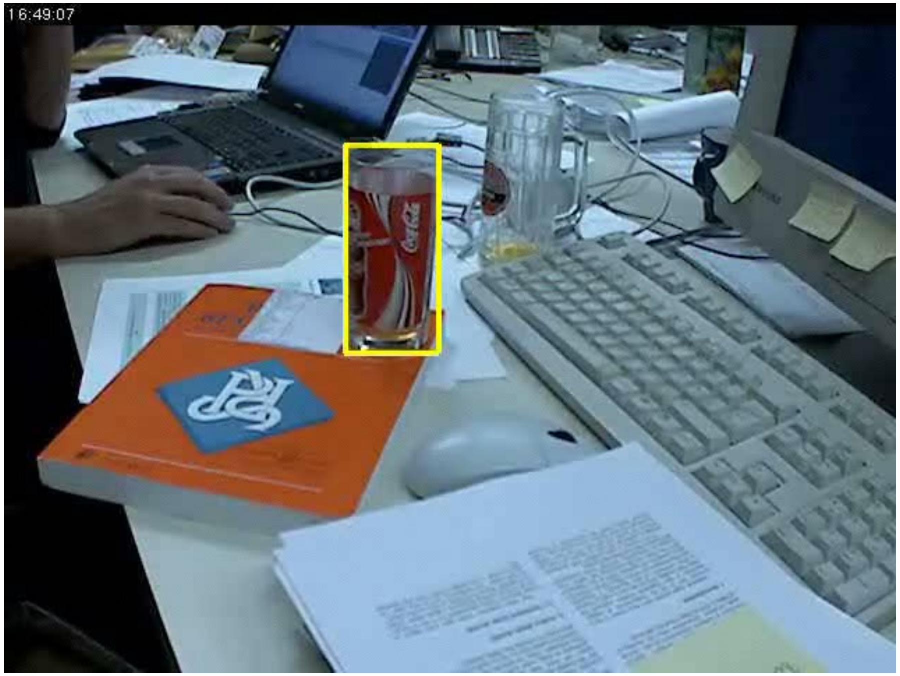
Tracking by online Adaboost

- Realtime performance
  - Fast feature computation
  - Efficient update of classifier

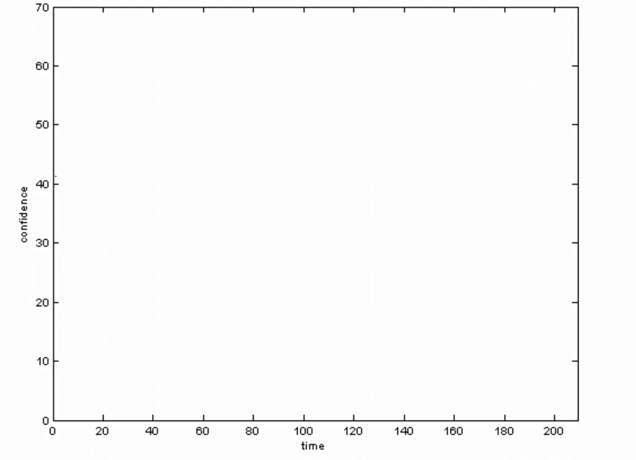
Confidence Map



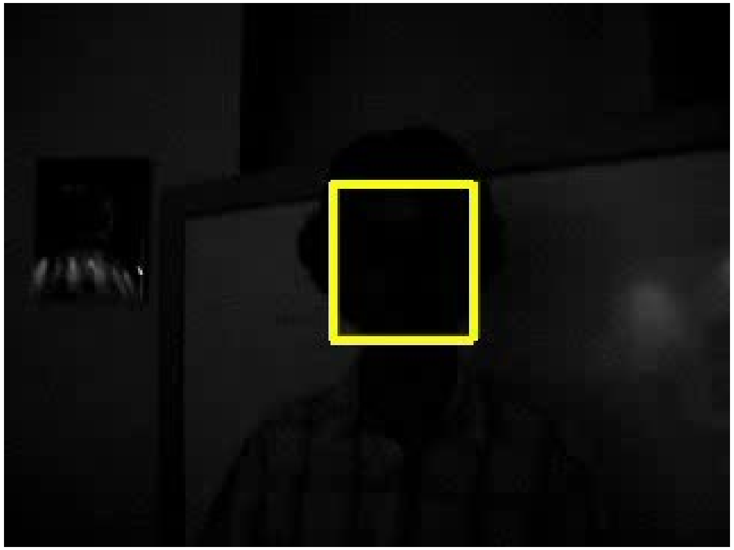
Tracking

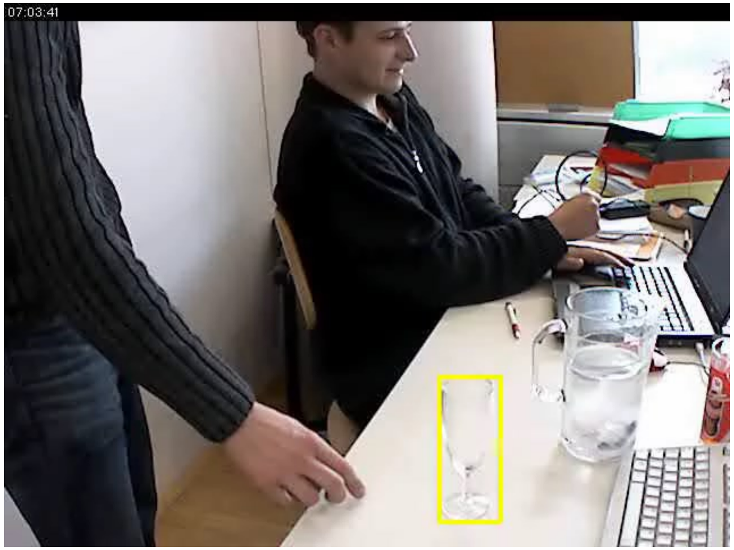


Max. Confidence Value



Tracking by online Adaboost

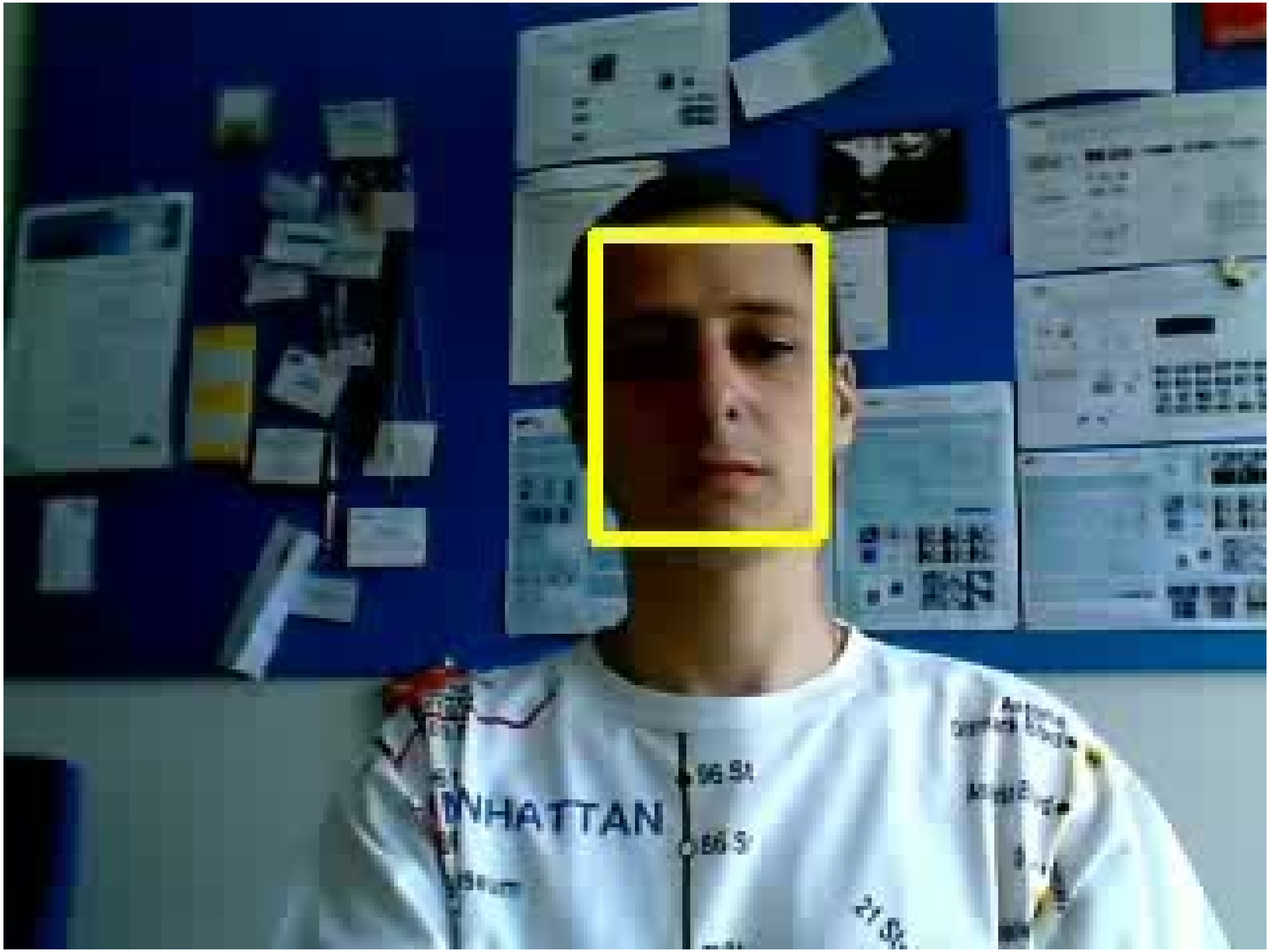




H. Grabner et. al., Real-Time Tracking via On-line Boosting . BMVC, 2006.

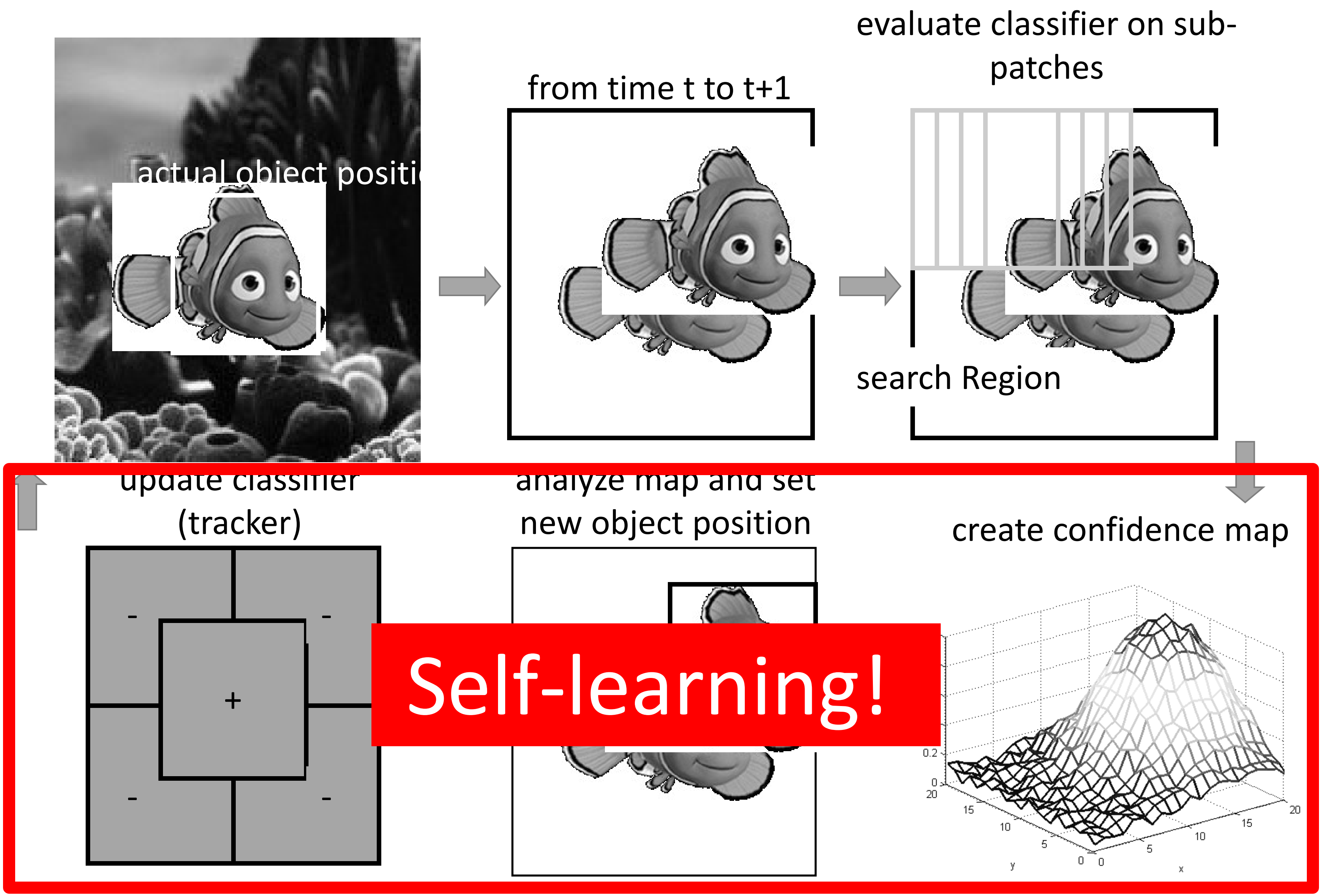
Slide credit: Helmut Grabner

Failure modes



Slide credit: Helmut Grabner

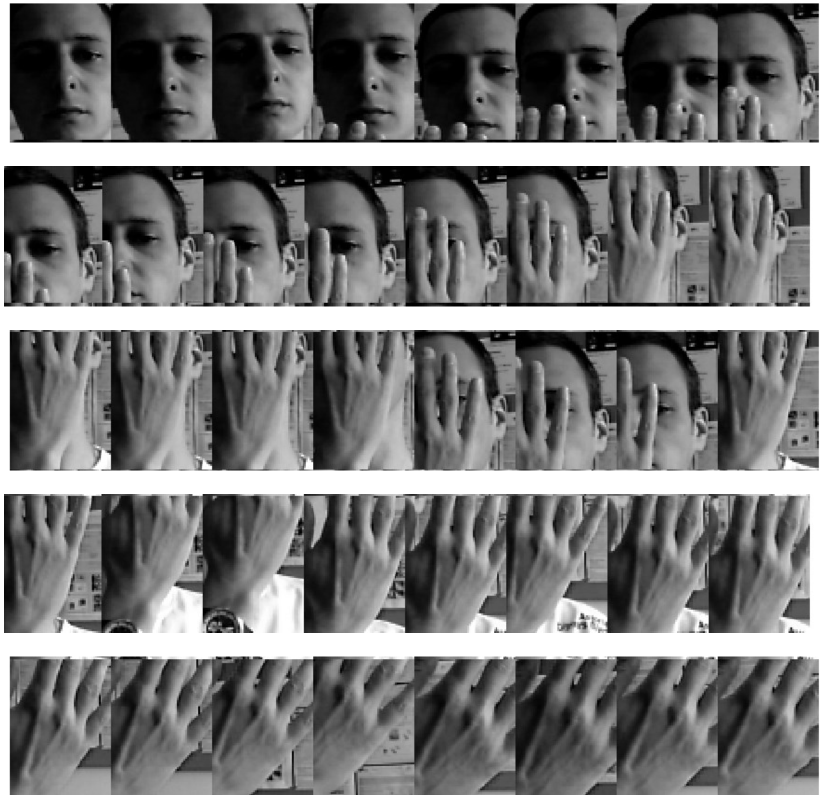
Why does it fail…



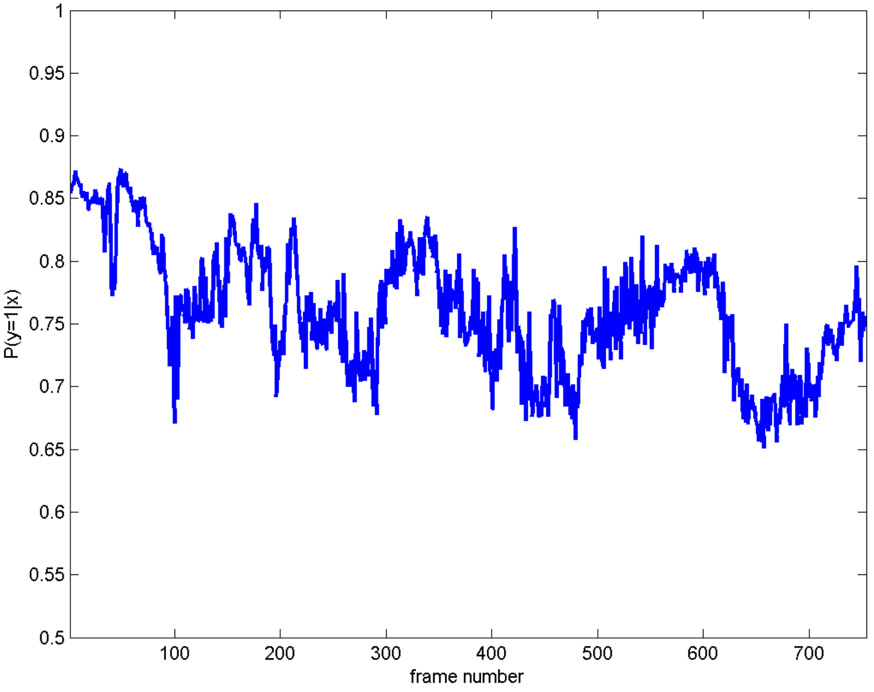
Slide credit: Helmut Grabner

Constant self-adaptation leads to drifting

Tracked Patches



Confidence



Slide credit: Helmut Grabner

Constant self-adaptation leads to drifting

- A poor update at time-step k may lead to poor localization at k+1
- This leads to even a poorer update, etc.

