Fast Learnable Object Tracking and Detection in High-resolution Omnidirectional Images

David Hurych, Karel Zimmermann, Tomáš Svoboda

Center for Machine Perception

Department of Cybernetics

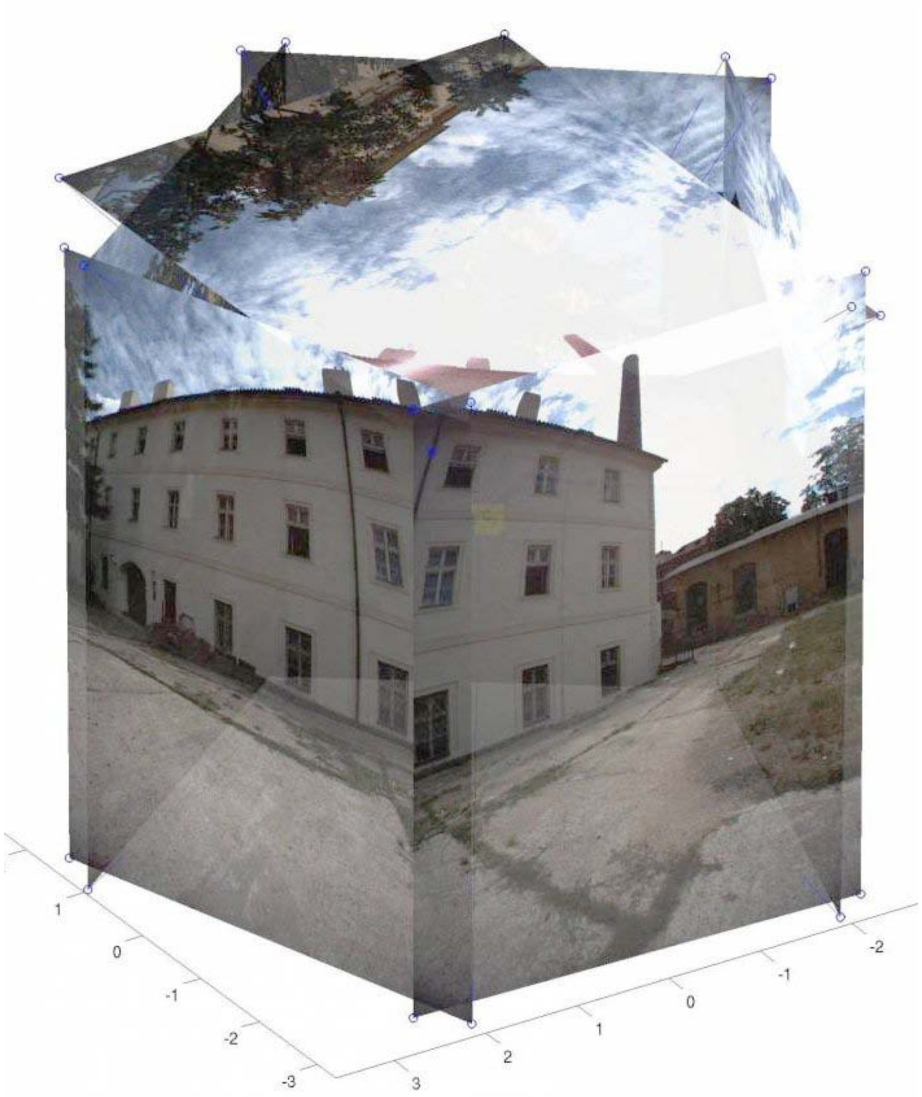
Faculty of Electrical Engineering

Czech Technical University in Prague

Motivation

- Visual subsystem of a rescue robot for EU project NIFTi
  - Victim detection
  - Localization of dangerous materials
  - Building a 3D map of the crash site

Ladybug 3



Example images

Cylindrical projection of six 2 MPix images

to one 12 MPix image

Frame No: 0195

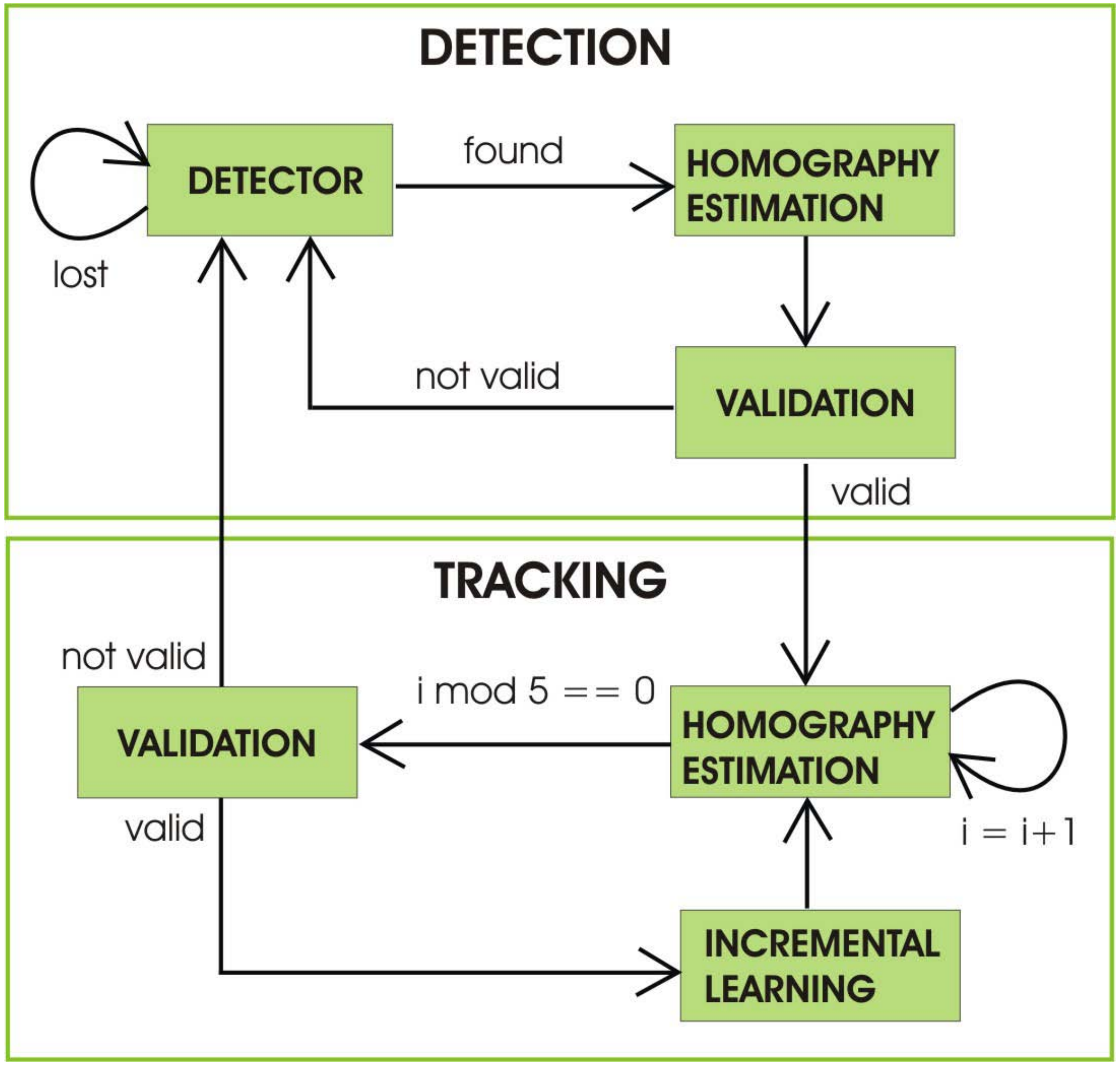
Application

- GOAL
- Real-time (close to) object detection and tracking in 5 fps sequence of large resolution images

