NYU TANDON SCHOOL OF ENGINEERING

AI4CE

Robot Vision

Visual Tracking

Dr. Chen Feng

cfeng@nyu.edu

ROB-UY 3203, Spring 2024

Overview

- Visual Tracking
  - Basics definitions
  - KLT
  - Mean-shift
  - Correlation filter
- CNNs for Tracking
  - Supervised

References

- Szileski 2022
  - Section 9.3, 9.4.4
- Forsyth & Ponce 2011
  - Chapter 11
- Baker, Simon, and Iain Matthews. "Lucas-kanade 20 years on: A unifying framework." *International journal of computer vision*56.3 (2004): 221-255.
- Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5 (2002): 603-619.
- Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Real-time tracking of non-rigid objects using mean shift." In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 2000.
- Bertinetto, Luca, Jack Valmadre, Joao F. Henriques, Andrea Vedaldi, and Philip HS Torr. "Fully-convolutional siamese networks for object tracking." In European conference on computer vision, pp. 850-865. Springer, Cham, 2016.

Tracking an Object

Why Tracking?

Other than *classification* and *detection*, object *tracking* is one of the fundamental primitive tasks that is often required for more complex, real-world relevant downstream applications.

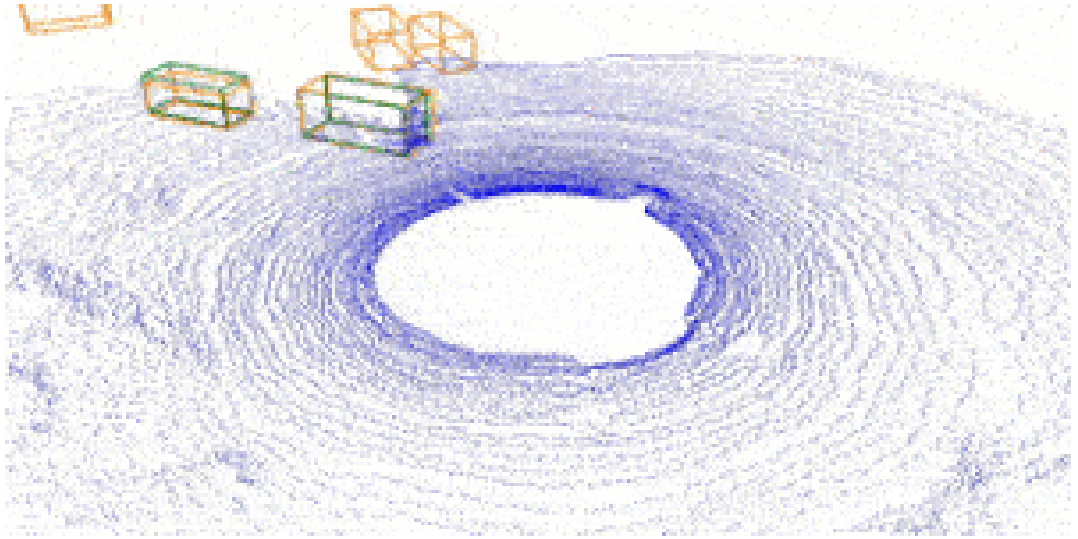
Can you think of any such downstream tasks?