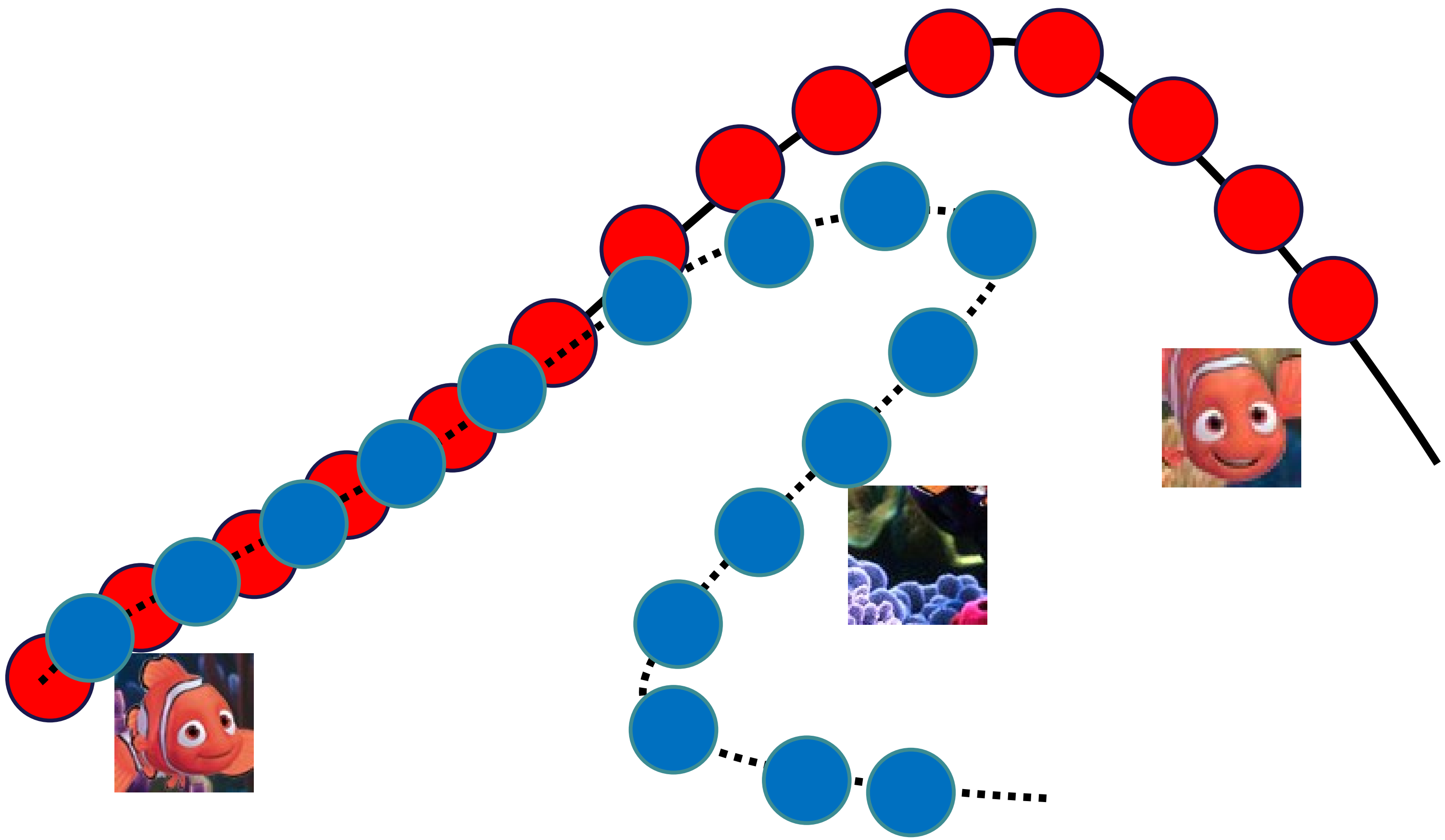
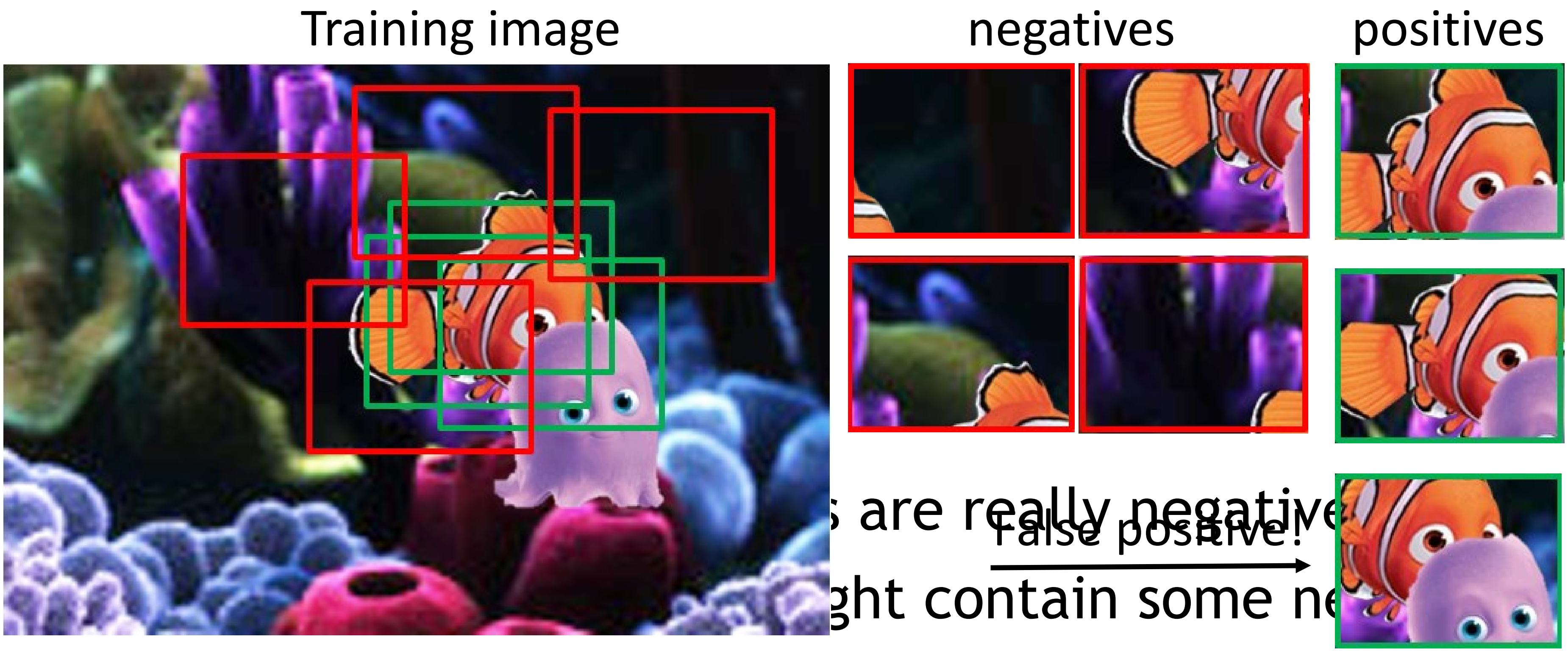


Image credit: Helmut Grabner

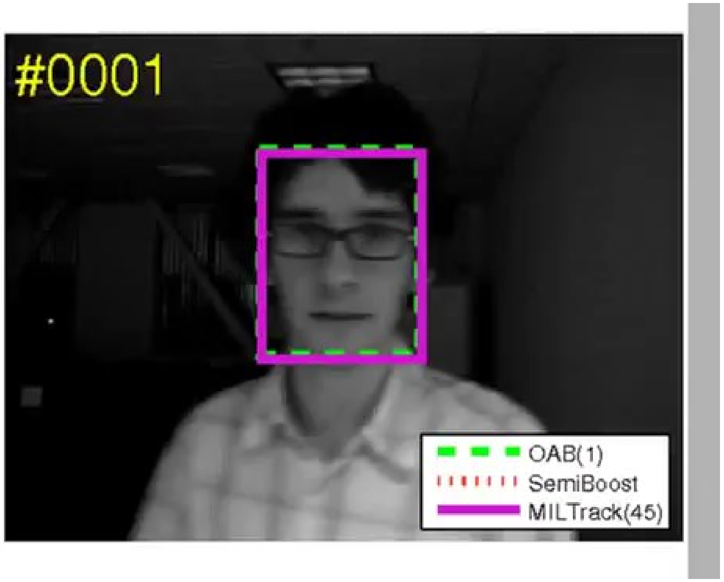
Do not trust all learning examples

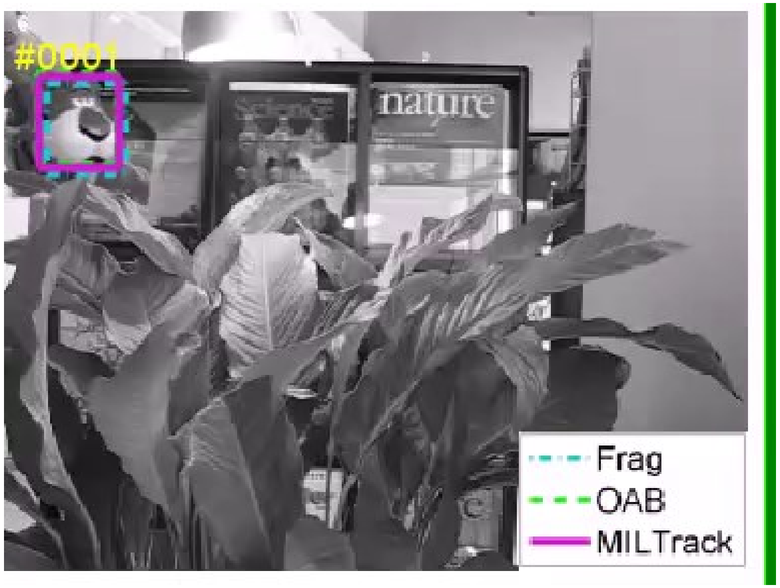


- A multiple instance learning problem!

Babenko et al.,"[Robust Object Tracking with Online Multiple Instance Learning]", TPAMI2011

Do not trust all learning examples





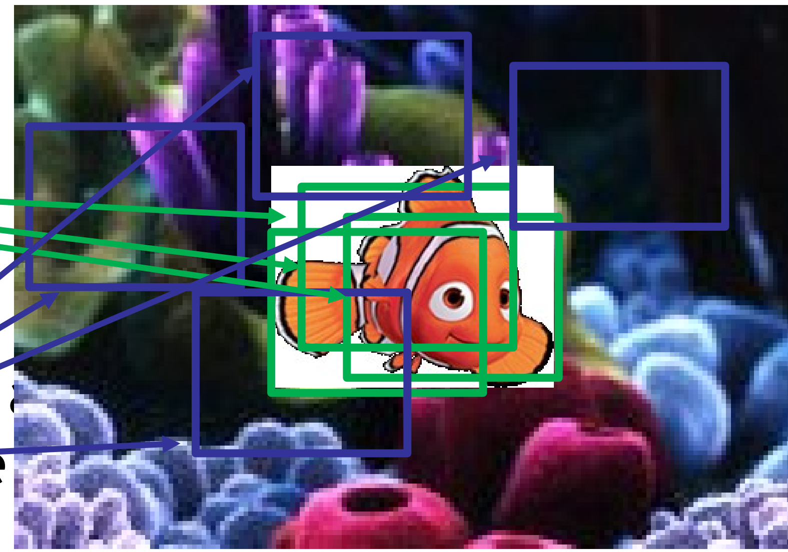
- Note that the online Adaboost failed *in this run* on the David sequence!
- Be sure that TMIL authors worked to show this, but it also says a lot about robustness of oAB to initialization!
- Code for TMIL available here.

Babenko et al.,"Robust Object Tracking with Online Multiple Instance Learning", TPAMI2011

Apply weights to training examples

- Online AdaBoost and TMIL make hard decision on the class identity :

$p(c_1) = 1$

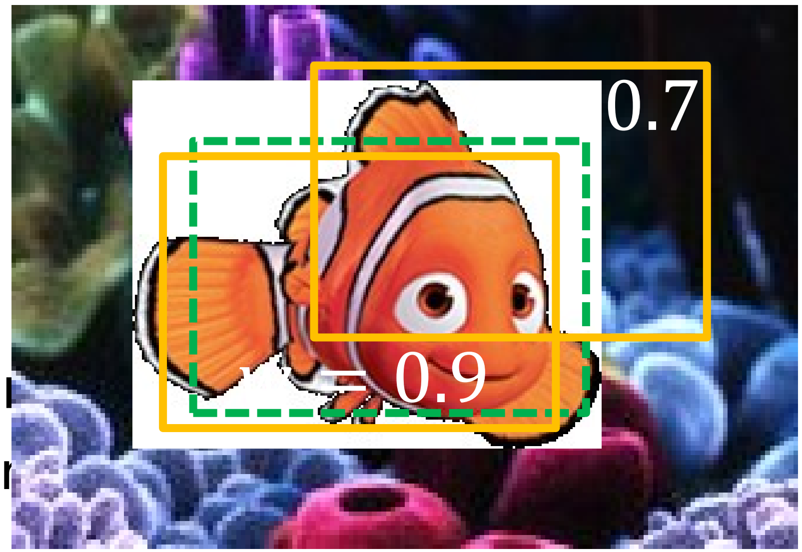


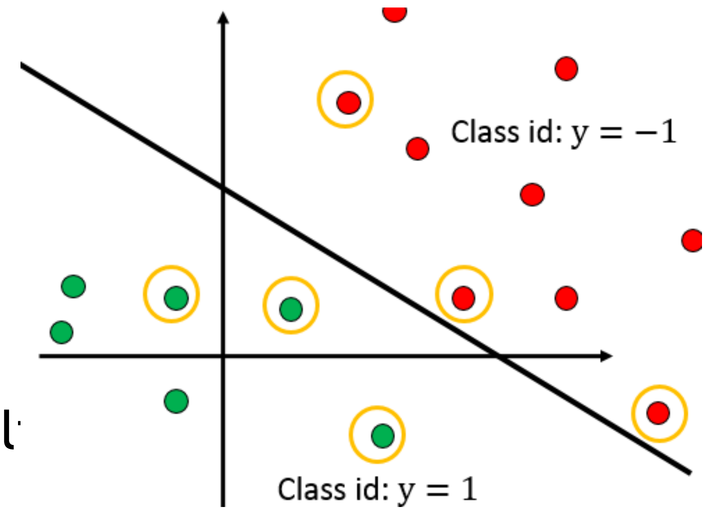
$p(c_1) = 0$

- But some positive examples a… rs and some negative examples are … ..

Apply weights to training examples

- Weights proportional to estimated position overlap:

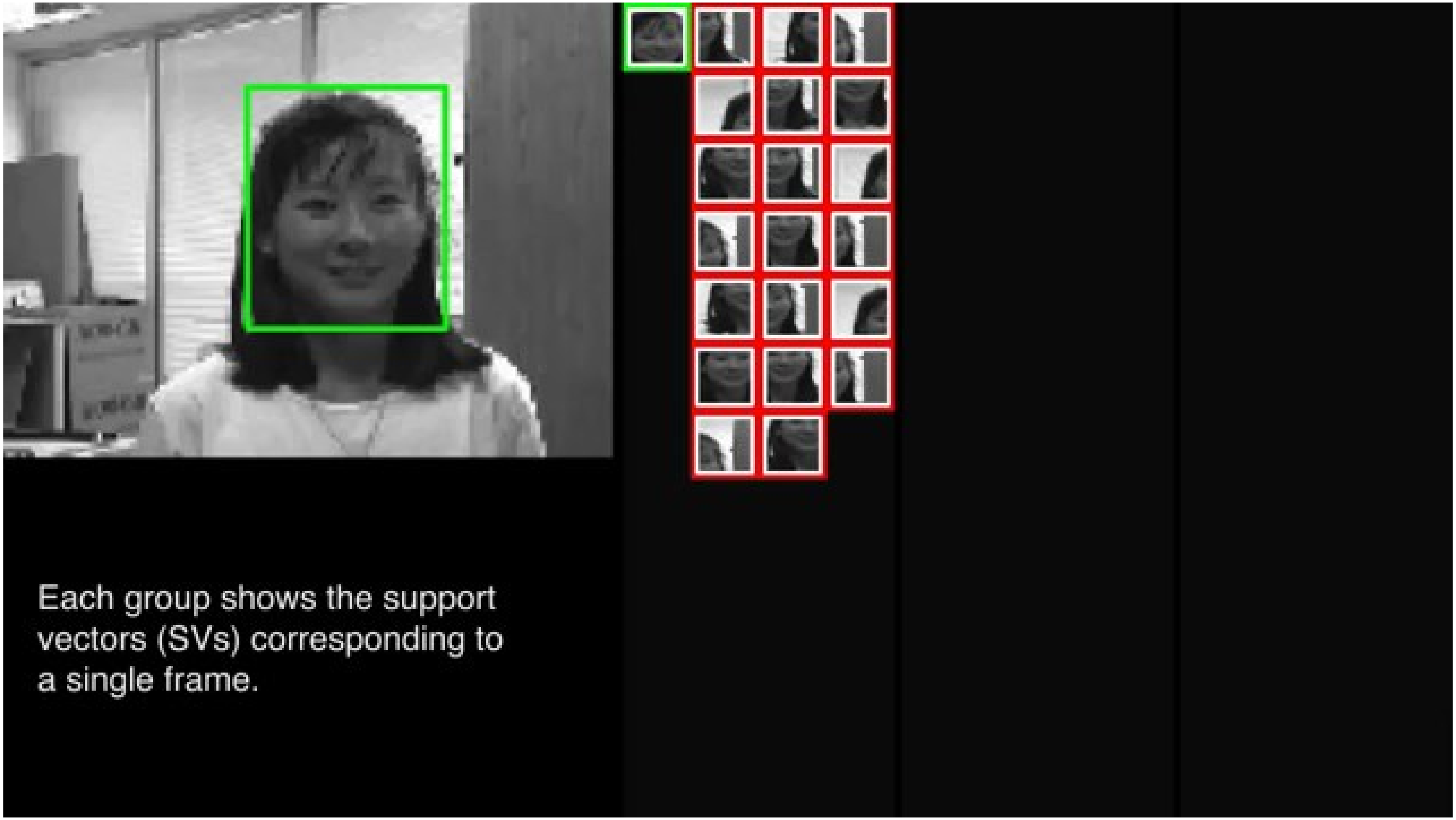




- Lea
  – Str (onl

Sam Hare, Amir Saffari, Philip H. S. Torr, Struck: Structured Output Tracking with Kernels, ICCV 2011

Struck tracking example

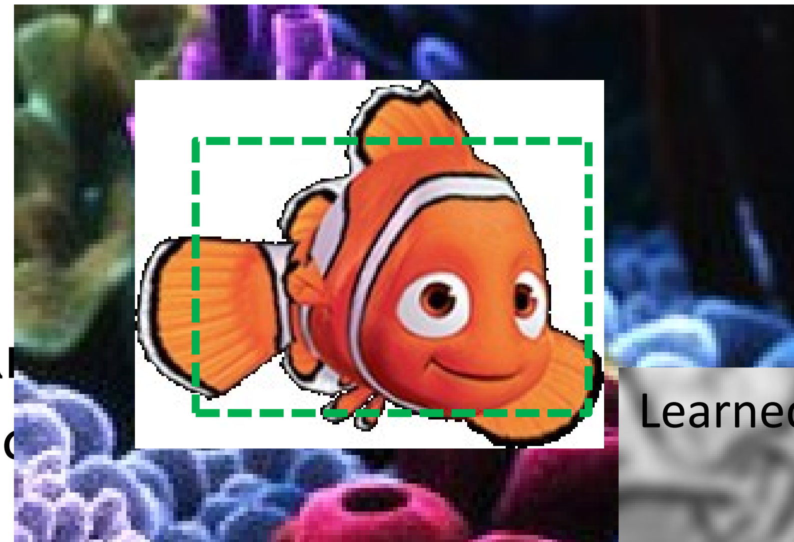


Sam Hare, Amir Saffari, Philip H. S. Torr, [Struck: Structured Output Tracking with Kernels](), ICCV 2011