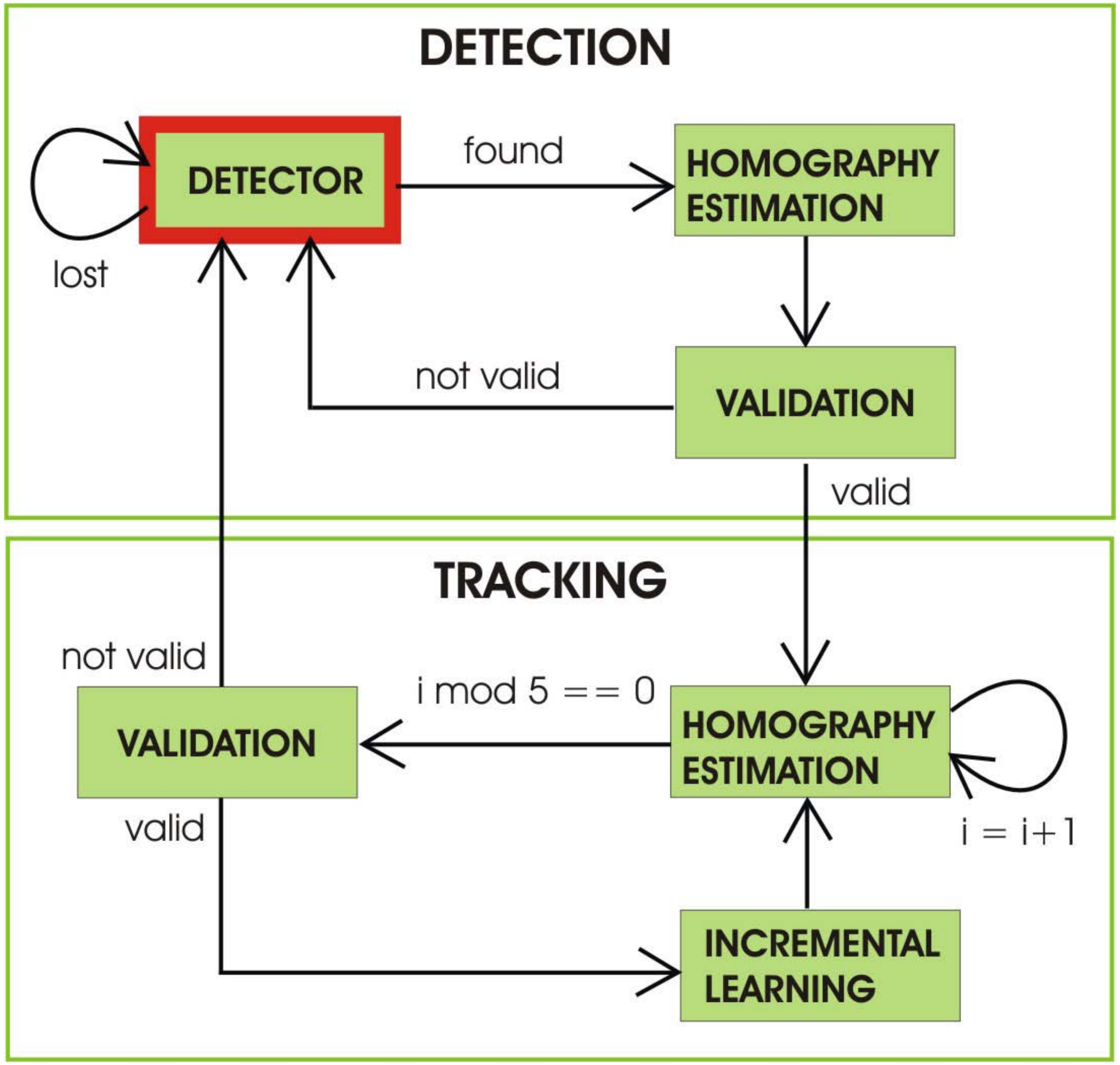
- SOLUTION
- Combine fast object detector with fast tracker
- Take advantage of learning-based methods
  - Improves tracking speed and more robust to deformation

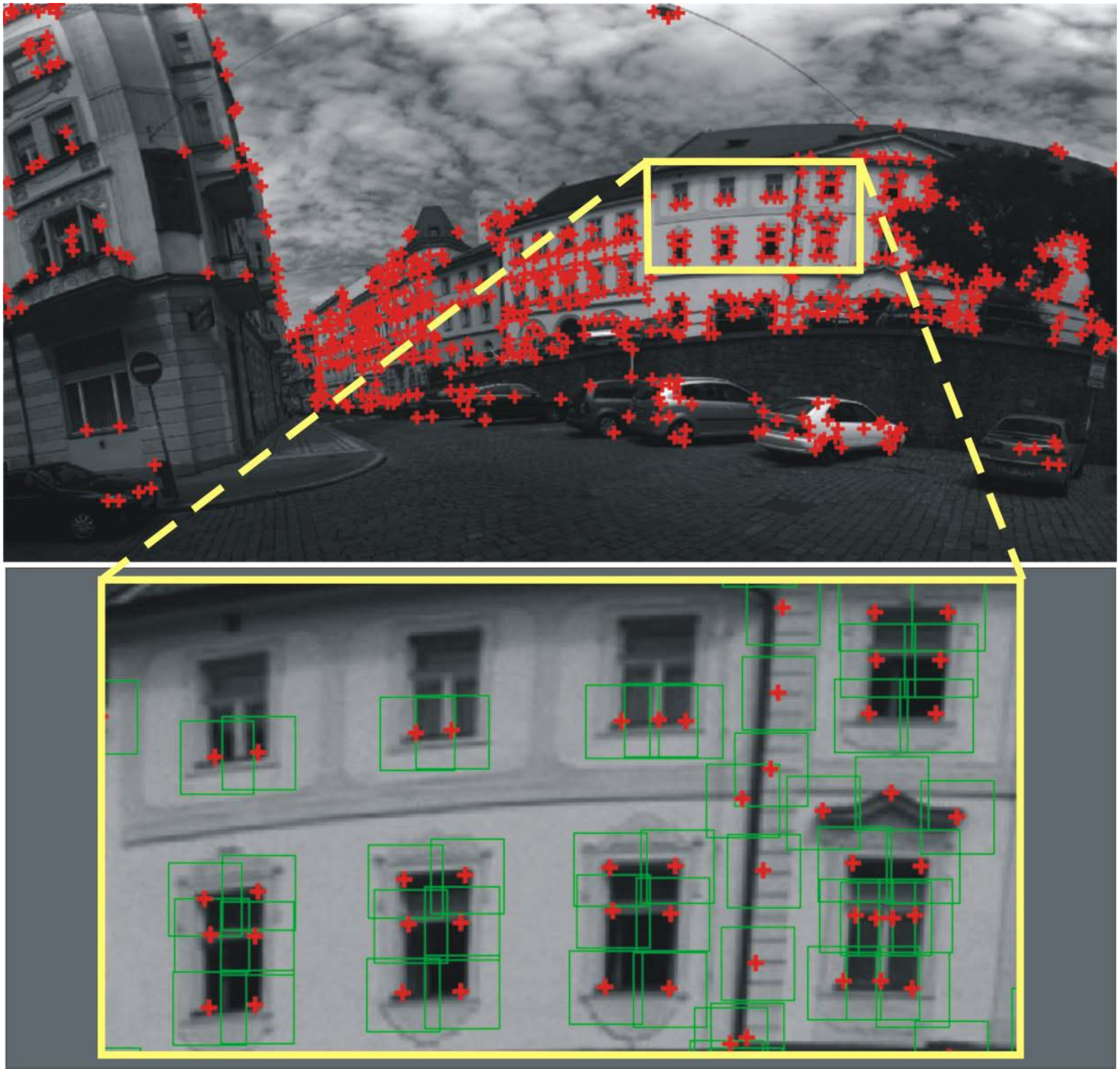
    (Ozuysal et al. 2010, Hinterstoisser et al. 2008)