Gif from https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/

Tracking Applications

- Autonomous driving

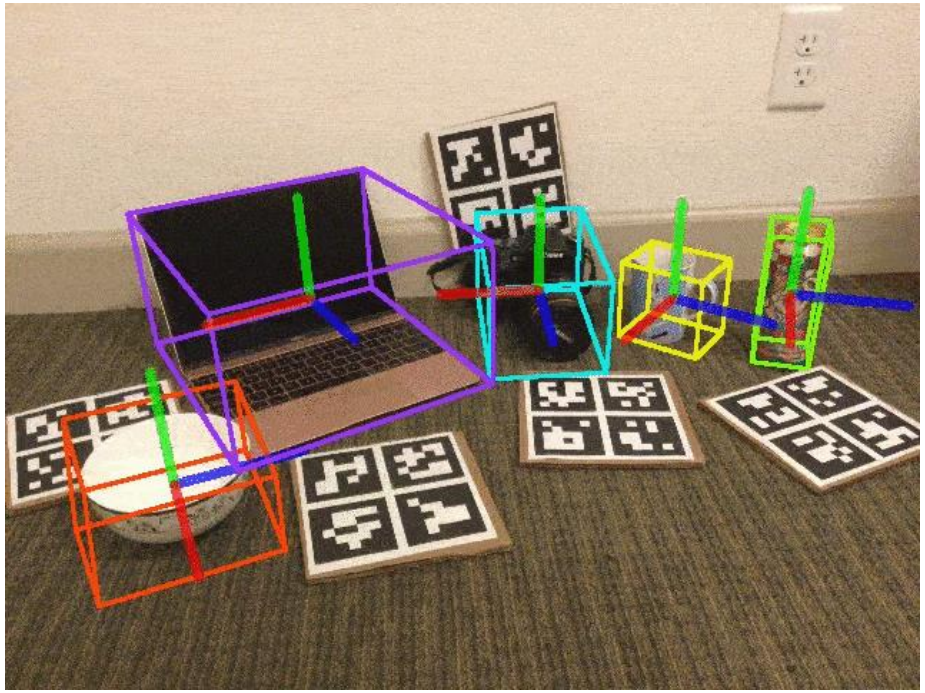
CVPR 2020 GNN3DMOT

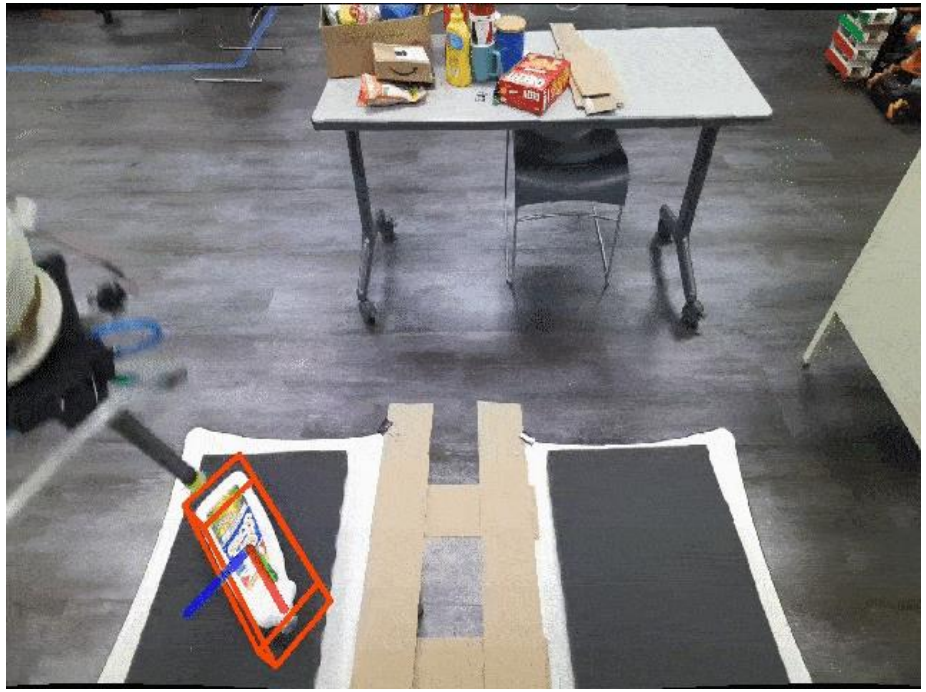


CoRL 2020 Inverting the Pose Forecasting Pipeline with SPF2

Tracking Applications