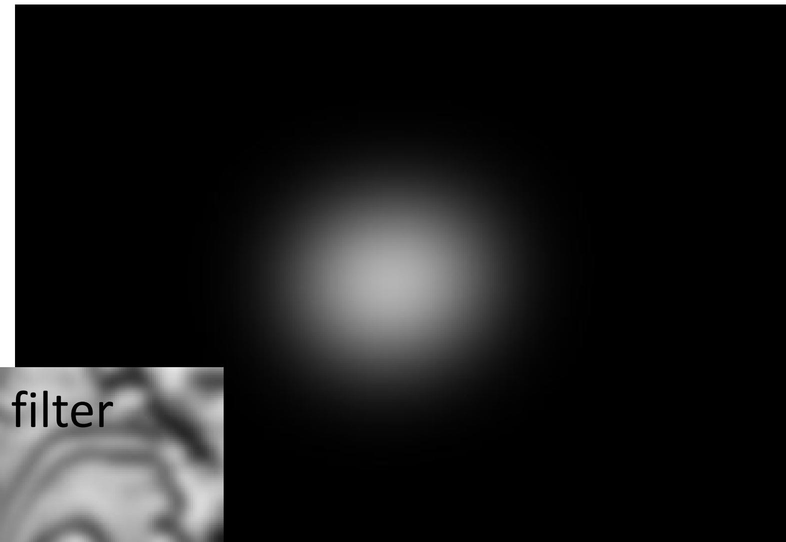
Apply weights to training examples

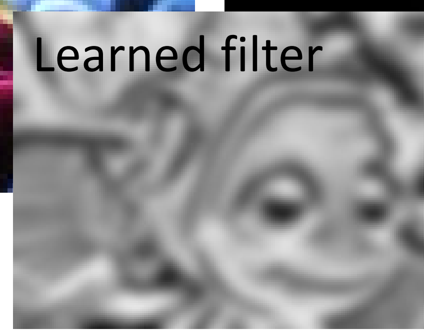
- Weights proportional to distance from the estimate:

Training image



Weights for all displacements

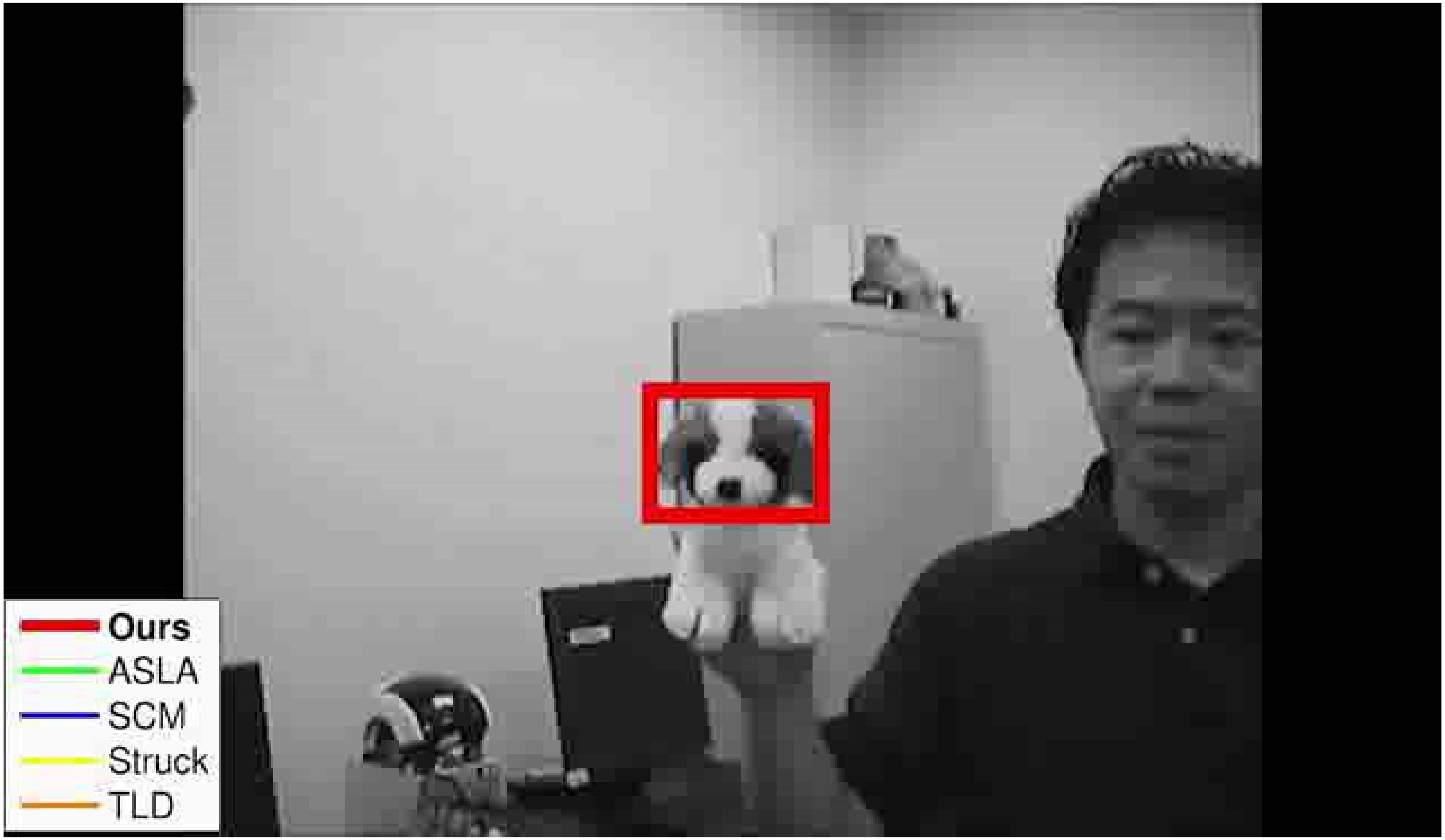




- Lea
  - Ri

* Bolme, Beveridge, Draper, and Y. M. Lui. Visual Object Tracking using Adaptive Correlation Filters. CVPR 2010.
* Henriques, Caseiro, Martins, Batista, High-Speed Tracking with Kernelized Correlation Filters TPAMI2015
* Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. BMVC2014

Correlation filters tracking example



Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: [Accurate scale estimation for robust visual tracking](). BMVC2014

References

- Key-point-based tracking:
  - [1] Özuysal, Calonder, Lepetit, Fua: Fast Keypoint Recognition Using Random Ferns. TPAMI2010
- Online Adaboost for tracking:
  - [2] H. Grabner et. al., Real-Time Tracking via On-line Boosting . BMVC, 2006.
- Multiple instance learning for tracking:
  - [3] Babenko et al.,"Robust Object Tracking with Online Multiple Instance Learning", TPAMI2011
- Structured SVM tracking:
  - [4] Hare, Saffari, Torr, Struck: Structured Output Tracking with Kernels, ICCV 2011
- Correlation filter tracking:
  - [5] Bolme, Beveridge, Draper, and Y. M. Lui. Visual Object Tracking using Adaptive Correlation Filters, CVPR 2010.
  - [6] Henriques, Caseiro, Martins, Batista, High-Speed Tracking with Kernelized Correlation Filters, TPAMI2015
  - [7] Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking, BMVC2014

The Flock of Trackers (with error prediction)

work with T. Vojir

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*

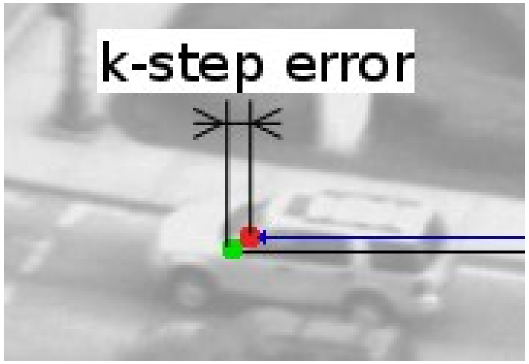
Two classical Failure Predictors

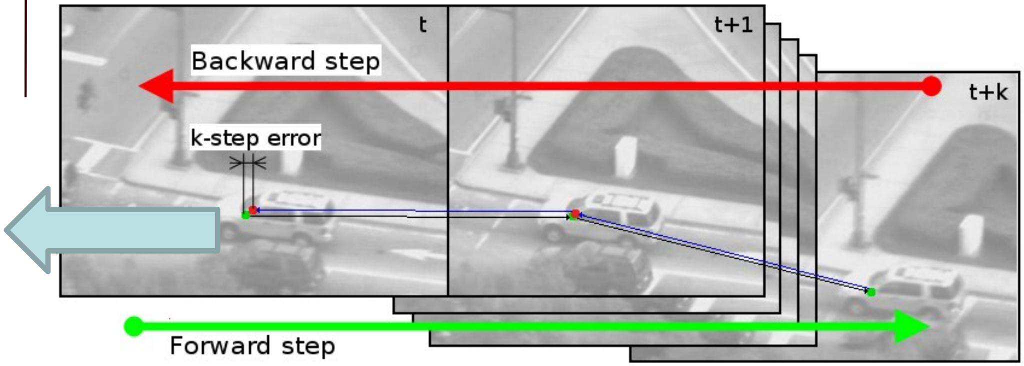
Normalized Cross-correlation

- 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

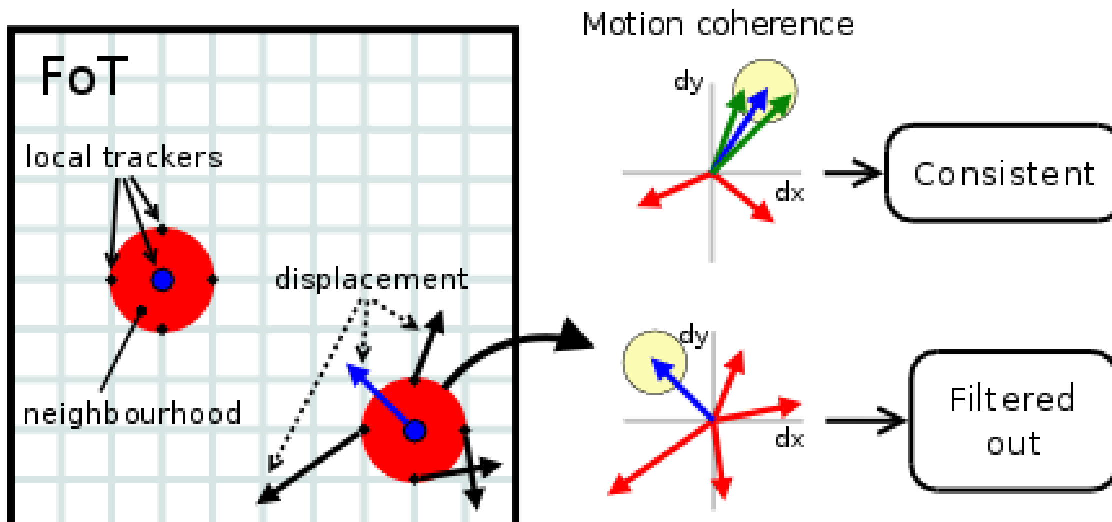
- For each local tracker *i* is computed neighbourhood consistency score as follows :

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

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

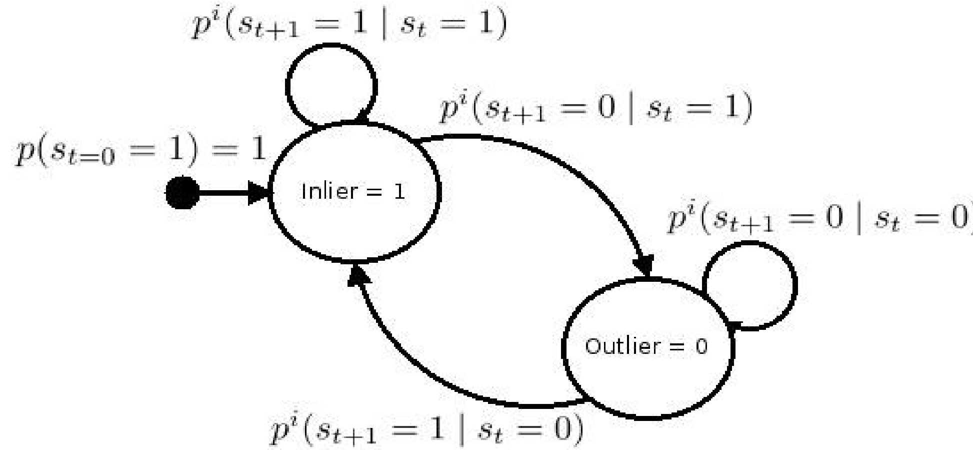
- Local trackers with $S_i^{Nh} < \Theta_{Nh}$ are filtered out

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



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

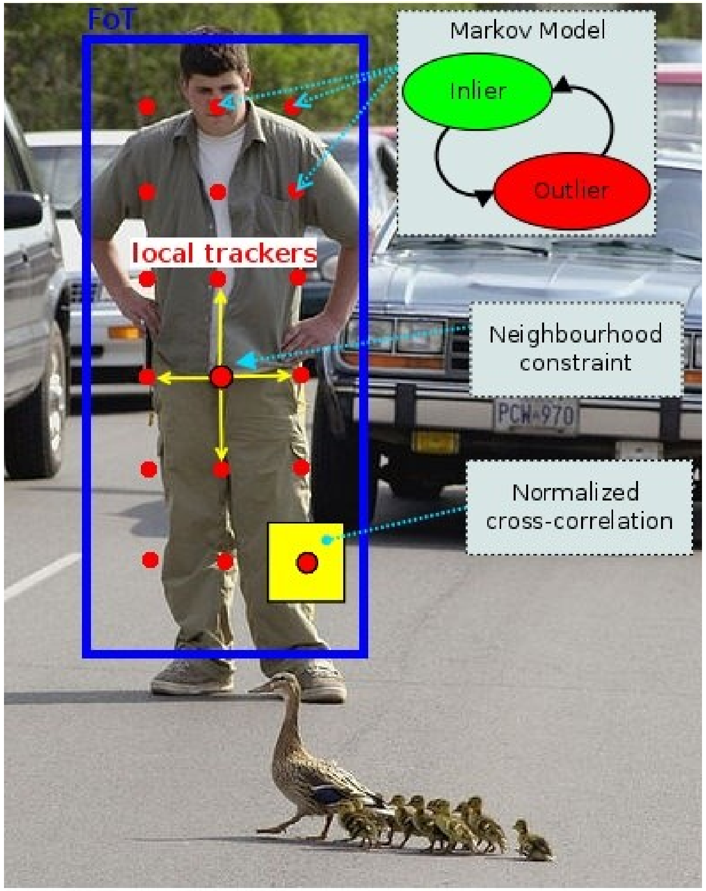
- Learning is done incrementally (learns are the transition probabilities between states)

- 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)**

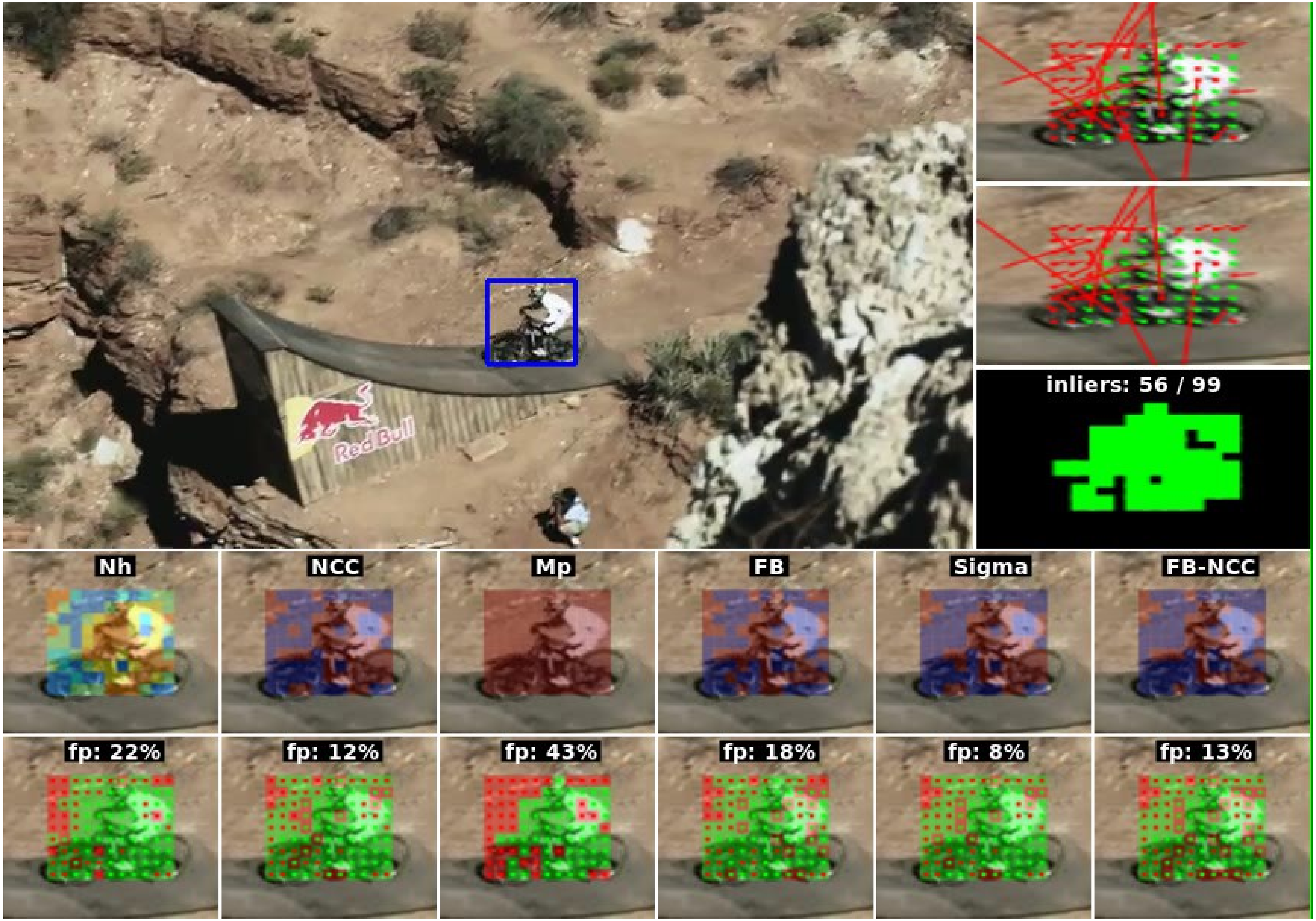


- Together form very a stronger predictor than the popular forward-backward

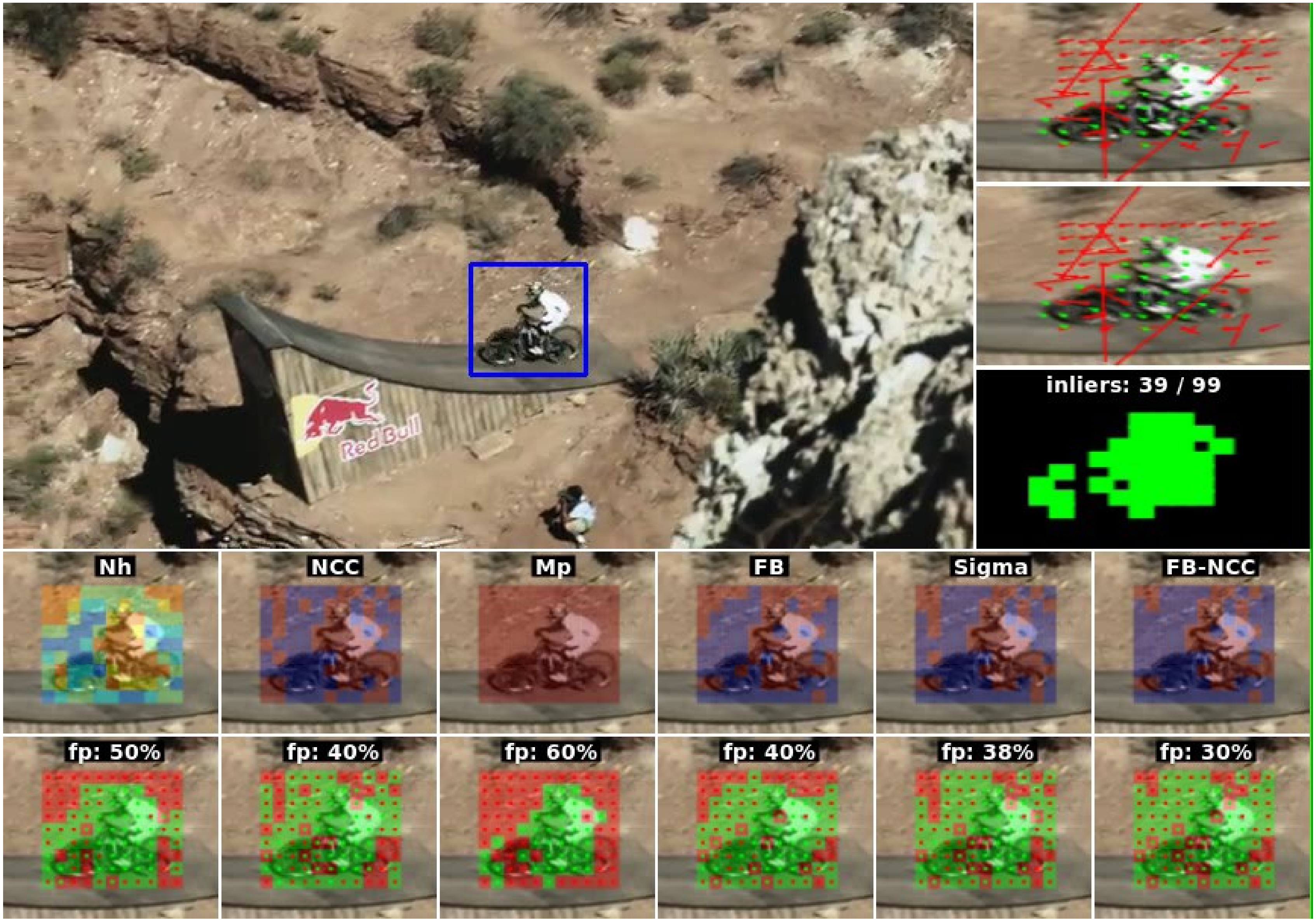
- Negligible computational cost (less than 10%)