Proposed Algorithm



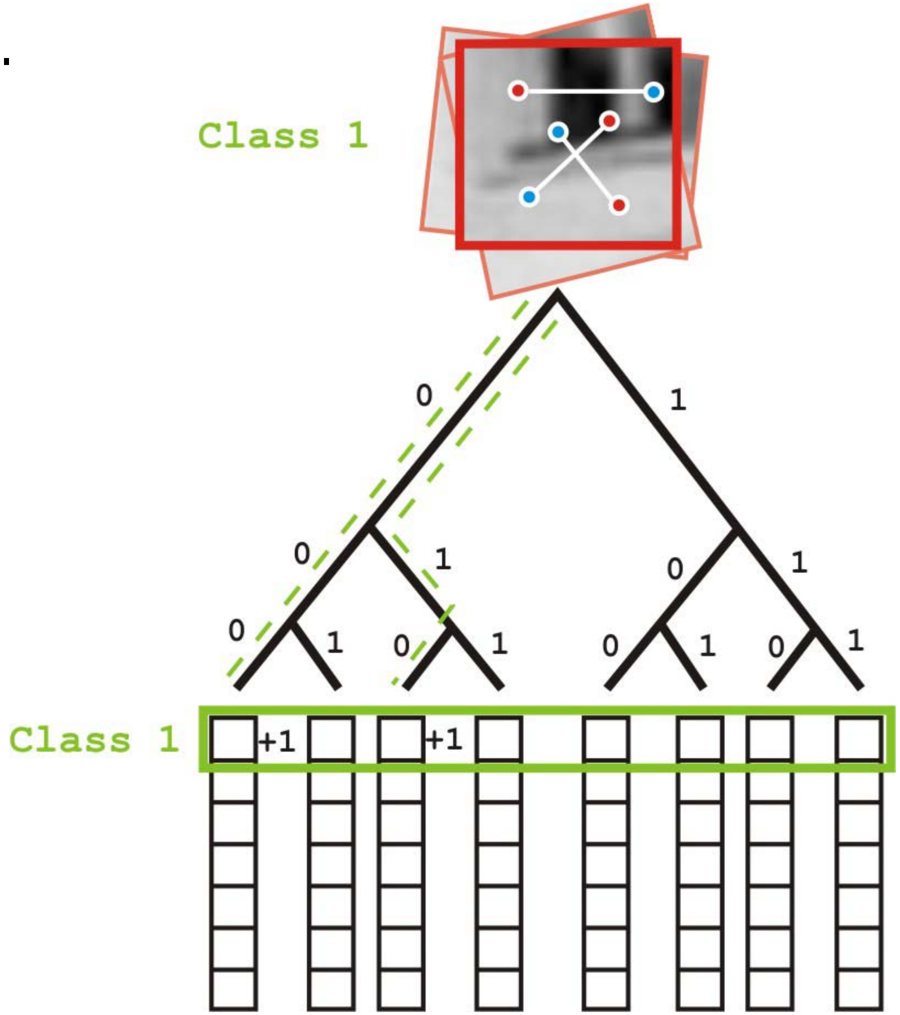
Proposed Algorithm



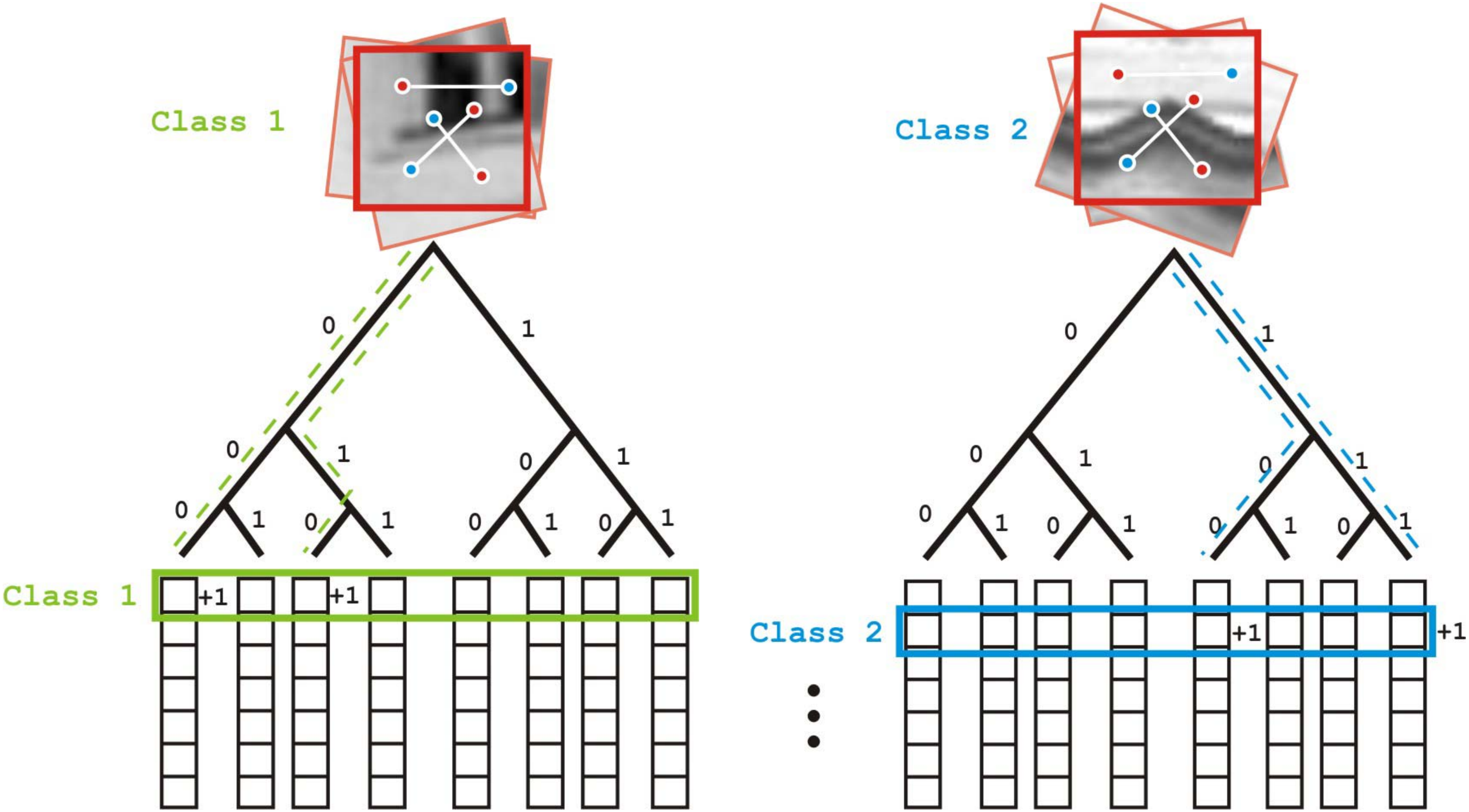


Ferns detector

- Proposed by Ozuysal et al.
  Fast keypoint recognition using random ferns,
  in PAMI 2010