- Robotic manipulation





IROS 2021 BundleTrack: https://github.com/wenbowen123/BundleTrack

Tracking Applications

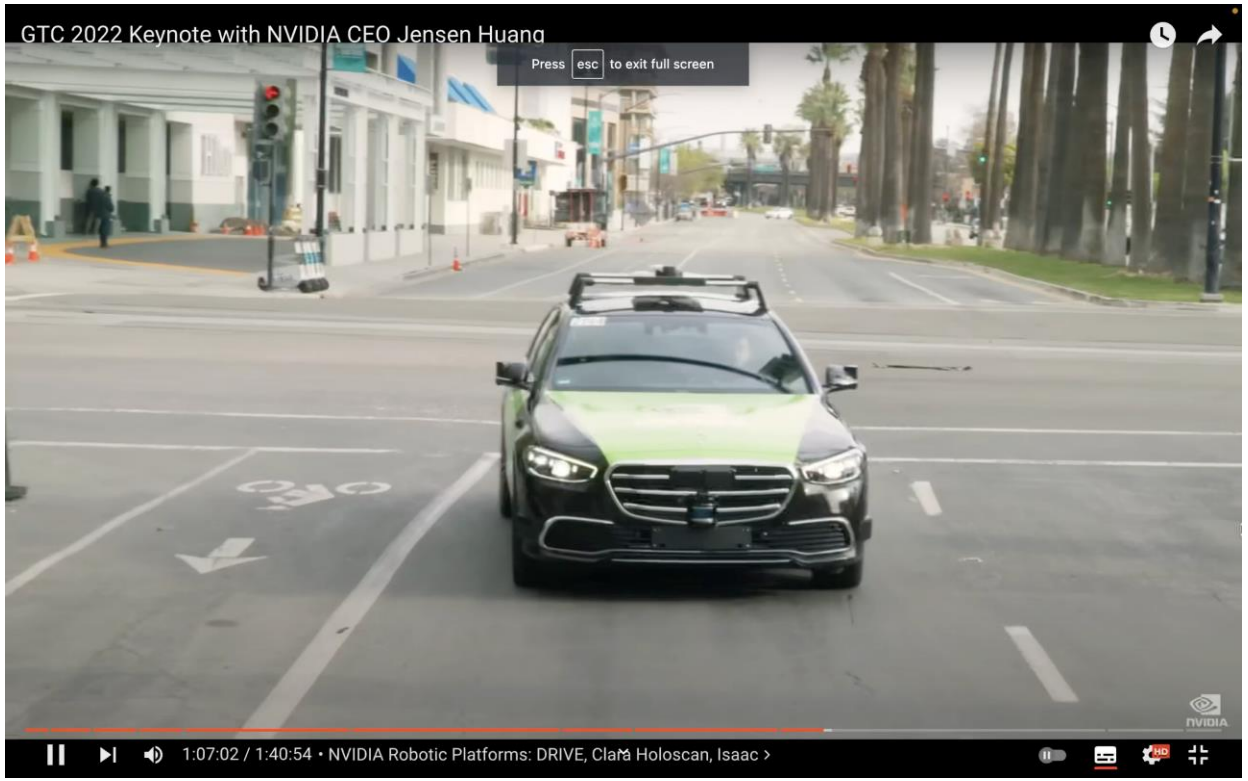
- Traffic analysis

Real-time object detection and tracking using YOLOv3 and Deep SORT

Tracking Applications

- Human computer interaction (HCI)