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

FoT Error Prediction Bike tight box (ext. viewer)

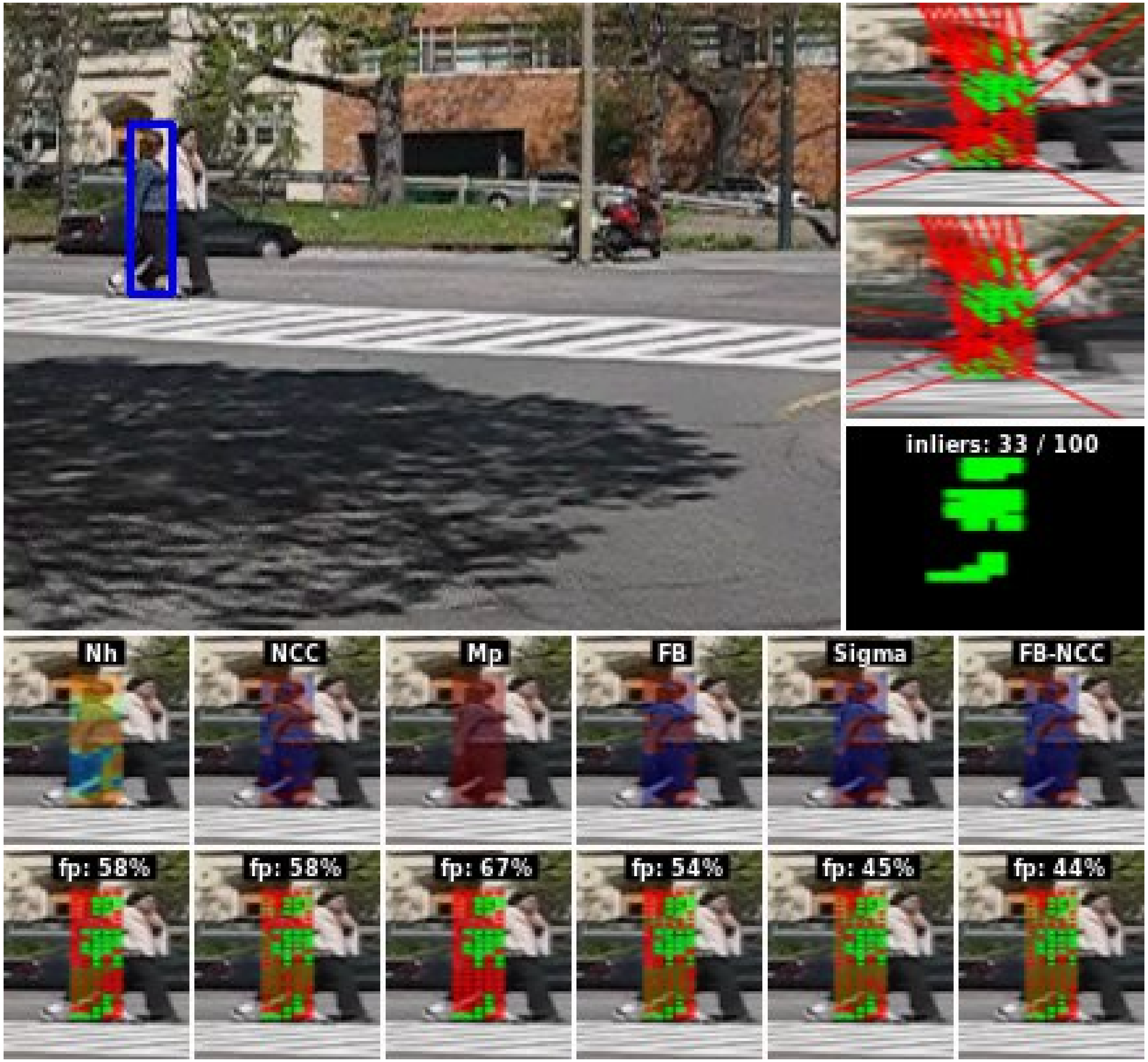


FoT Error Prediction Bike loose box (ext. viewer)



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**

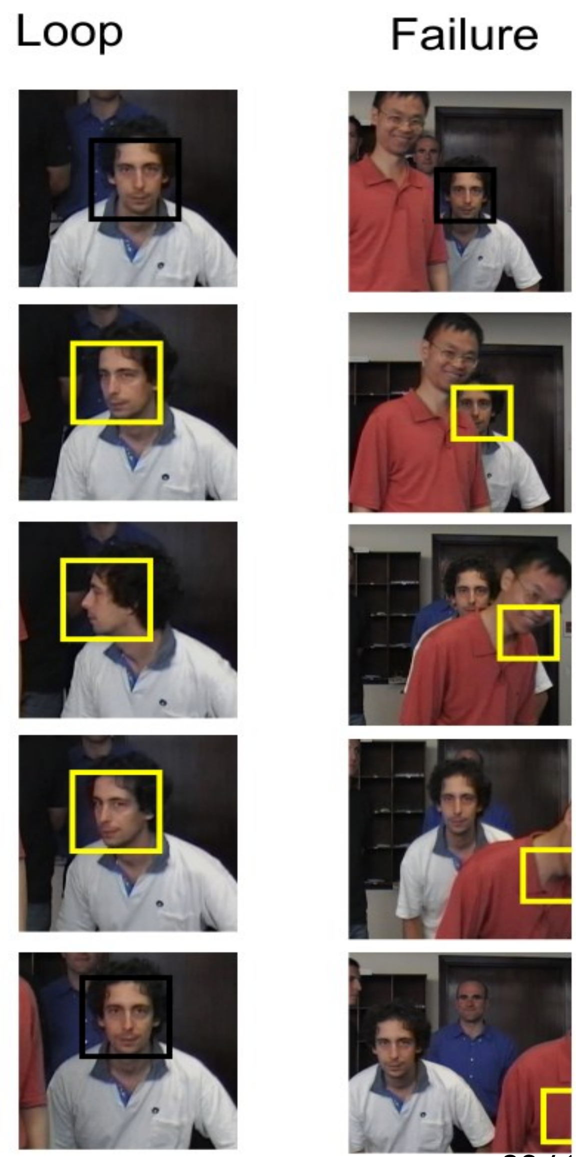
TLD a.k.a. PN Tracker a.k.a. “The Predator”

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 th
  added to the model

Tracker responses



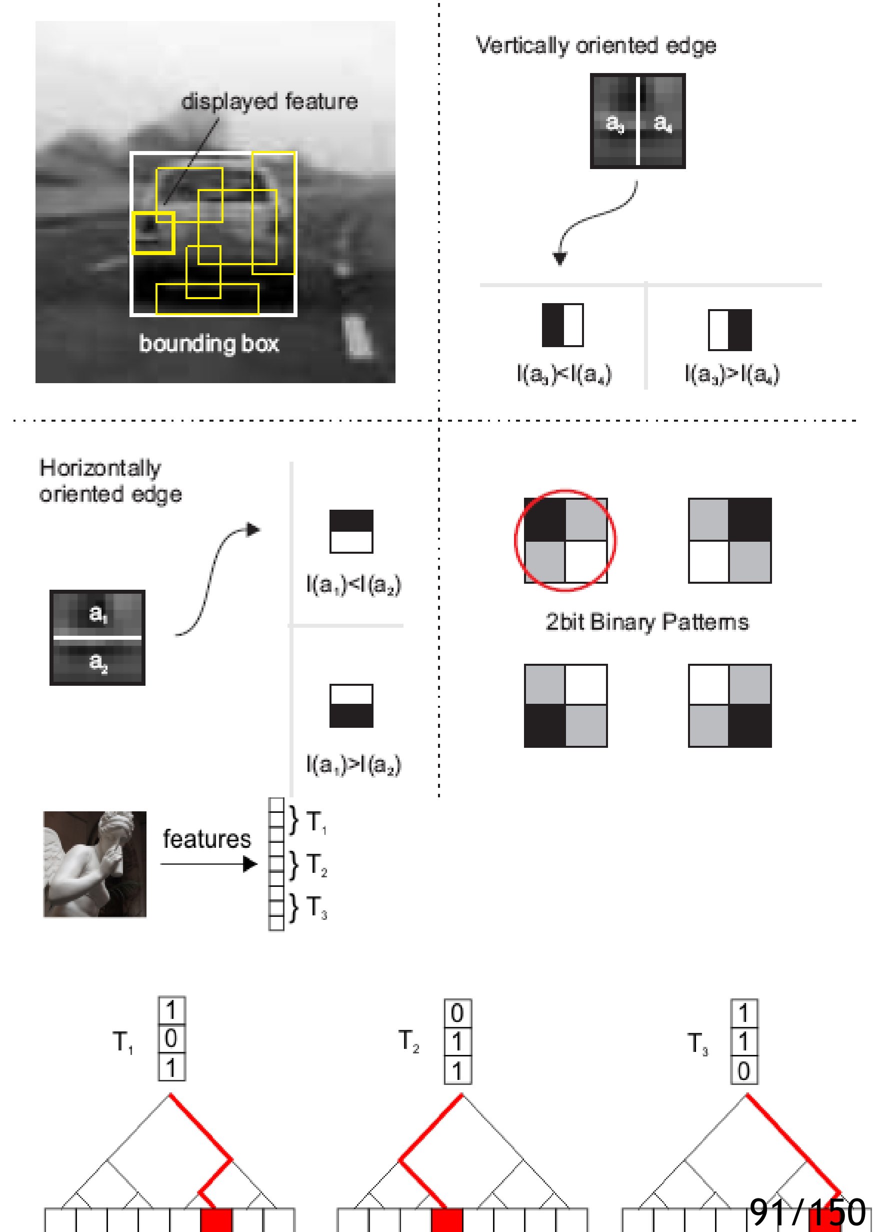
N-event: Uniqueness Enforcement

- 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

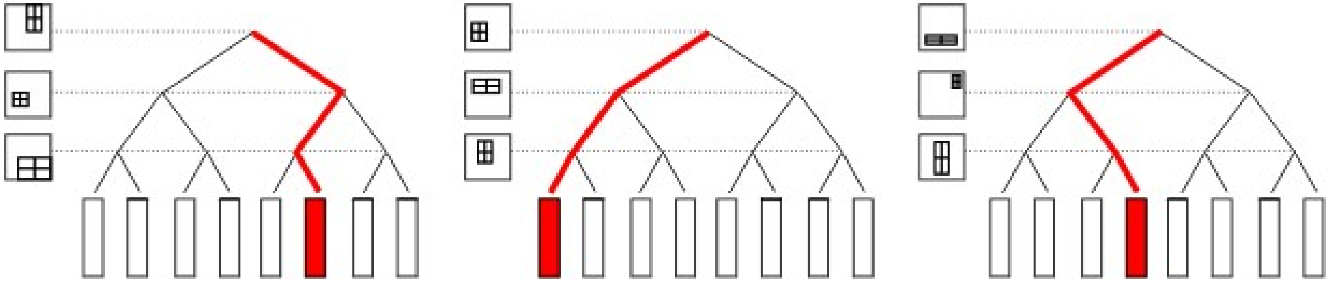
The Detector

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

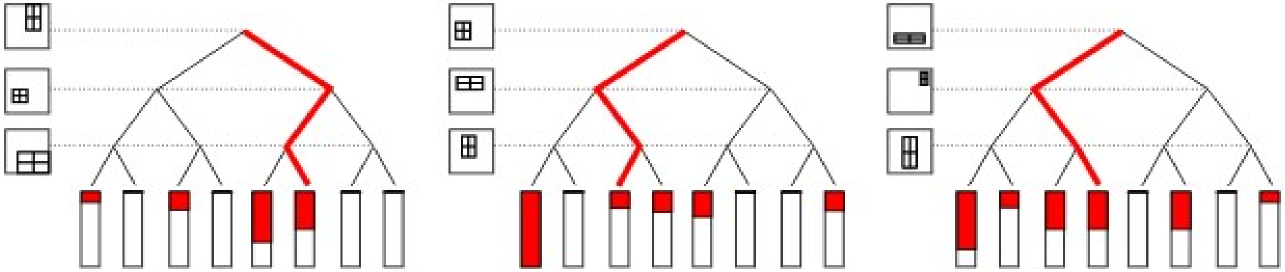
- High accuracy, 8 trees of depth 10
- 2bit Binary Patterns Combined Haar and LBP features
- Tree depth controls complexity & discriminability; currently not adaptive