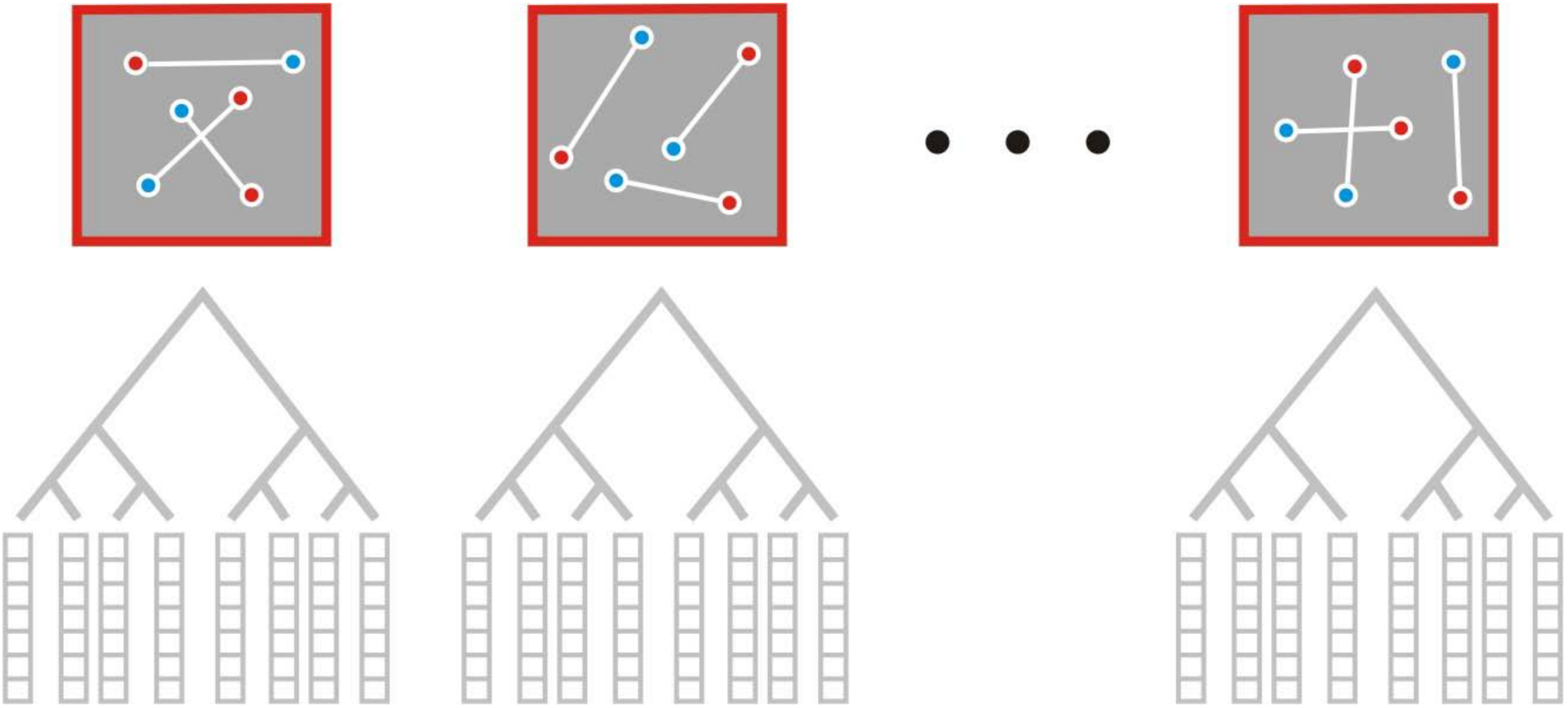


Ferns – multiple classes

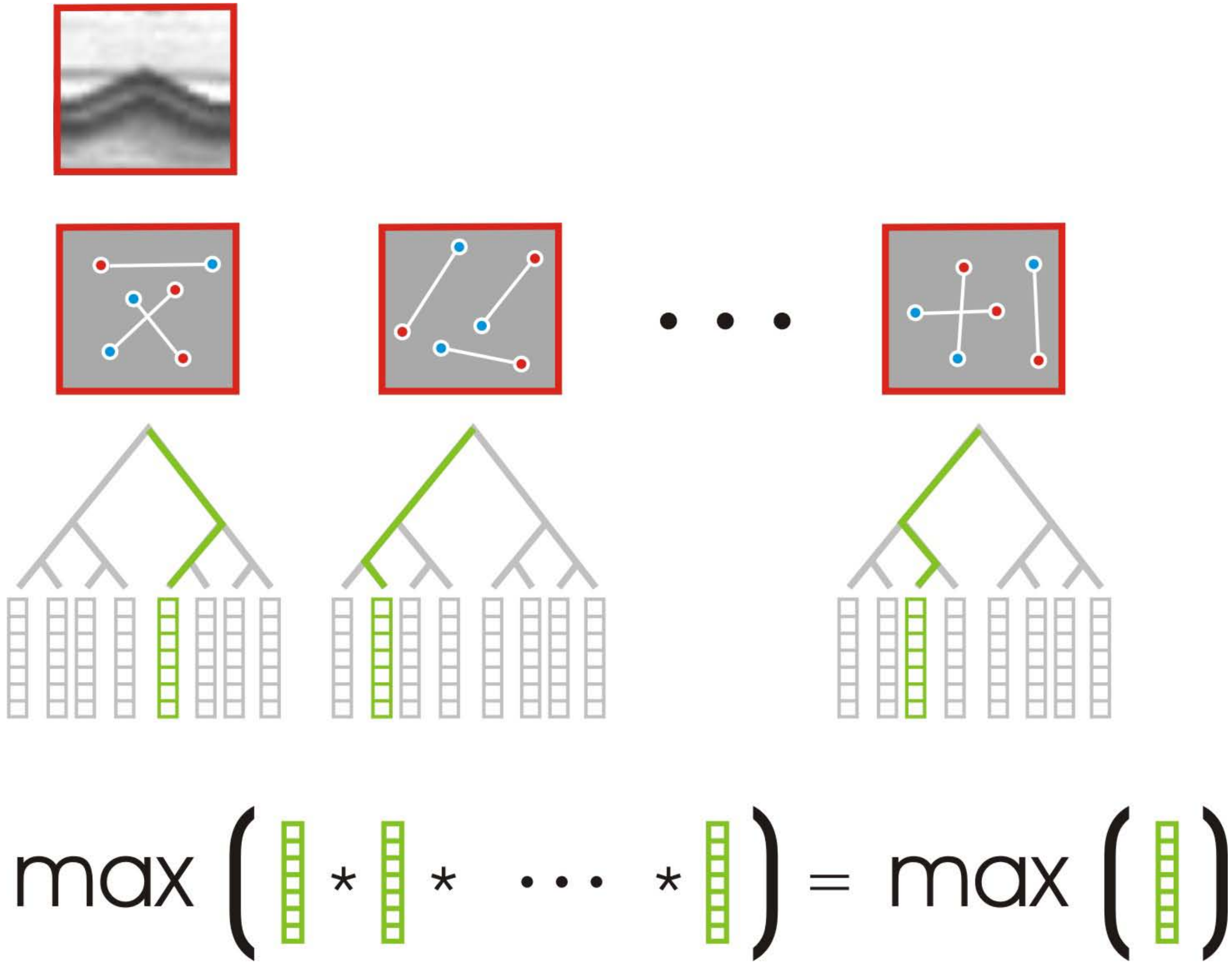


Ferns – multiple sets

Sets of binary tests are considered to be independent



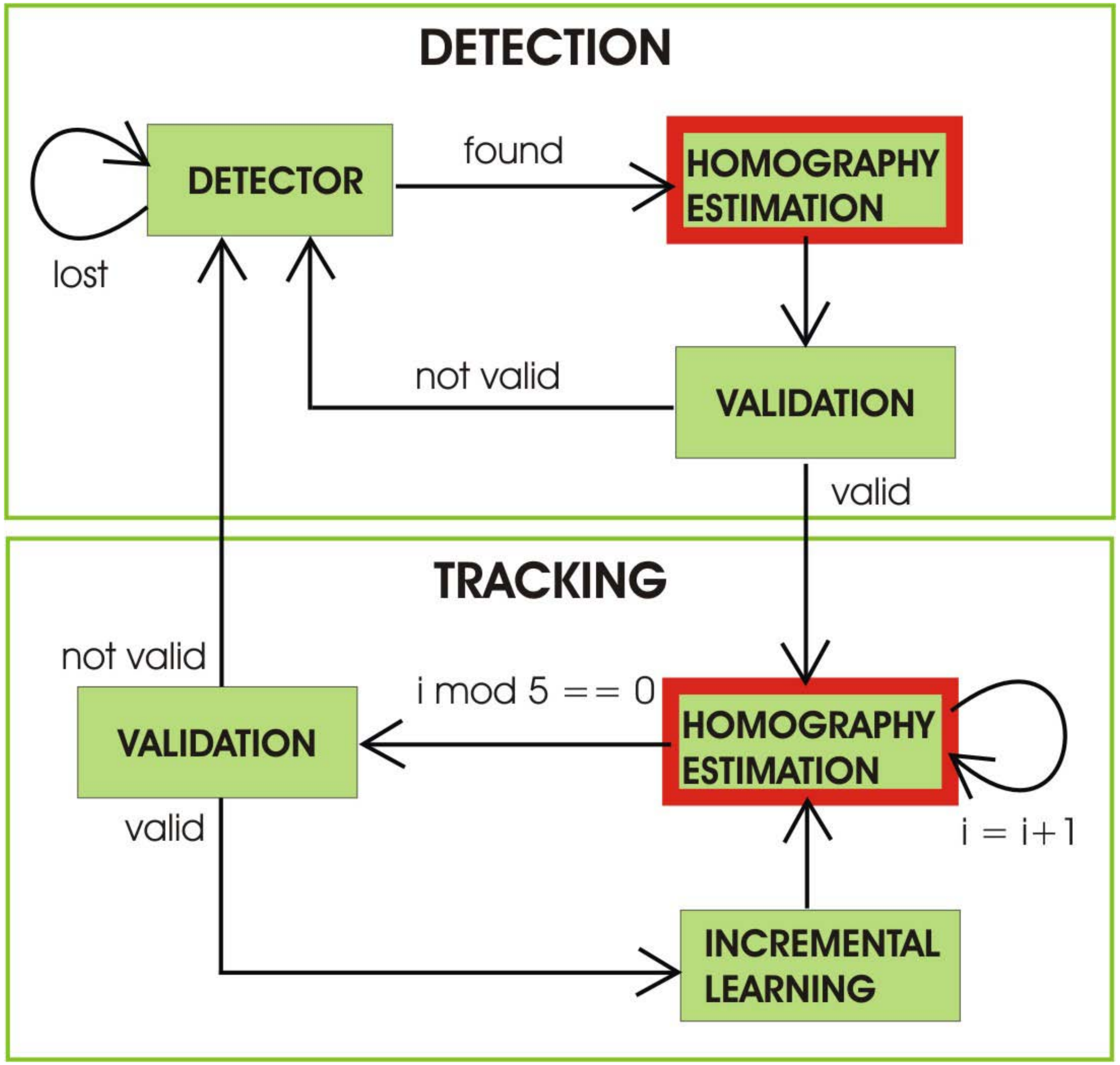
Ferns Detector - detection



RANSAC

- All keypoints are assigned to some class
- We run RANSAC over all detected keypoints and determine the affine transformation
- Detection confidence = number of RANSAC inliers