https://github.com/antoinelame/GazeTracking



https://www.youtube.com/watch?v=39ubNuxnrK8

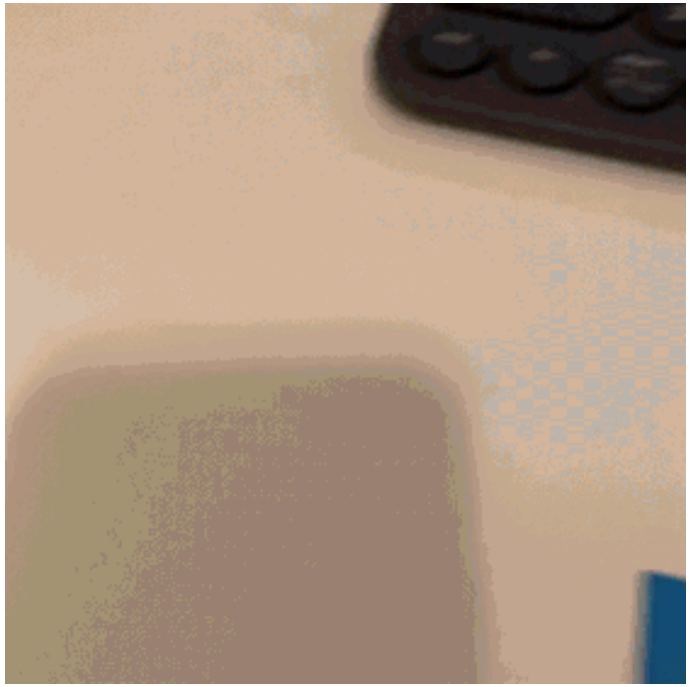
Tracking Applications

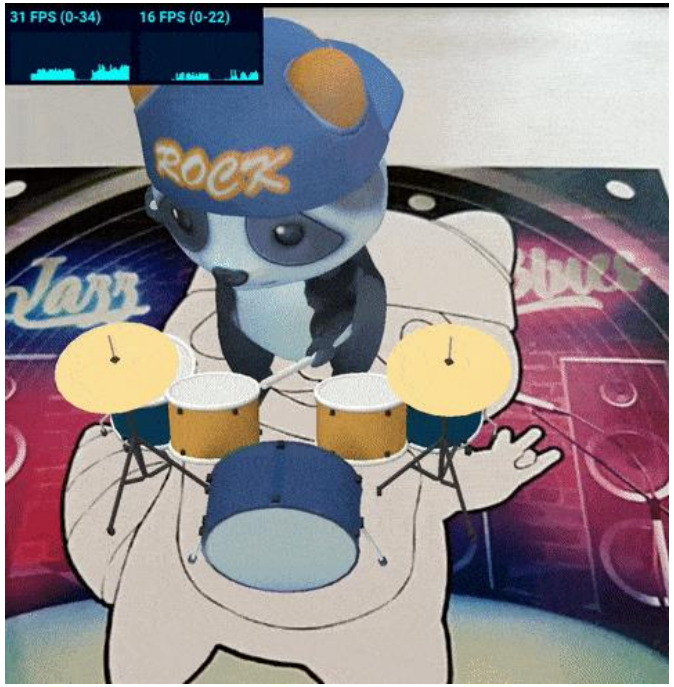
- Management of video content

Automating Basketball Highlights with Object Tracking

Tracking Applications

- Augmented reality





https://github.com/hiukim/mind-ar-js

Tracking Applications

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

Slides from Jiri Matas 2016

Tracking Definition

- [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.”
- Another definition
  - Establishing point-to-point correspondences in consecutive frames of an image sequence
- Yet another definition
  - Given an initial estimate of its position, locate X in a sequence of images
  - X can be a region, an interest point, or an object

Slides from Jiri Matas 2016

Tracking Definition

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

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

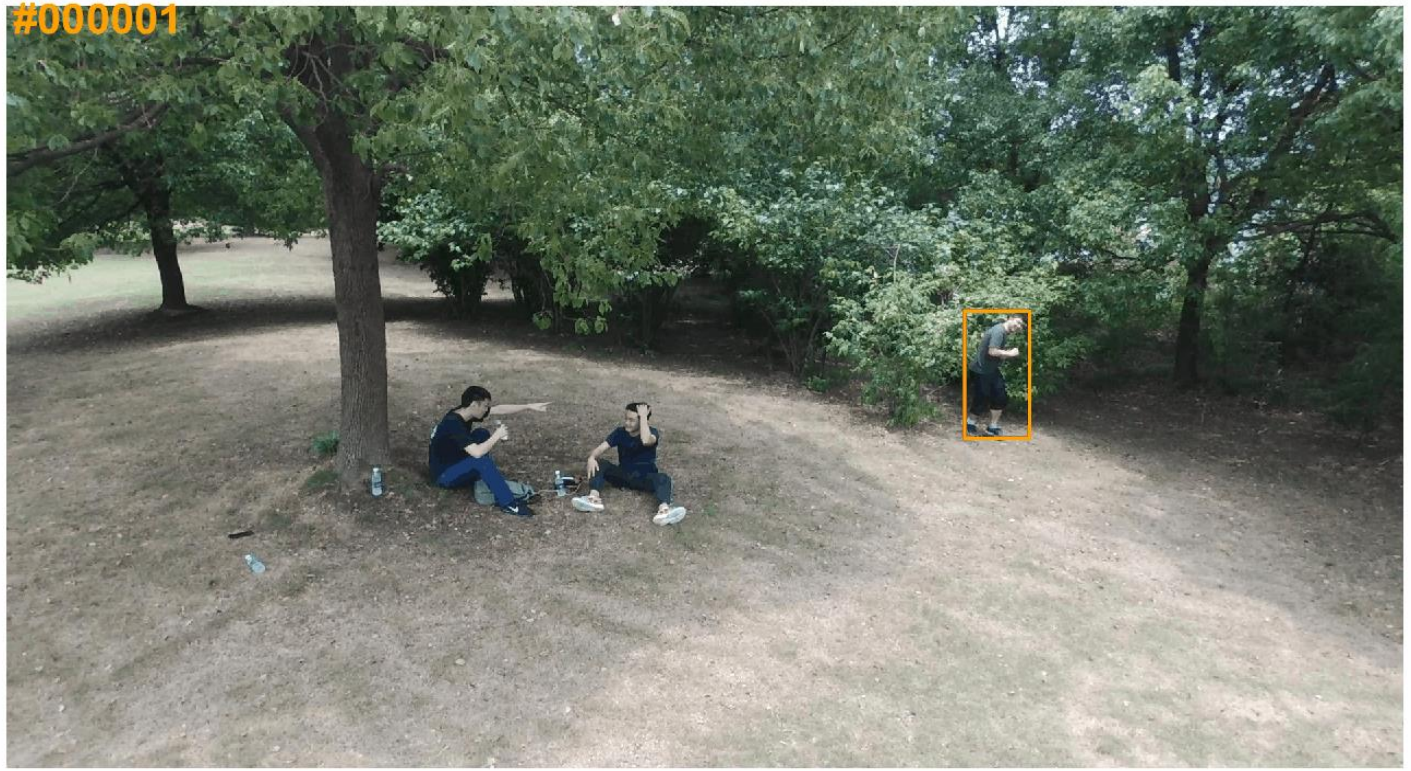
1. *estimate the pose and state of X*
2. *(optionally) update the model of X*