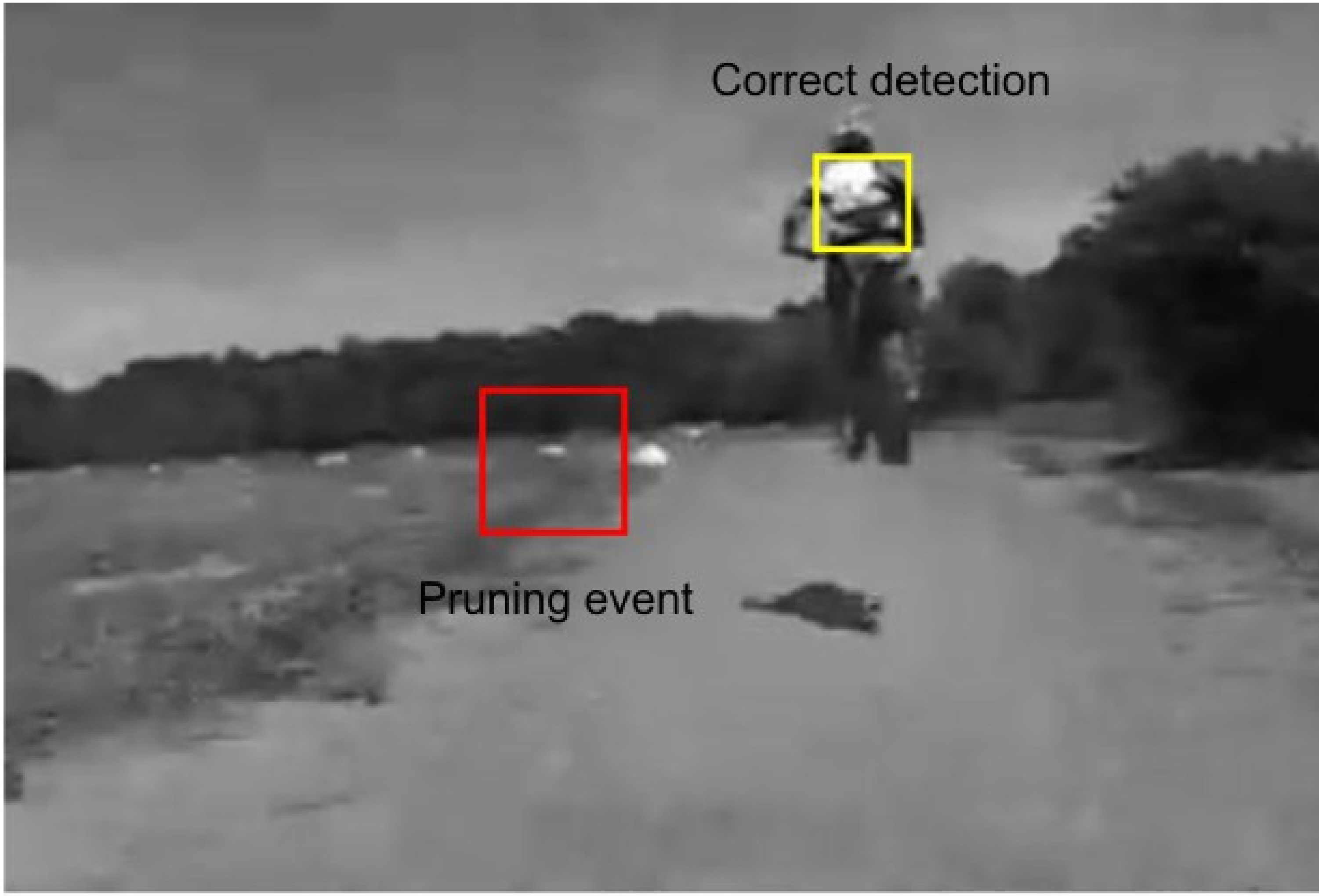


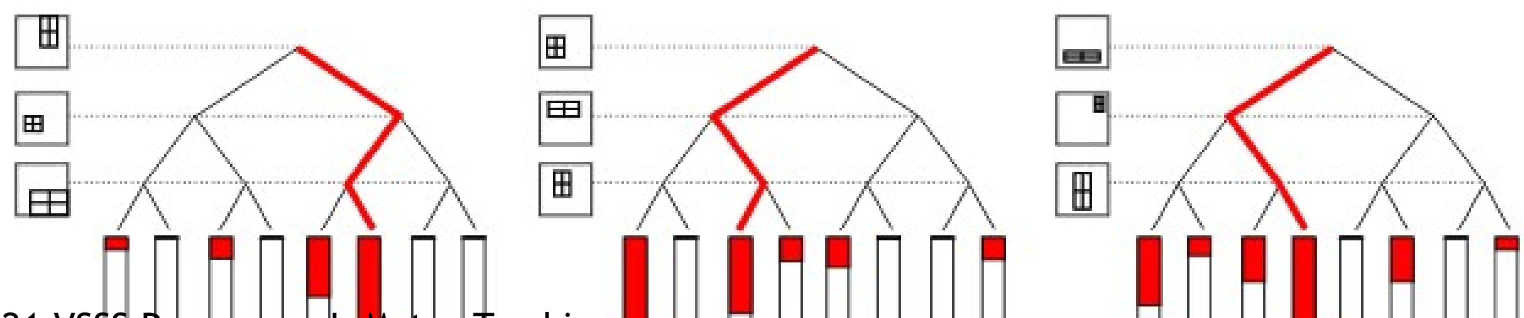

Growing event



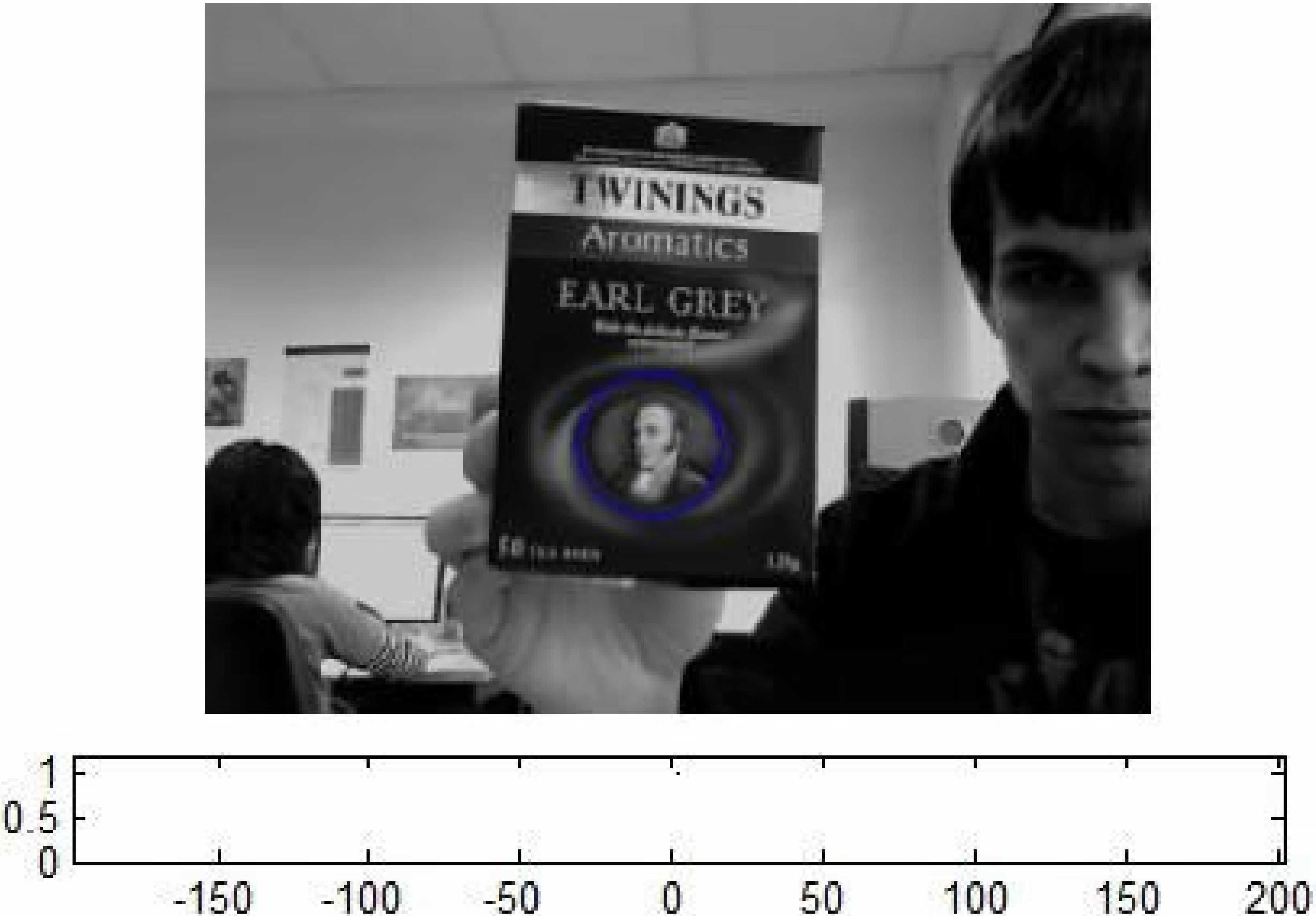

Correct detection

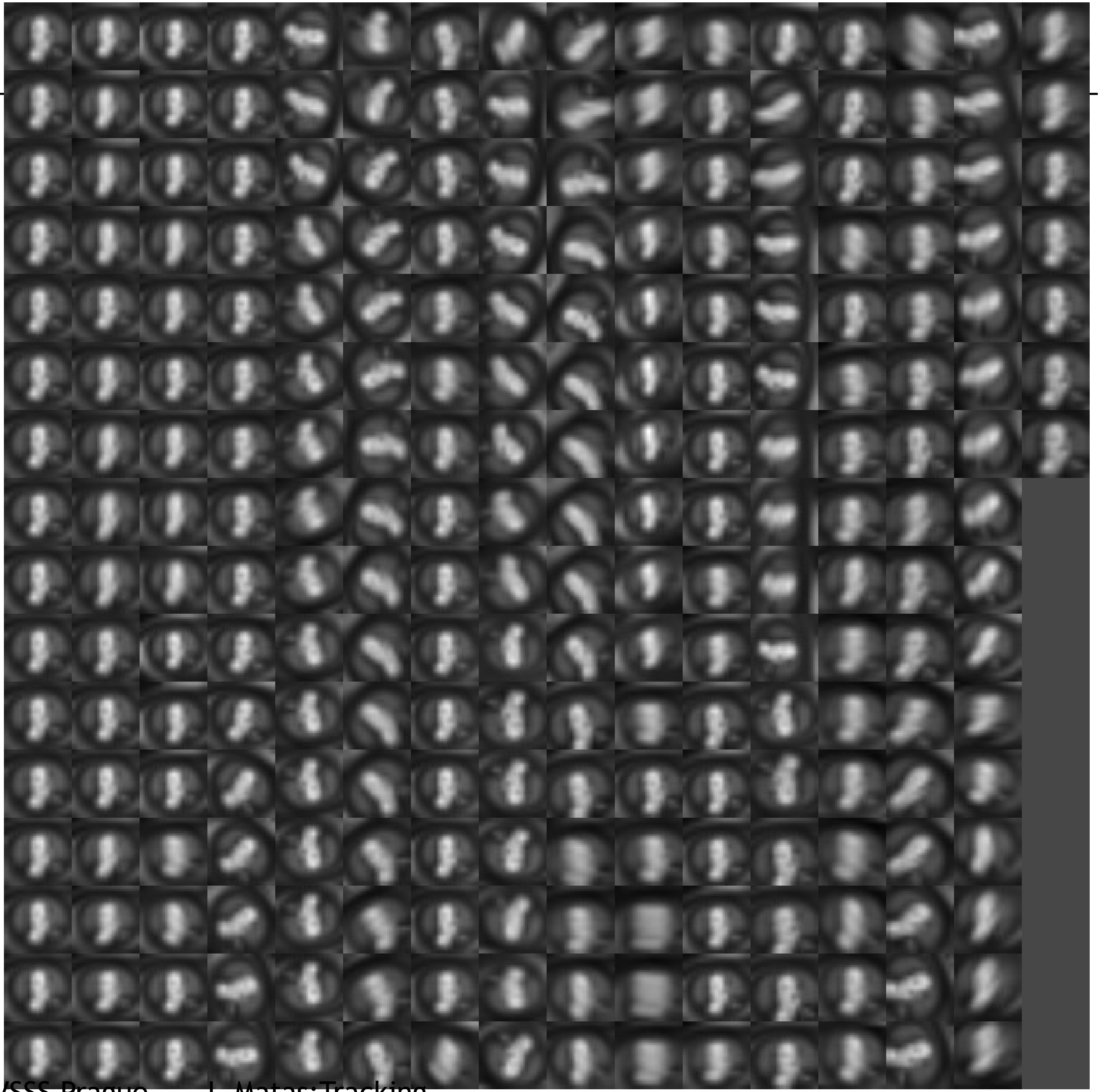



Correct detection
Pruning event



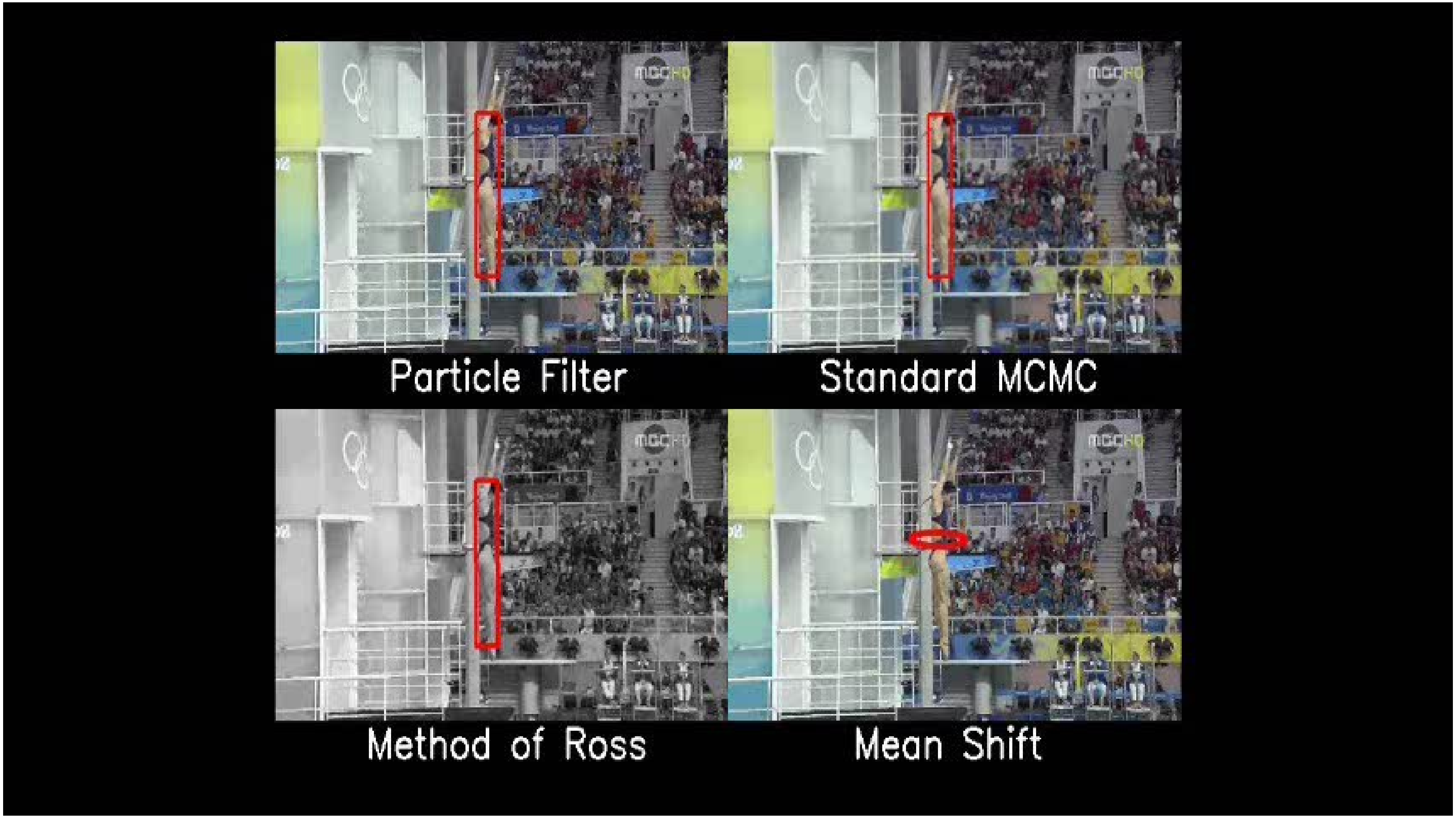
2





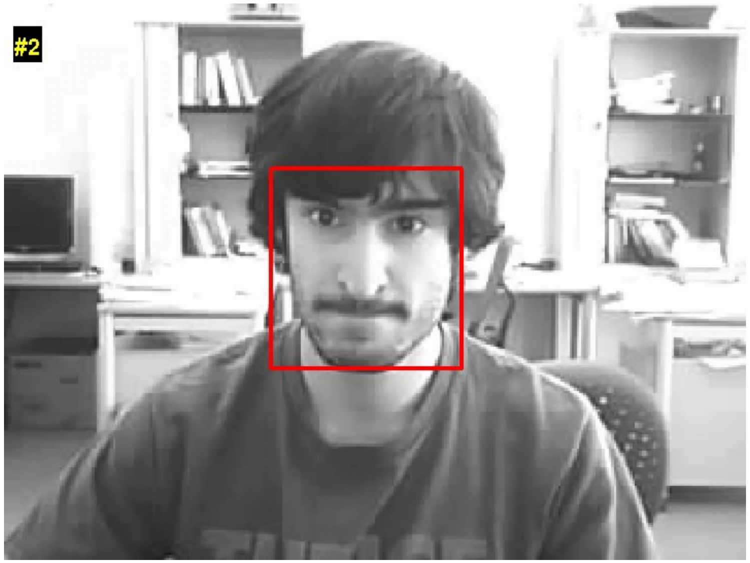
Tracking: Which methods work?

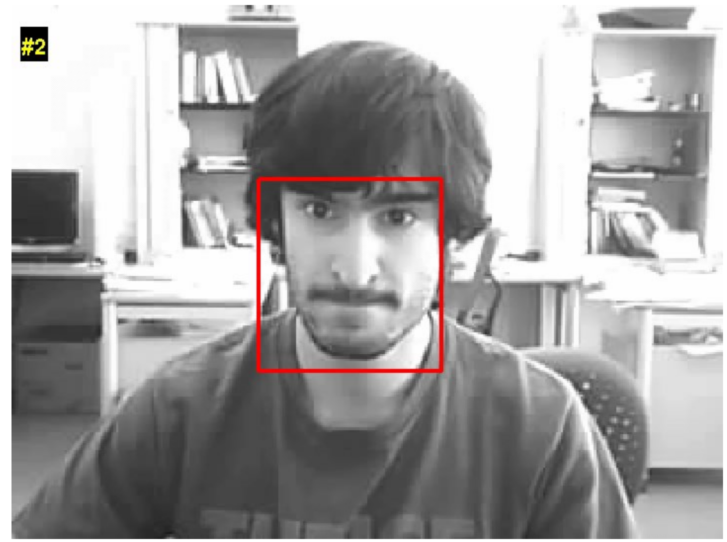
Tracking: Which methods work?

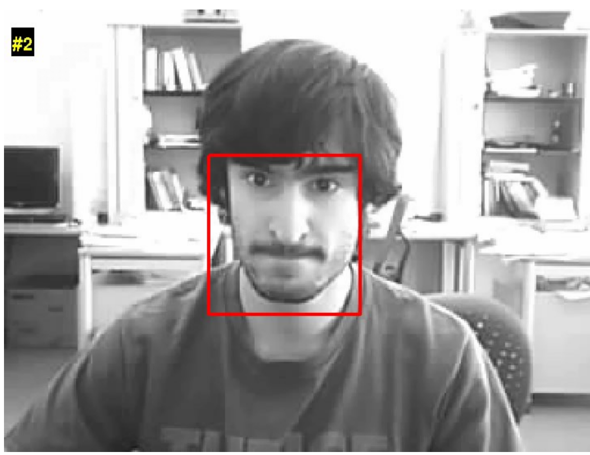


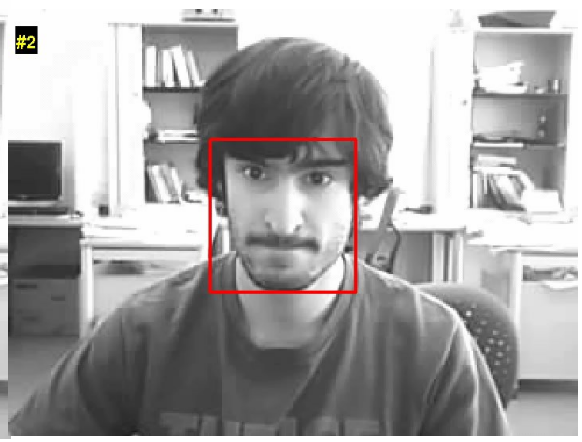
What works? “The zero-order tracker” ☺

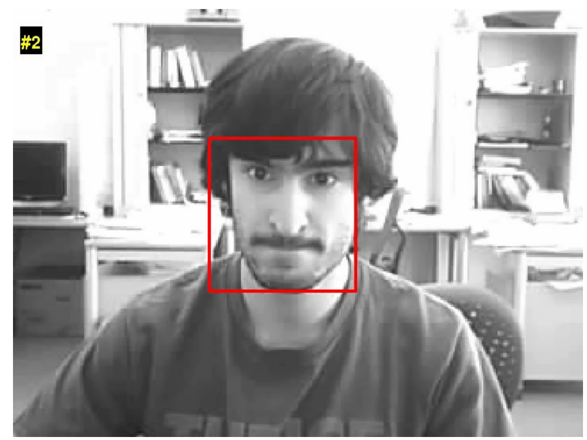
Compressive Tracker (ECCV'12). Different runs.











The Visual Object Tracking Challenge @ ICCV

In conjunction with ICCV 2013

VOT2013

IEEE workshop on visual object tracking challenge

December 2, 2013 - Sydney, Australia

Home | Results | Participation | Evaluation Kit | Submission | People | Supporters

Call for participation and for papers

Research on visual tracking remains limited due to the lack of standardised evaluation protocols and online reference repositories showing results on community accepted reference videos. VOT2013 invites researchers to participate in a first challenge focusing on single object visual tracking.

The aim of the accompanying VOT2013 workshop, which will be held in conjunction with ICCV2013, is to provide a common platform for comparison, analysis and discussion of existing as well as new single object trackers. We provide an evaluation kit that includes videos with annotated ground truth as well as evaluation software to automate the experiments and calculate the performance measures.

How to participate

The participants have to download the VOT2013 evaluation kit, and run the experiments on their tracker. The results of the experiments have to be submitted, along with a summarising description of the tracker, through the VOT2013 submission page. The results of Level 1 and Level 2 participants (see "Levels of participation"), accompanied with the tracker description, will become part of a co-authored paper which will be published in the ICCV VOT2013 workshop proceedings.