Proposed Algorithm



Sequential Learnable Linear Predictor (SLLiP)

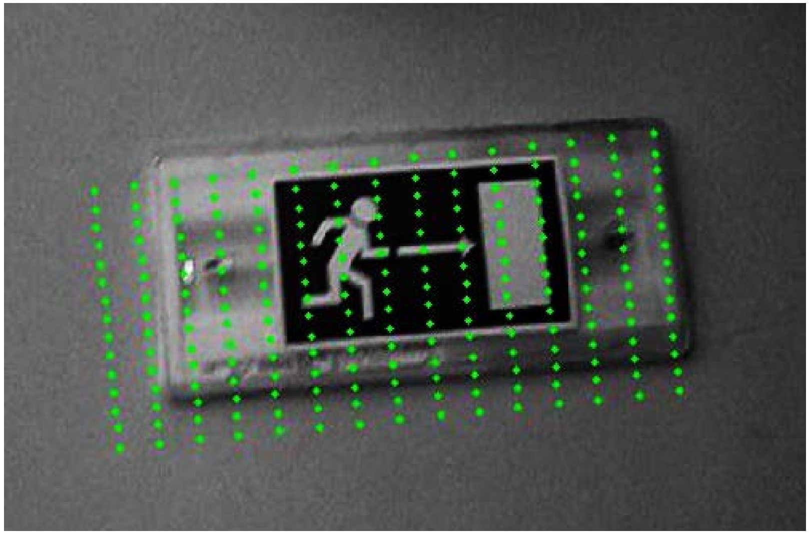
Predictor tracking

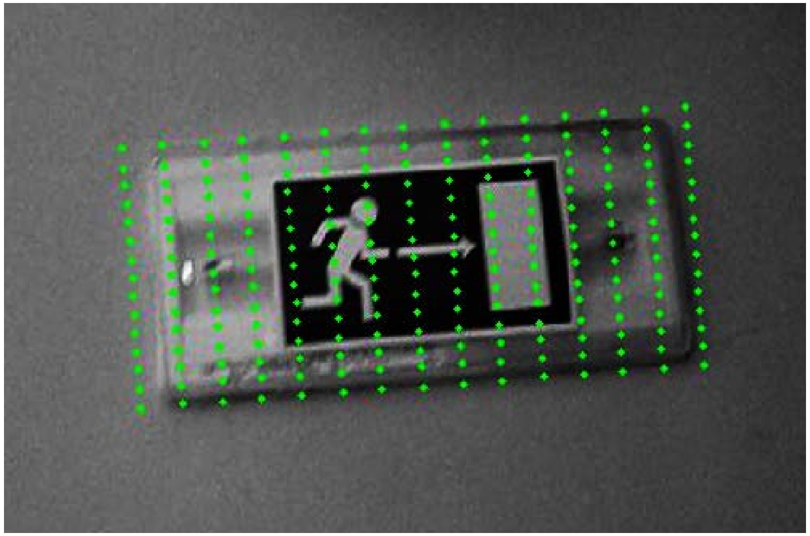
$$\mathbf{t} = \mathtt{H} I\left(X\right)$$

Zimmermann – PAMI 2009

SLLiP tracking

$$\begin{aligned}
\mathbf{t}_1 &= \mathtt{H}_1 I\left(X_1\right) \\
\mathbf{t}_2 &= \mathtt{H}_2 I\left(\mathbf{t}_1 \circ X_2\right) \\
\mathbf{t}_3 &= \mathtt{H}_3 I\left(\mathbf{t}_2 \circ X_3\right) \\
&\vdots \\
\mathbf{t} &= \bigcirc_{(i=1,\dots,k)} \mathbf{t}_i
\end{aligned}$$



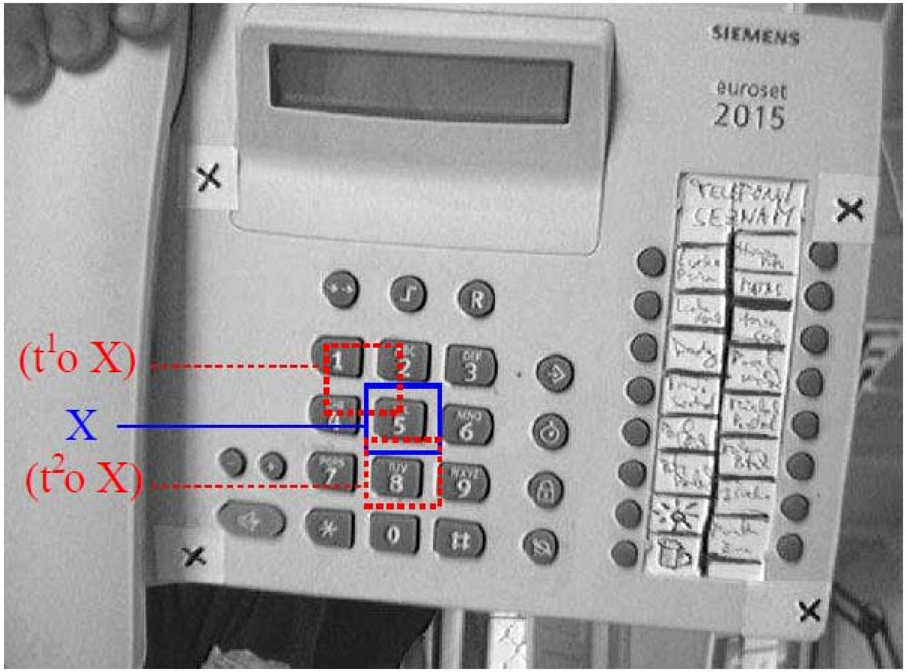


Linear Predictor Learning

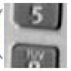
Least squares learning

$$\mathrm{H}^* = \underset{\mathrm{H}}{\operatorname{argmin}} \|\mathrm{HD} - \mathrm{T}\|_F^2$$