- 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 $t \leq T$

Slides from Jiri Matas 2016

Tracking Challenges

- Illumination change
- Camera motion



Tracking Challenges

- Similar objects
- Internel deformation

Tracking Challenges

- Partial occlusion
- Viewpoint change
- Scale variation

Types of Tracking – Time Variations

Short-term Trackers:

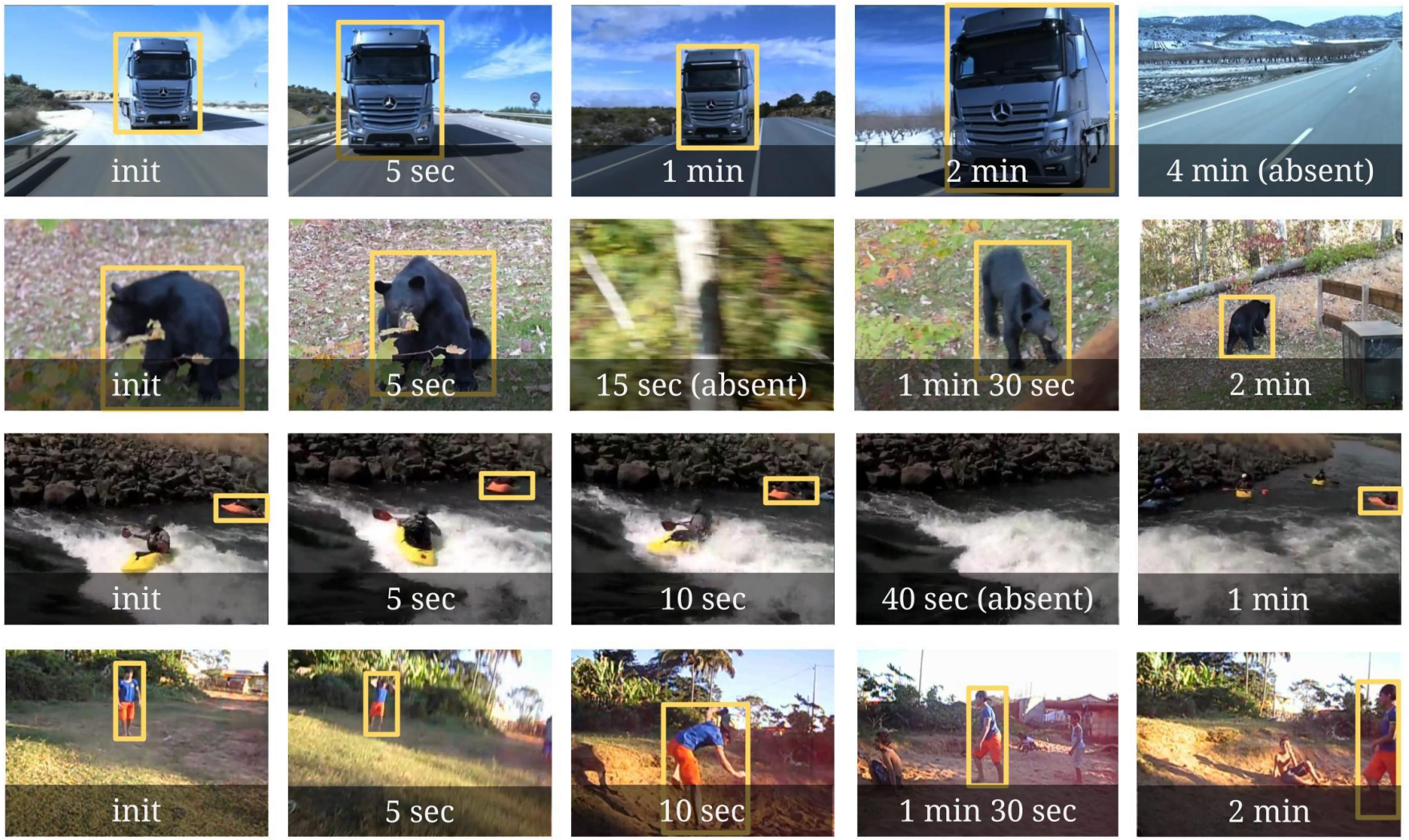
- primary objective: "where is X?" = precise estimation of pose
- secondary: be fast; don't lose track
- evaluation methodology: frame number where failure occurred
- examples: Lucas Kanade tracker, mean-shift tracker