Important dates

~~Evaluation kit available:~~	~~2013/07/10~~
Workshop paper submission	~~2013/09/11~~
(Extended deadline):	**2013/09/14**
Challenge results submission	~~2013/09/11~~
(Extended deadline):	**2013/09/14**
Paper Acceptance:	2013/10/02
Camera Ready Paper:	2013/10/11
Date of the workshop:	2013/12/02

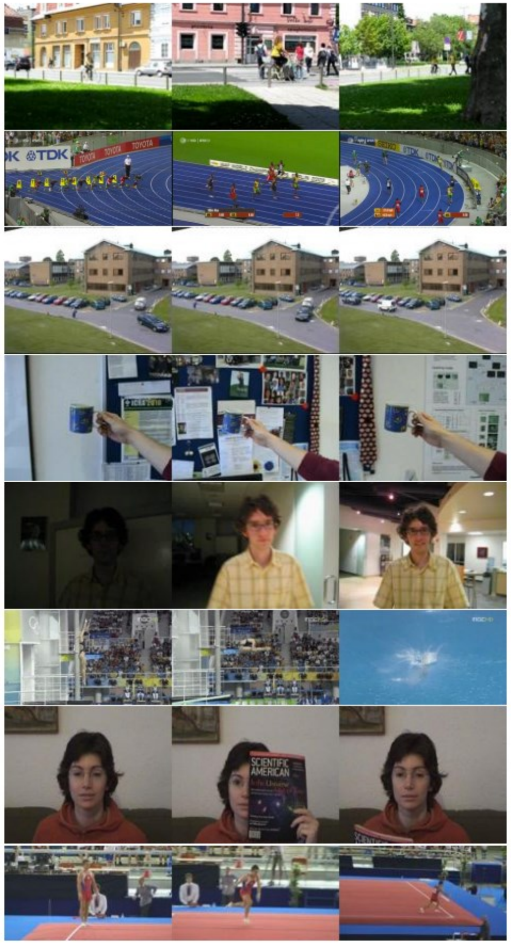
News

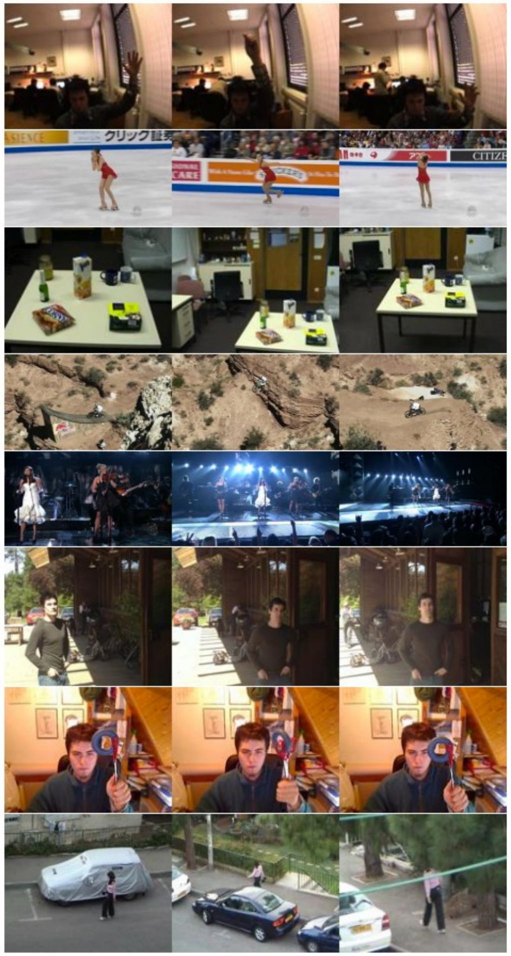
Submission deadline was extended **2013/09/09**

VOT 2013 – the dataset and protocol

- a pool of commonly used sequences annotated by several attributes
- 16 selected semi-automatically

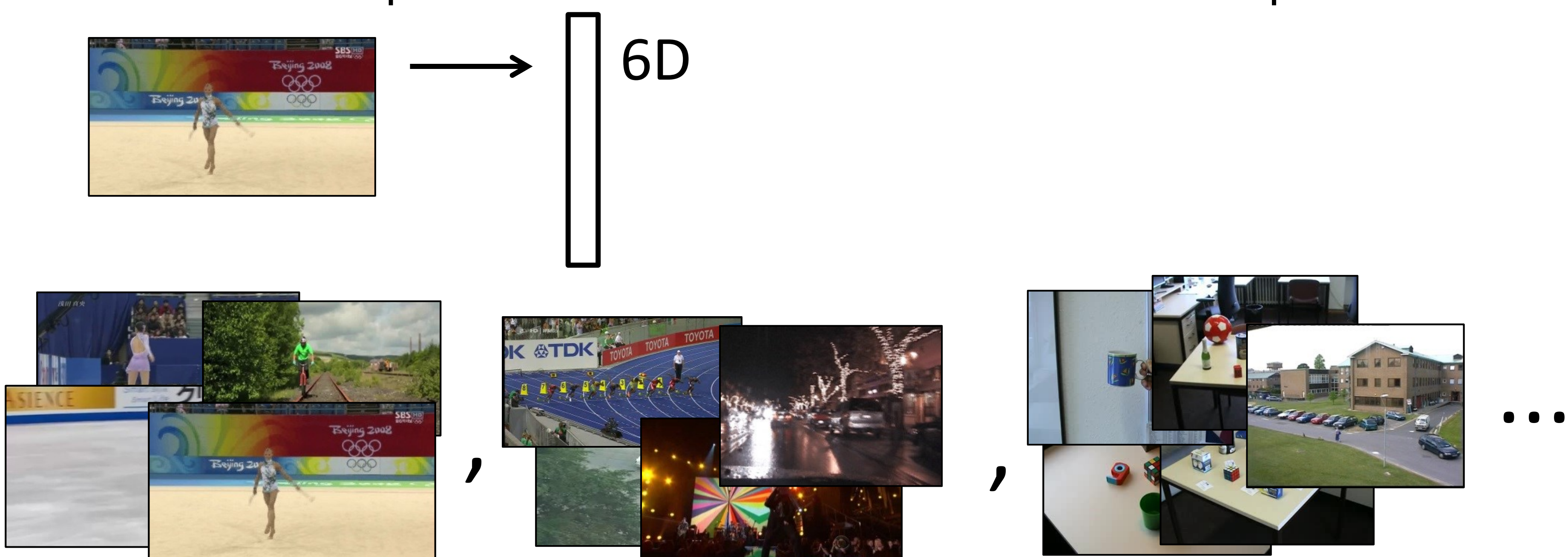
- **Performance measure**: accuracy & robustness
- No common rule for ground-truth bounding box
- Three experiments: baseline, noise, grayscale



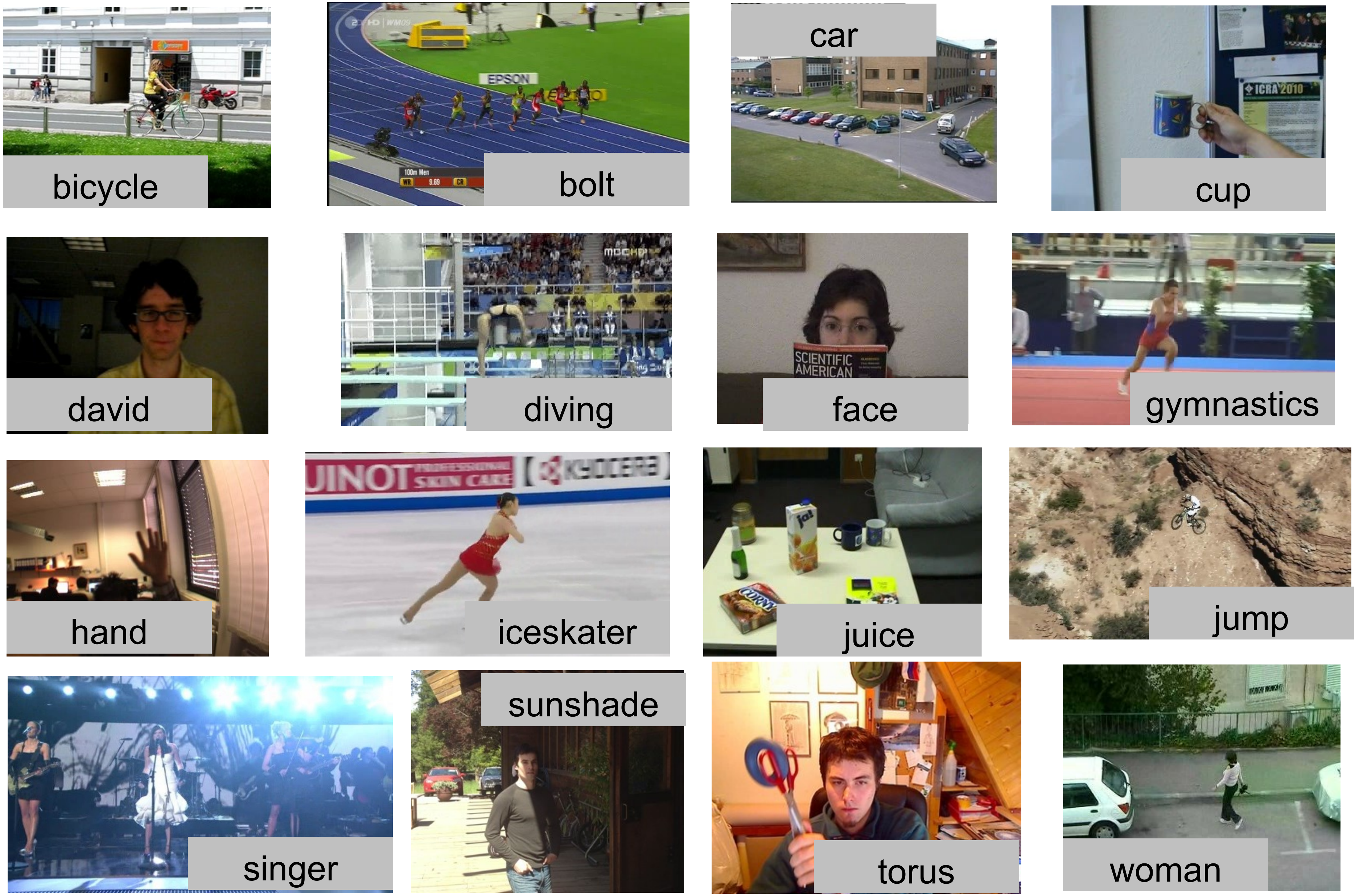


VOT2013 Dataset Construction

- Approach:
  - Include various attributes
  - Keep number of sequences low (Time for performing experiments)
- Collected a pool of ~60 commonly used sequences
- Sequences clustered into 16 clusters by attributes
- A single video selected from each cluster manually.
  - Make sure that phenomena like occlusion were still well represented.



VOT2013 dataset



Sequence ranking based on VOT 2013 results

- Challenging:
  bolt, hand, diving, gymnastics
- Itermediate:
  torus, skater
- Surprise: Less challenging
  David and Singer (overfitting?)
- Easiest: Cup

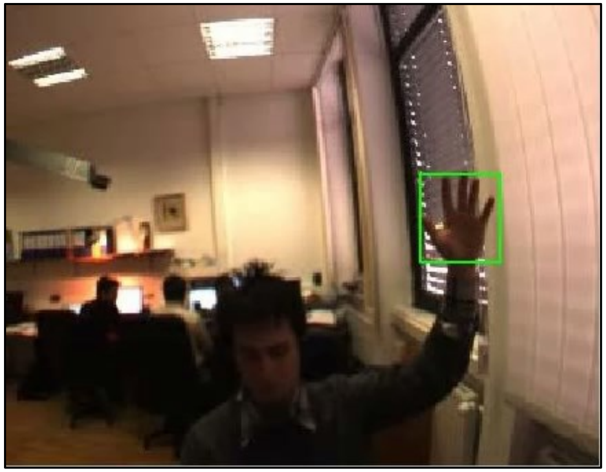
- Locality: a sequence may be challenging only locally

Sequence	Baseline (Av)	Baseline (Max)	Baseline (Frame)
bolt	4,28	13	242
diving	4,23	9	105
hand	4,22	14	51
gymnastics	3,13	12	98
woman	2,86	15	565
sunshade	2,79	11	85
torus	2,67	8	189
iceskater	2,38	6	227
singer	1,68	4	268
david	1,36	4	337
face	1,22	3	140
bicycle	1,22	11	178
juice	1,12	4	242
jump	0,93	4	203
car	0,92	5	253
cup	0,22	2	232

Sequence ranking: Challenging

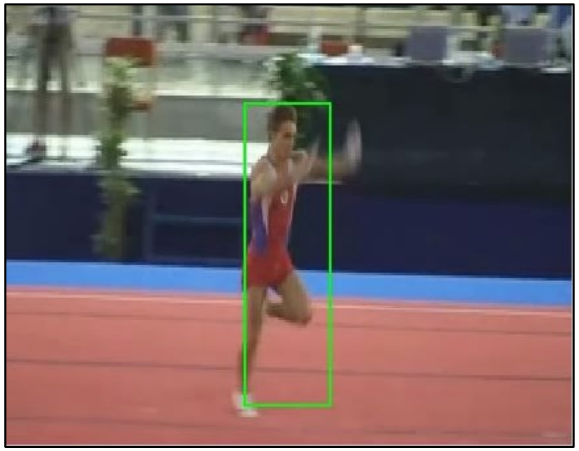
bolt
(camera motion, object motion)

hand
(object motion and size change)



diving (most challenging part)
(camera motion at the end, size change)

gymnastic (most challenging part)
(camera and object motion + size change)



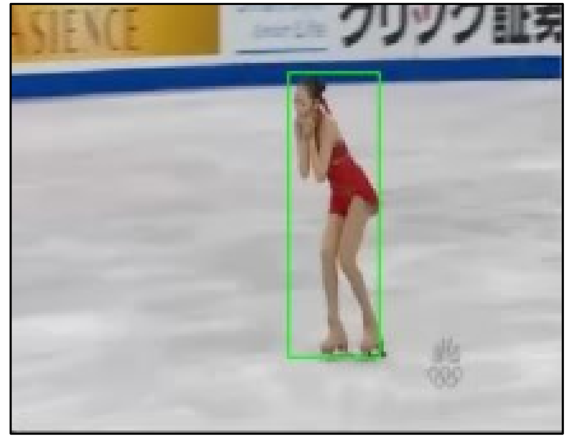
Sequence
bolt
diving
hand
gymnastics
woman
sunshade
torus
iceskater
singer
david
face
bicycle
juice
jump
car
cup

Sequence ranking: Other

- Intermediate (torus, skater)

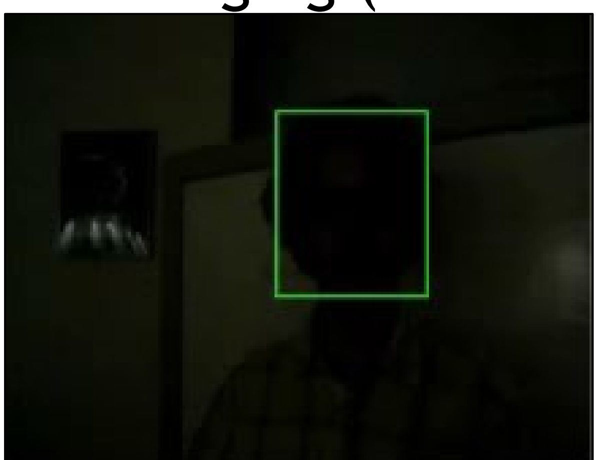
(object motion)

(camera motion, size change)



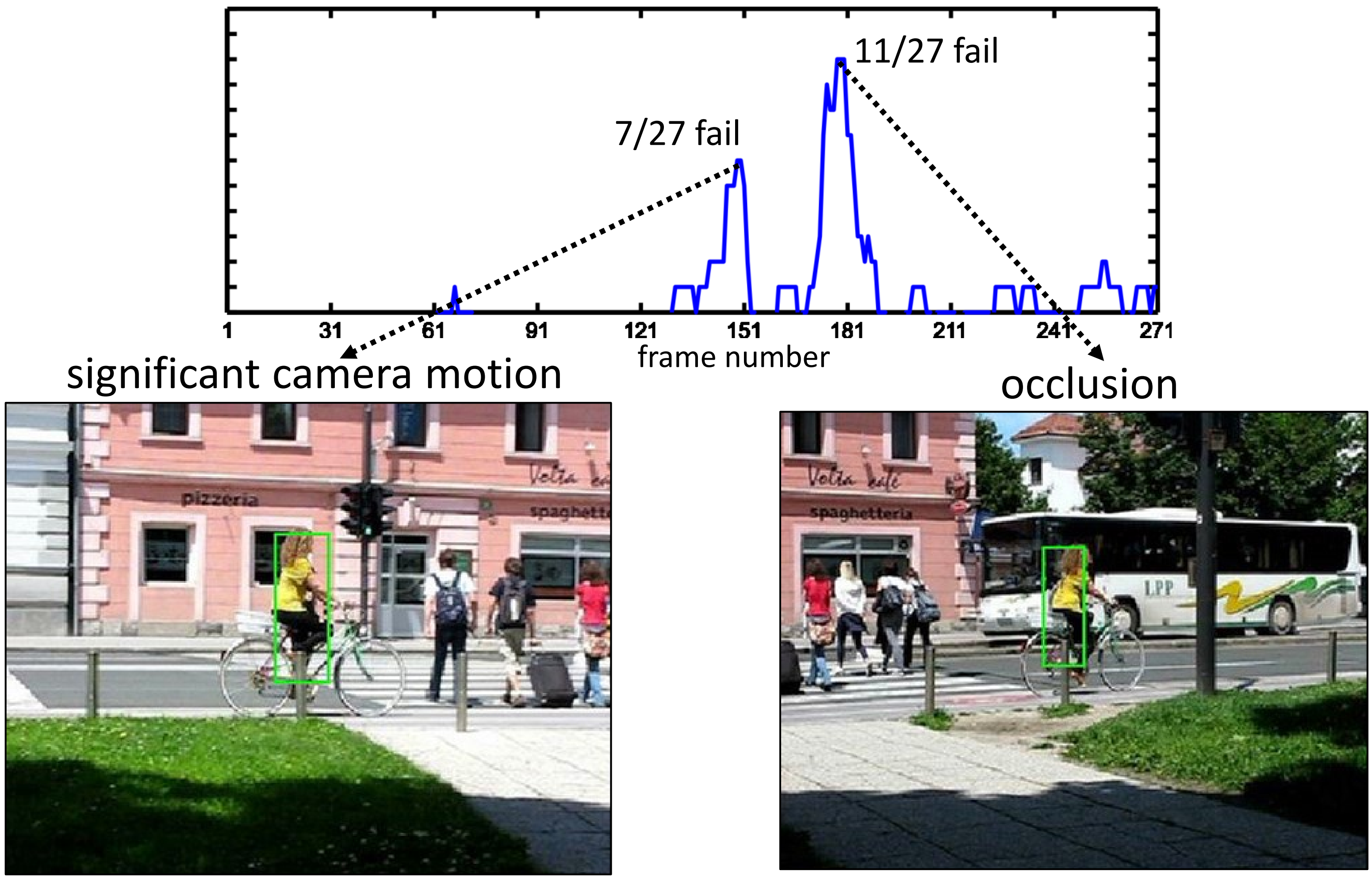
Sequence
bolt
diving
hand
gymnastics
woman
sunshade
torus
iceskater
singer
david
face
bicycle
juice
jump
car
cup

- Less challenging (David and Singer)



Sequence ranking: Locality

- Bicycle: on average not challenging, but very challenging at particular frames where many trackers fail



VOT 2013 dataset

Name	Number of frames	Description
Bicycle	271	bike, occlusion, scale change
Bolt	350	body, articulation, scale change
Car	374	car
Cup	303	cluttered background
David	770	head, illumination and scale change
Diving	231	body, articulated, rotation
Face	415	head, occlusion
Gymnastics	207	body, articulated, scale change
Hand	244	hand, articulated
Iceskater	500	body, articulated
Juice	404	box
Jump	228	bike, scale change
Singer	351	body, illumination and scale change
Sunshade	172	head, illumination change
Torus	264	interesting geometry
Woman	597	body, occlusion, scale change

Class of trackers tested

- Single-object, single-camera
- Short-term causal tracking
- Short-term:
  - Trackers performing without re-detection
- Causality:
  - Tracker is not allowed to use any future frames
- No prior knowledge about the target
  - Only a single training example – BBox in the first frame
- Object state encoded by an axis-aligned bounding box

Submitted trackers. Rough categorization.