$$\mathrm{H}^* = \mathrm{TD}^T \left(\mathrm{DD}^T\right)^{-1}$$



$\theta($  $) = (0,0)^\top$ $\theta($  $) = (25,25)^\top$

$\theta($  $) = (0,15)^\top$ $\theta($  $) = (-15,0)^\top$

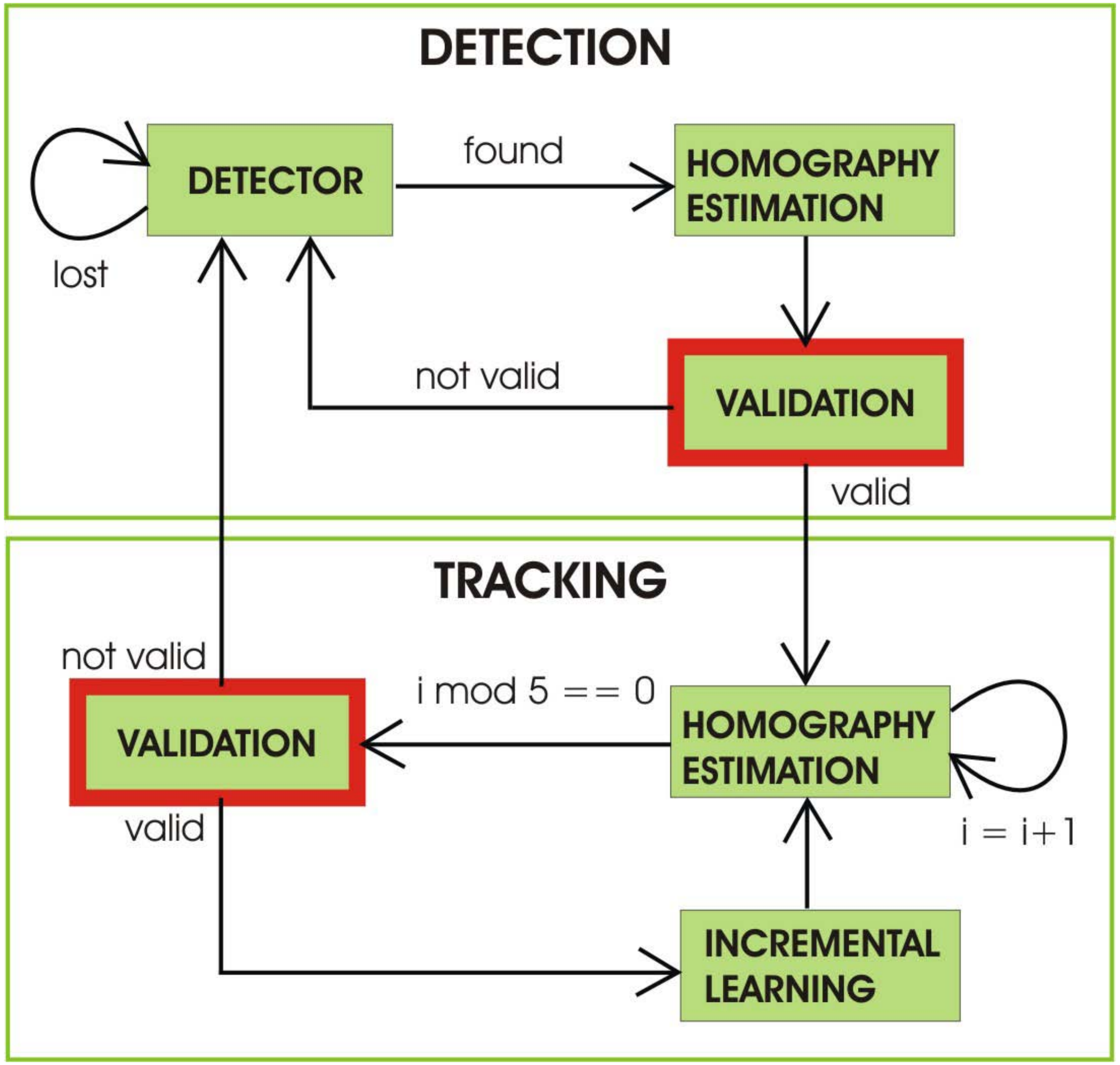
SLLiP tracking model

$$\theta_s = \left|\{\mathrm{H}_1, X_1\}, \{\mathrm{H}_2, X_2\}, \ldots, \{\mathrm{H}_k, X_k\}\right|$$

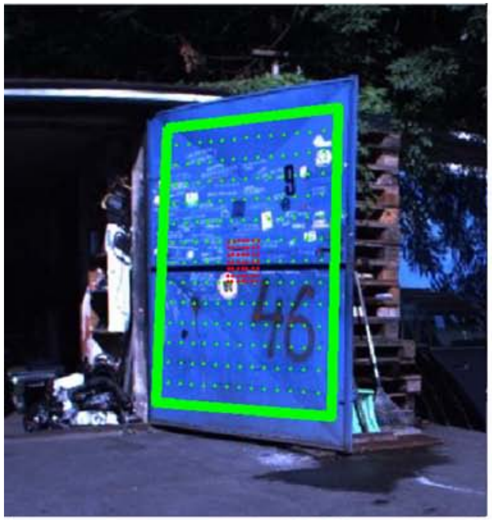
Modeled Transformation

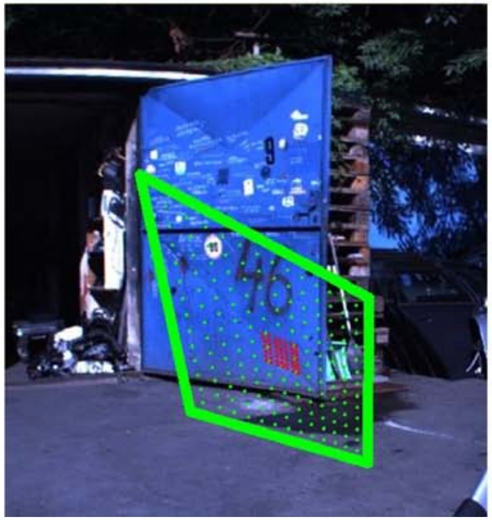
- Ferns detector + RANSAC
  - Affine transformation
- 2-SLLiP tracker
  - 2D translation
  - Homography - parameterized by position of 4 patch corners