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.

Slides from Jiri Matas 2016

Example of Long-Term Tracking Sequences



Long-term Tracking in the Wild: A Benchmark, Jack valmadre et al., 2018

Types of Tracking – Instance Variations

Note that the variations are *orthogonal* – *e.g.* can be both **MOT** and **Long-Term** in a given task!

Single Object Tracking (SOT)

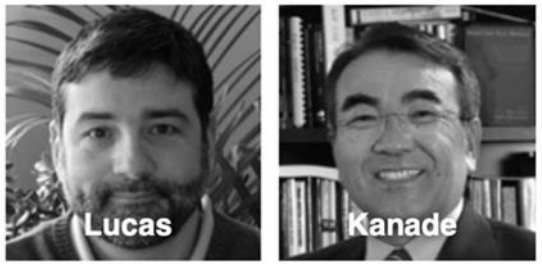
- Tracks a single subject (object) only.
- Usually the simpler task. We can do MOT by stacking SOT *if* we can tell the entities apart.

Multi Object Tracking (MOT)

- Tracks multiple subjects (objects) in a given seen.
- Typically, the harder task since we have to not only *track* a subject, but also often times we have to *disambiguate* the subjects from one another.

MotionTrack: Learning Robust Short-term and Long-term Motions for Multi-Object Tracking, Zheng Qin et al., 2023

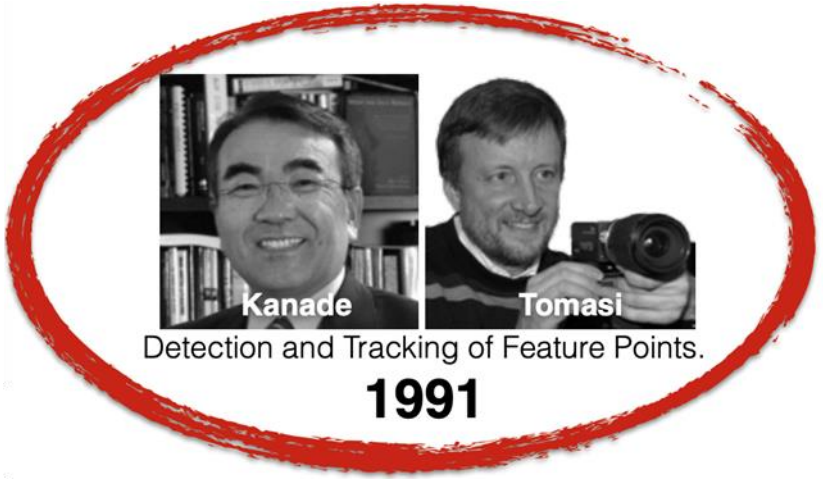
KLT Tracker – History of Kanade-Lucas-Tomasi Tracker



An Iterative Image Registration Technique with an Application to Stereo Vision.

1981

How to align/track an image patch?



Detection and Tracking of Feature Points.

1991

How to select features to track?

Good Features to Track.