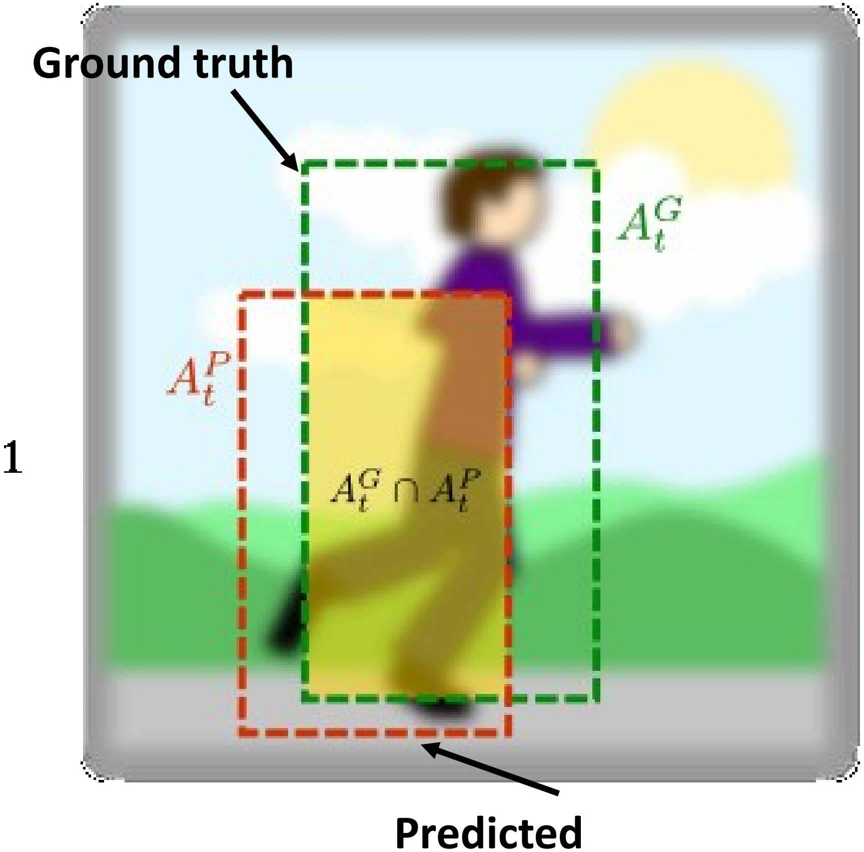
Very diverse set of 27 entries, 19 entries from various authors + 8 baselines contributed by the VOT2013 committee:

- Background-subtraction-based
  (MORP, STMT)
- Optical-flow/motion -based
  (FoT, TLD, SwATrack)
- Key-point-based
  (SCTT, Matrioska)
- Complex appearence-model-based
  (IVT, MS, CCMS, DFT, EDFT, AIF,CactusFl, PJS-S, SwATrack)
- Discriminative models – single part
  (MIL, STRUCK, PLT, CT, RDET, ORIA, ASAM, GSDT)
- Part-based models
  (HT, LGT, LGT++, LT-FLO, TLD)

VOT2013 measures: Accuracy

- Overlap between the ground-truth BB and the BB, predicted by a tracker

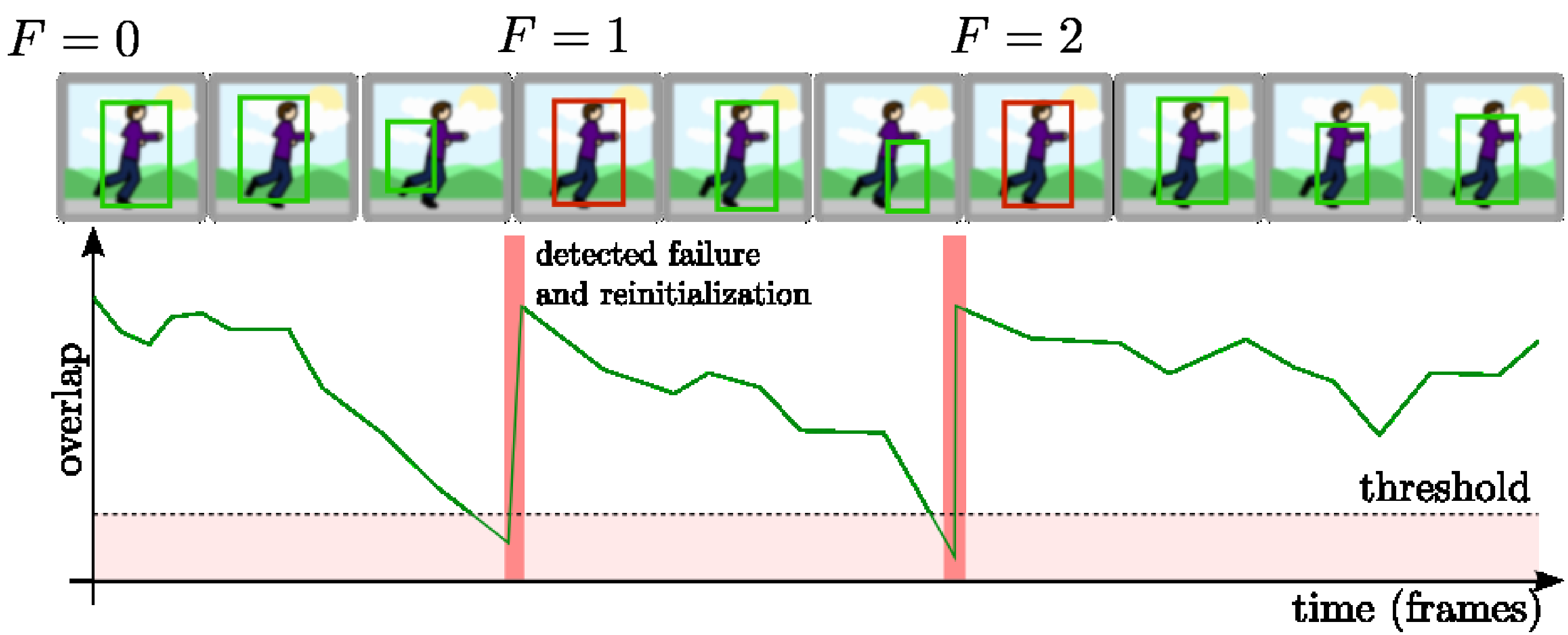
$$\Phi(\Lambda_G, \Lambda_P) = \left\{ \frac{A_t^G \cap A_t^P}{A_t^G \cup A_t^P} \right\}_{t=1}^{N}$$



VOT2013 measures: Robustness

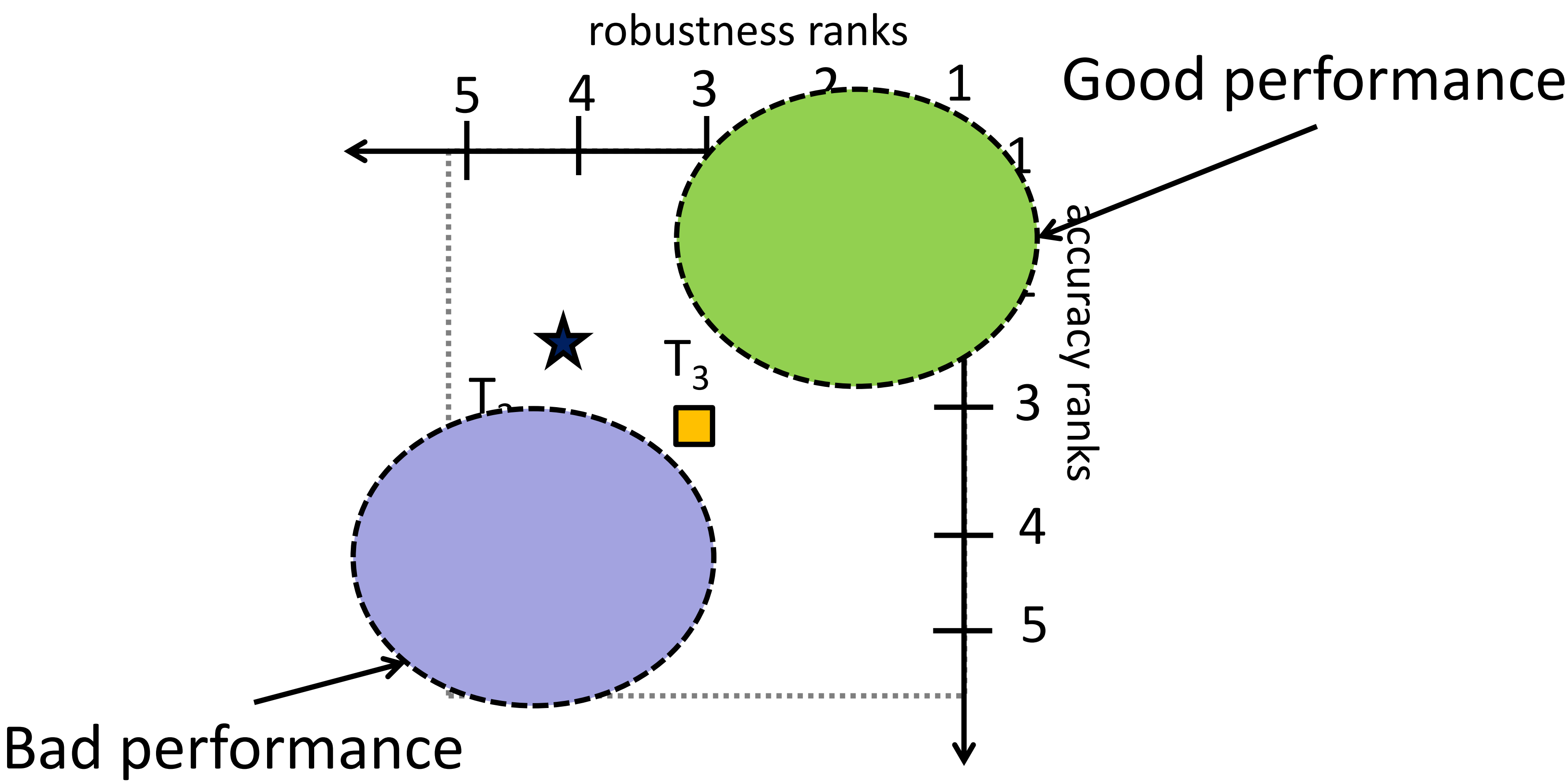
- Counts the number of times the tracker failed and had to be reinitialized
- Failure detected when the overlap drops below a threshold

$$\Phi(\Lambda_G, \Lambda_P)$$



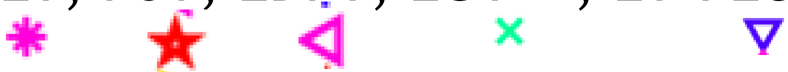
Visualizing the results

- A-R rank plots inspired by [Čehovin et al. 2013]
  - Each tracker is a single point in the rank space



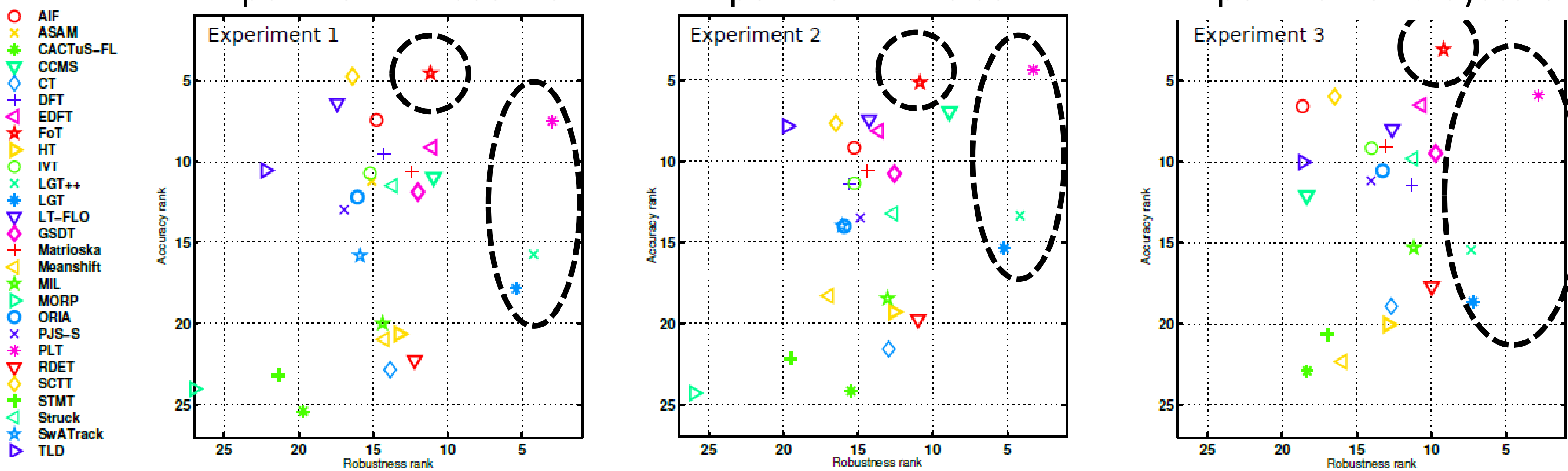
Results: Experiments 1,2,3

- Considering all 3 experiments:
  PLT, FoT, EDFT, LGT++, LT-FLO



	R_Σ
PLT*	4.48
FoT*	7.33
EDFT	[illegible]
LGT++*	10.05
LT-FLO	11.02
GSDT	11.07
	11.29
	11.36
	11.61
	11.68
	11.98
	12.01
	12.25
	12.62
	13.66
	13.92
	14.83
	15.38
	15.48
	16.46
	17.13
Meanshift*	18.12
SwATrack	19.29
STMT	20.63
CACTuS-FL	20.99
ASAM	22.39
MORP	25.89

Experiment1: Baseline Experiment2: Noise Experiment3: Grayscale



Results: Top trackers

- PLT: single-scale, detection-based tracker that applies online structural SVM on color, grayscale and grayscale derivatives.
- FoT: presented in the talk

Tracker	Scale adapt.	Dynamic model	Global vis. mod.	Localization
PLT	no	no	no	determinist.
FoT	yes	no	no	determinist.

	Experiment 1		
	R_A	R_R	R
PLT*	7.51	3.00	5.26
FoT*	4.56	11.15	*7.85*
EDFT*	9.14	11.04	10.09
LGT++*	15.73	*4.25*	9.99
LT-FLO	6.40	17.40	11.90
GSDT	11.87	11.99	11.93
SCTT	*4.75*	16.38	10.56
CCMS*	10.97	10.95	10.96
LGT*	17.83	5.42	11.62
Matrioska	10.62	12.40	11.51
AIF	7.44	14.77	11.11
Struck*	11.49	13.66	12.58
DFT	9.53	14.24	11.89
IVT*	10.72	15.20	12.96
ORIA*	12.19	16.05	14.12
PJS-S	12.98	16.93	14.96
TLD*	10.55	22.21	16.38
MIL*	19.97	14.35	17.16
RDET	22.25	12.22	17.23
HT*	20.62	13.27	16.95
CT*	22.83	13.86	18.35
Meanshift*	20.95	14.23	17.59
SwATrack	15.81	15.88	15.84
STMT	23.17	21.31	22.24
CACTuS-FL	25.39	19.67	22.53
ASAM	11.23	15.09	13.16
MORP	24.03	27.00	25.51

Tracking speed

- PLT (C++) ~169fps
- FoT (C++) ~156fps

	FPS	Implem.	Hardware
PLT		C++	Intel Xeon E5-16200
FoT		C++	Intel i7-3770
EDFT	12.82	Matlab	Intel Xeon X5675
LGT++	5.51	Matlab / C++	Intel i7-960
LT-FLO	4.10	Matlab / C++	Intel i7-2600
GSDT	1.66	Matlab	Intel i7-2600
SCTT	1.40	Matlab	Intel i5-760
CCMS		Matlab	Intel i7-3770
LGT	2.25	Matlab / C++	AMD Opteron 6238
Matrioska	16.50	C++	Intel i7-920
AIF	30.64	C++	Intel i7-3770
Struck	3.46	C++	Intel Pentium 4
DFT	6.65	Matlab	Intel Xeon X5675
IVT	5.03	Matlab	AMD Opteron 6238
ORIA	1.94	Matlab	Intel Pentium 4
PJS-S	1.18	Matlab / C++	Intel i7-3770K
TLD	10.61	Matlab	Intel Xeon W3503
MIL	4.45	C++	AMD Opteron 6238
RDET	22.50	Matlab	Intel i7-920
HT	4.03	C++	Intel i7-970
CT	9.15	Matlab / C++	Intel Pentium 4
Meanshift	8.76	Matlab	Intel Xeon
SwATrack	2.31	C++	Intel i7
STMT	0.24	C++	Intel Xeon X7460
CACTuS-FL	0.72	Matlab	Intel Xeon X5677
ASAM	0.93	Matlab	Intel i5-2400
MORP	9.88	Matlab	Intel i7

The Visual Object Tracking Challenge @ ECCV

In conjunction with ECCV 2014

VOT2014

workshop on visual object tracking challenge

September, 6th 2014 - Zurich, Switzerland

Short-term Tracking

September 6

Home | News | Participation | Dataset | Submission | People | Program | Supporters | VOT Challenge

Call for participation and for papers

The VOT challenges provide the tracking community with a precisely defined and repeatable way of comparing short-term trackers as well as a common platform for discussing the evaluation and advancements made in the field of visual tracking.

The first challenge - (VOT2013) - introduced a new evaluation kit (software plus 16 well-known short videos) and compared 27 single-target trackers submitted by 51 participants. The results were published in a joint paper presented at an ICCV2013 workshop which was attended by over 70 researchers.

This year, the VOT committee is organizing the VOT2014 challenge in conjunction with the ECCV2014 and invites researchers from academia and industry to participate. Similarly to VOT2013, the challenge aims at **single-object short-term trackers** that do not apply pre-learned models of object appearance (**model-free**). Trackers do not necessarily need to be capable of automatic re-initialization, as the objects are visible over the whole course of the sequences.

The VOT2014 committee solicits VOT2014 challenge tracking results of old and new trackers as well as full-length papers. The scope of full-length papers are papers describing original trackers as well as papers describing improvements of existing trackers and papers giving new insights into existing trackers or classes of trackers.