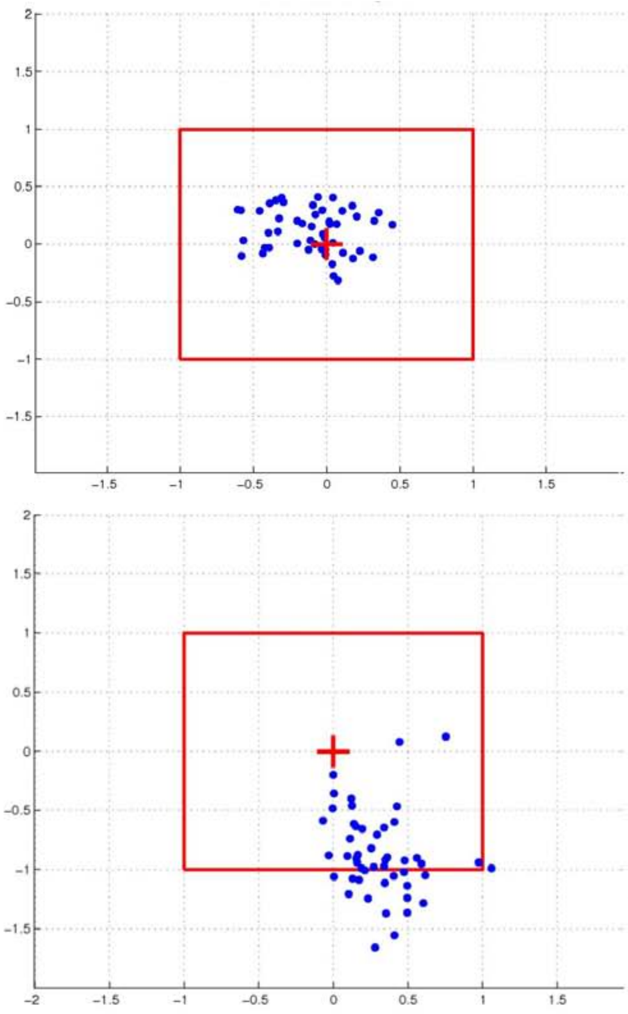
Proposed Algorithm

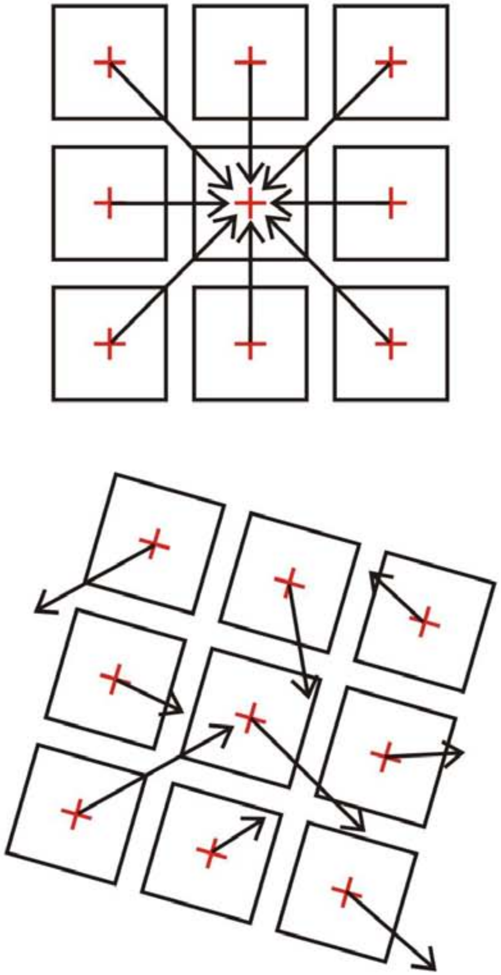


Tracking Validation

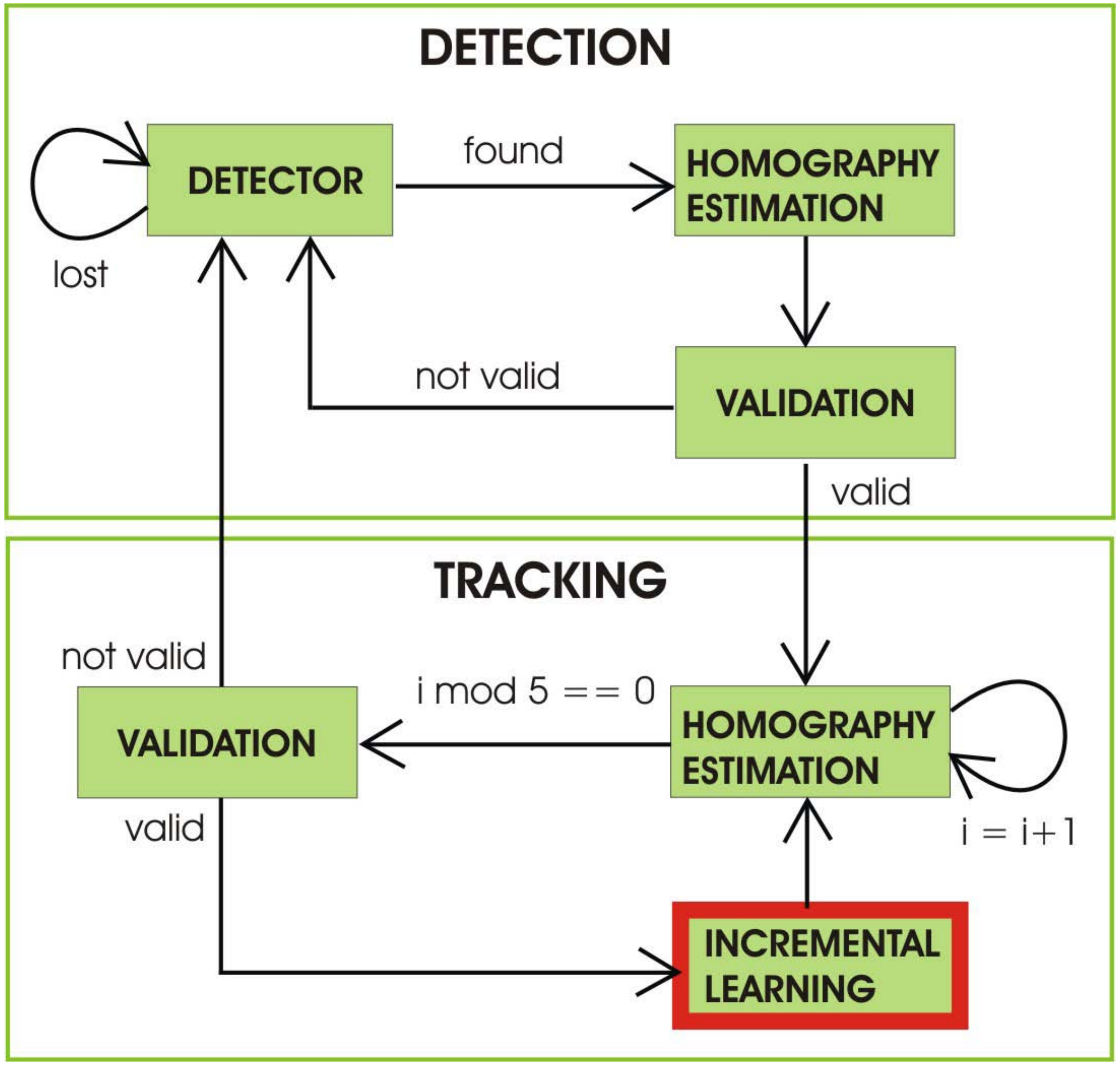








Proposed Algorithm



Incremental Learning

update of matrices $\mathtt{H}_i$ during tracking

- proposed by Hinterstoisser et al. in

  Online learning of patch perspective rectification for efficient object detection. In CVPR 2008.

$$\cancel{\mathtt{H}^* = \mathtt{T}\mathtt{D}^T \left(\mathtt{D}\mathtt{D}^T\right)^{-1}}$$

$$\mathtt{H}_i = \mathtt{Y}_i \mathtt{Z}_i$$

$$\mathtt{Y}_i = \mathtt{T}_i \mathtt{D}_i^T$$

$$\mathtt{Z}_i = \left(\mathtt{D}_i \mathtt{D}_i^T\right)^{-1}$$

$$\mathbf{d} = I\left(X\right)$$

$$\mathtt{Y}_i^{j+1} = \mathtt{Y}_i^j + \mathbf{t}\mathbf{d}^T$$

$$\mathtt{Z}_i^{j+1} = \mathtt{Z}_i^j - \frac{\mathtt{Z}_i^j \mathbf{d}\mathbf{d}^T \mathtt{Z}_i^j}{1 + \mathbf{d}^T \mathtt{Z}_i^j \mathbf{d}}$$

Experiments

- 3 types of experiments
  - Tracker evaluation
  - Detector evaluation
  - Evaluation of the combination of both

- Data
  - 4 sequences captured by Ladybug 3
  - 5 fps, 12 Mpix
  - 8 tested objects (6 semi-planar, 2 non-planar)

Examples of tested objects

Tracker evaluation

Implemented in Matlab

Learning:

6 sec – translation SLLiP, 1500 examples

9 sec – homography SLLiP, 3500 examples

Tracking:

4 ms both SLLiPs together

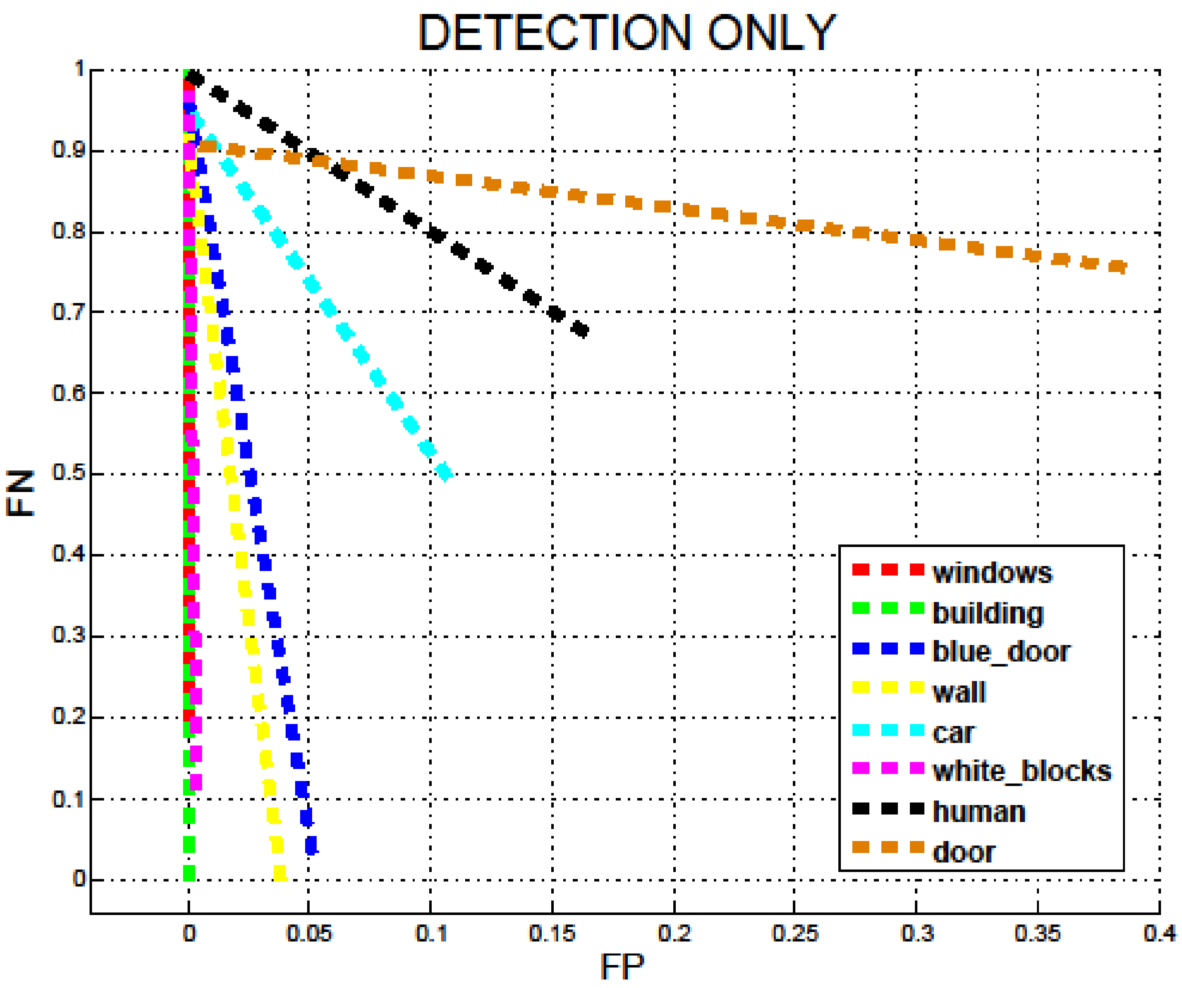
Validation:

72 ms

Incremental learning:

220 ms - each SLLiP, 10 examples

Detector evaluation



Implemented in C

Learning:

10 ms per class

11 binary tests

50 ferns

500 examples per class

Detection:

0,13 ms per one harris

In one Image:

5350 harrises

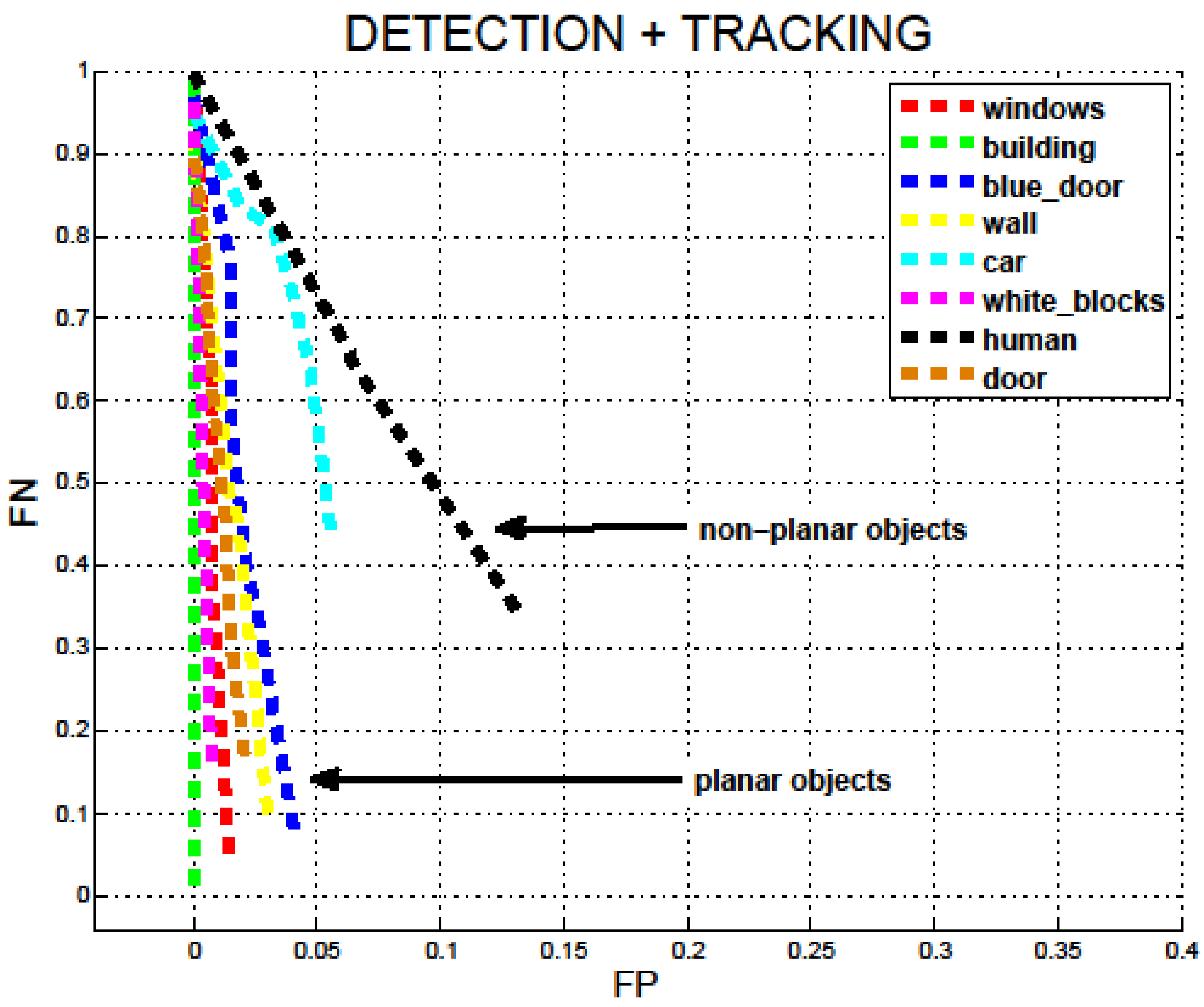
50 ferns

200 classes

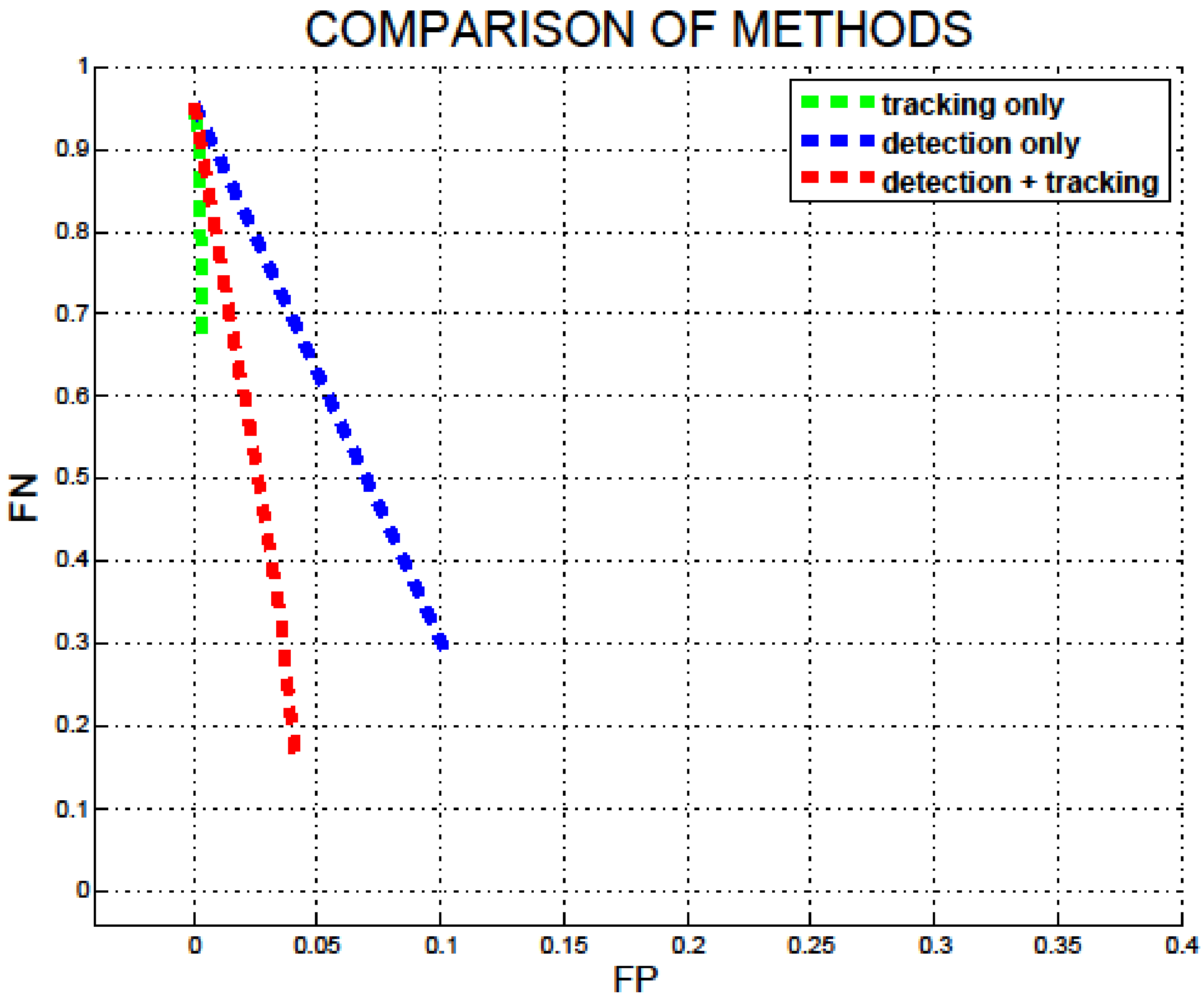
0,7 sec for 1 scale

run in 2 or 3 scales

Our Algorithm



Comparison


COMPARISON OF METHODS
tracking only
detection only
detection + tracking
FN
FP
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4


RESULTS OF DETECTION AND TRACKING
IN HIGH-RESOLUTION 8 FPS VIDEO

Thank you for attention