Important dates

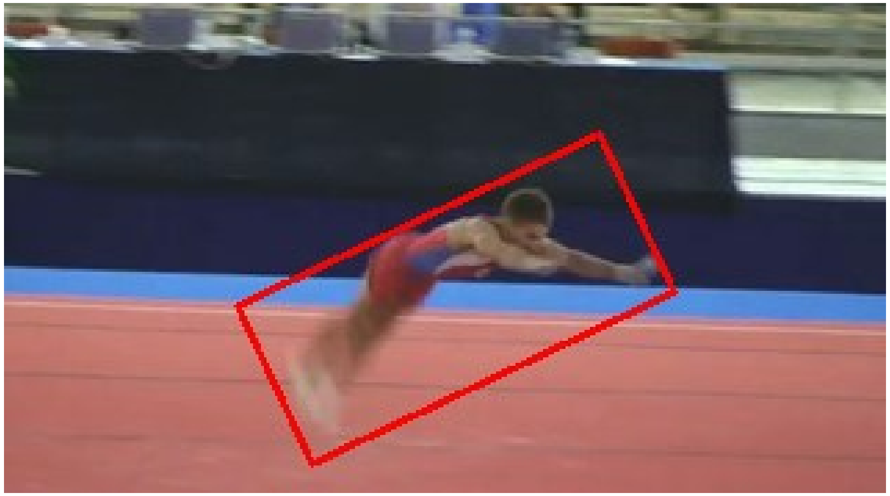
Start of challenge	19 May
~~Paper submission~~	~~7 July~~
~~Results submission~~	~~11 July~~
~~Notification of acceptance~~	~~18 July~~
~~Camera ready paper~~	~~25 July~~
~~Early registration~~	~~31 July~~
Workshop	**6 Sept**

See our latest news on the challenge

Ask questions?

VOT2014 highlights:

- An improved version of the cross-platform evaluation kit, which will execute the experiments much faster thanks to a powerful new communication protocol between kit and tracker
- The dataset is enriched with new videos (in total 25 sequences) and labelled with rotating bounding boxes rather than axis-aligned ones
- The dataset is per-frame labelled with attributes



Rotated B-Boxes – Interpretation?

Rotated B-Boxes – Interpretation?

Rotated B-Boxes – Interpretation?

VOT2014 results

- **total of 38 trackers with binaries/source code submitted**

- "Winning tracker is in <u>average rank</u> 8^{th} best performing tracker on sequence"

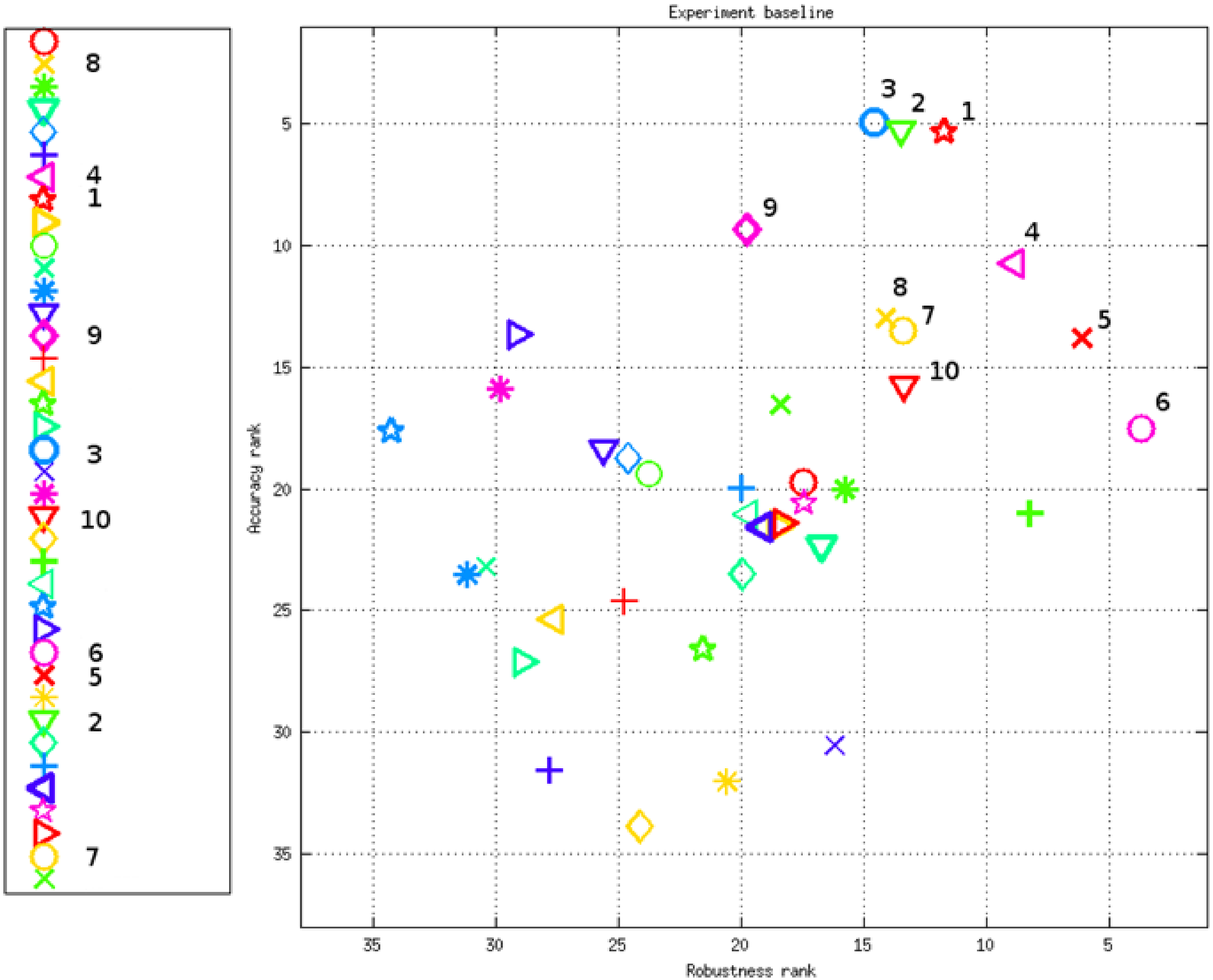
  → lot of space for improvement

- is one benchmark with known ground truth – what about overfitting?

  → several surprise evaluation (one of them is on the VOT2013 benchmark)

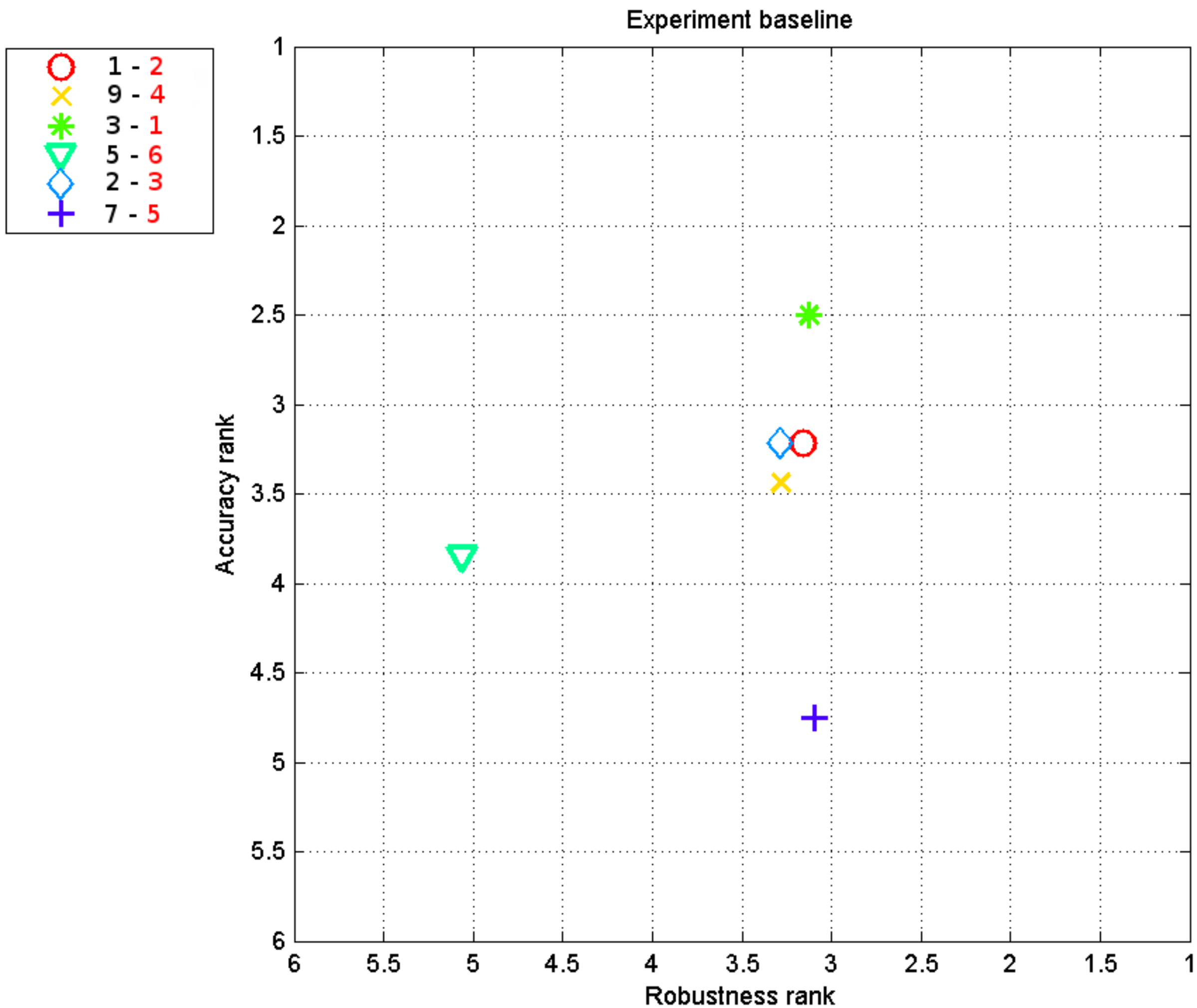
Best 10 methods	**baseline exp. rank**
1.	8.5
2.	9.4
3.	9.8
4.	9.8
5.	9.9
6.	10.6
7.	13.4
8.	13.5
9.	14.5
10.	14.6

VOT2014 results – AR plot



VOT2014 surprise evaluation

- several top performing trackers compared on VOT2013 benchmark

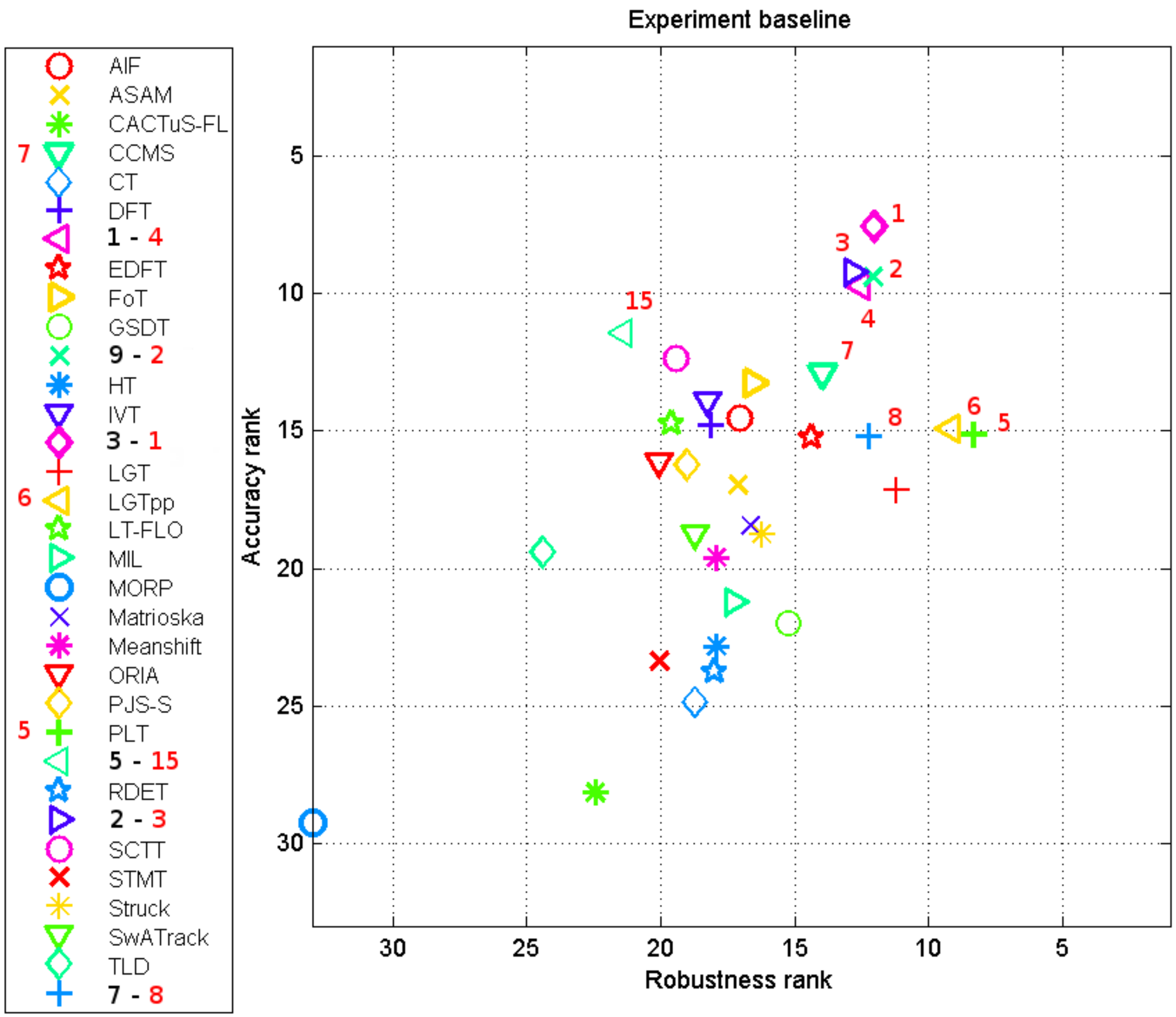


black number – rank of the tracker in the VOT2014 baseline experiment

red number – rank in the conducted experiment

VOT2014 surprise evaluation

- several top performing trackers compared with VOT2013 benchmark trackers

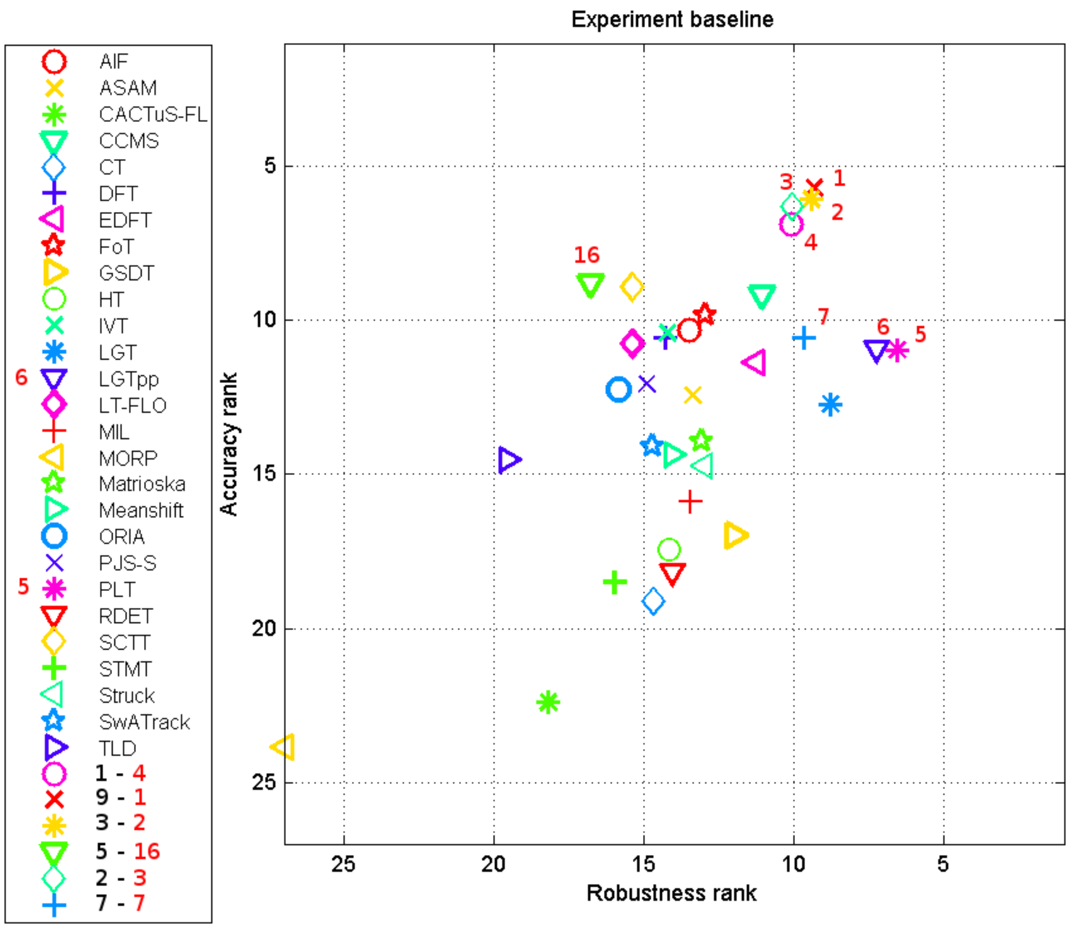


black number – rank of the tracker in the VOT2014 baseline experiment

red number – rank in the conducted experiment

VOT2013 surprise evaluation – on benchmark

- several top performing trackers compared on VOT2013 benchmark (how new trackers should be compared to VOT benchmarks, i.e. rank of trackers of VOT2013 is not changed, rank for new trackers is computed based on their standings against the trackers in



black number – rank of the tracker in the VOT2014 baseline experiment

red number – rank in the conducted experiment

VOT 2014 - Analysis

The winners

- do not estimate rotation
- do not use a dynamic model
- do not use model adaptation

- are performing tracking by detection ....

To me, the results are somewhat counter-intuitive:

- the best trackers do not estimate parameters of the object motion!
- some trackers are overfitting

VOT 2015 – any comments on benchmark improvements welcome.

VOT2015 Challenge

The VOT challenges provide the tracking community with a precisely defined and repeatable way of comparing short-term trackers as well as a common platform for discussing the evaluation and advancements made in the field of visual tracking.

Following two highly successful VOT challenges VOT2013 (ICCV 2013) and VOT2014 (ECCV2014), we are happy to announce the 3rd Visual Object Tracking Challenge VOT2015 to be held in conjunction with the ICCV2015. Researchers from industry as well as academia are invited to participate. Similarly to VOT2013 and VOT2014, the challenge aims at single-object short-term trackers that do not apply pre-learned models of object appearance (model-free). Trackers do not necessarily need to be capable of automatic re-initialization, as the objects are visible over the whole course of the sequences.

We are also announcing the first VOT thermal imagery tracking sub-challenge VOT-TIR2015. For convenience, the submission of the VOT2015 and the VOT-TIR2015 challenge is via a common submission system. The results of the VOT2015 and VOT-TIR2015 challenges will be presented at the ICCV2015 VOT workshop.

Call for participation and for papers

The VOT committee invites you to participate by:

1. Entering one or both of the following challenges:
   - **VOT2015 challenge** - Visit the participation page on running the VOT2015 experiments and submitting the results.
   - **VOT-TIR2015 challenge** - Visit the participation page on running the VOT-TIR2015 experiments and submitting the results.

1. Submitting a full-length paper describing:
   - Original or improved trackers as well as papers giving new insights into existing trackers or class of trackers. See

VOT15

visual object tracking challenge

Summary

- “Visual Tracking” may refer to quite different problems.

- Be careful when evaluating tracking results

- 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 where and tracking, 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?