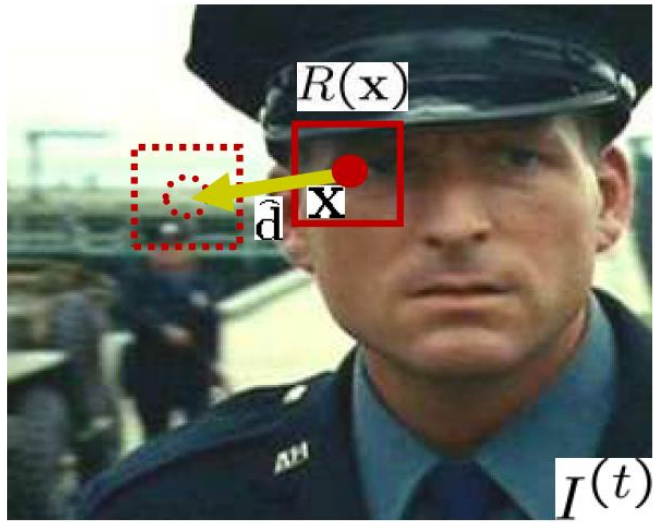
1994

Slides modified from Kris Kitani

KLT Tracker

– 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



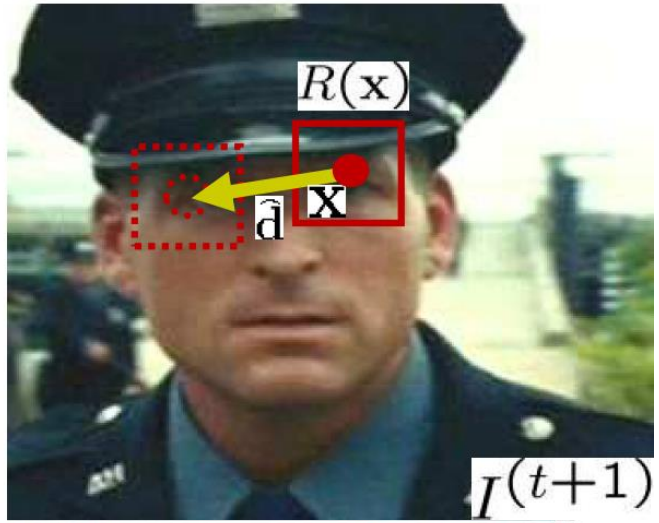
$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \underbrace{\sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2}_{\text{SSD}}$$

Slides from Jiri Matas 2016

KLT Tracker

– 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



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

Slides from Jiri Matas 2016

KLT Tracker

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

$$\hat{d} = \underset{d}{\operatorname{argmin}} \sum_{p \in R(x)} \left| I^{(t+1)}(p) + \nabla I^{(t+1)}(p)^T d - I^{(t)}(p) \right|^2$$

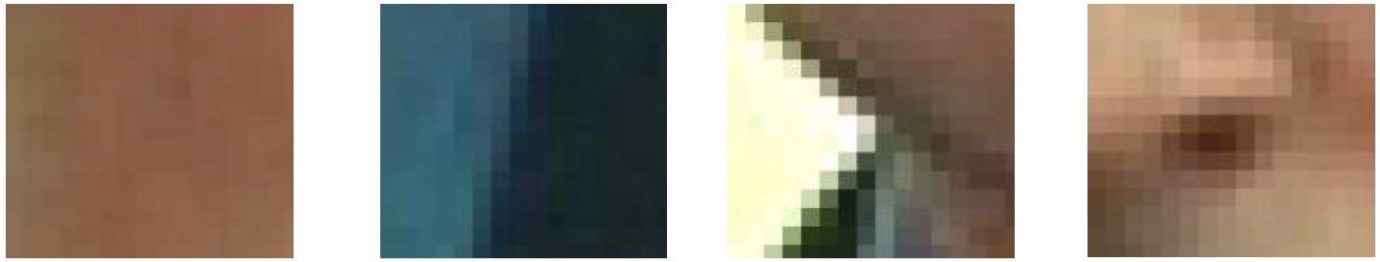
$$\hat{d} = -\left(\sum_{p \in R(x)} \nabla I^{(t+1)}(p) \nabla I^{(t+1)}(p)^T \right)^{-1} \sum_{p \in R(x)} \nabla I^{(t+1)}(p) [I^{(t+1)}(p) - I^{(t)}(p)]$$

Solving Ax=b

- Solve by least square: $(A^T A)^{-1} A^T b$

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]

Slides from Jiri Matas 2016

KLT Tracker

– 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

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

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

Slides from Jiri Matas 2016

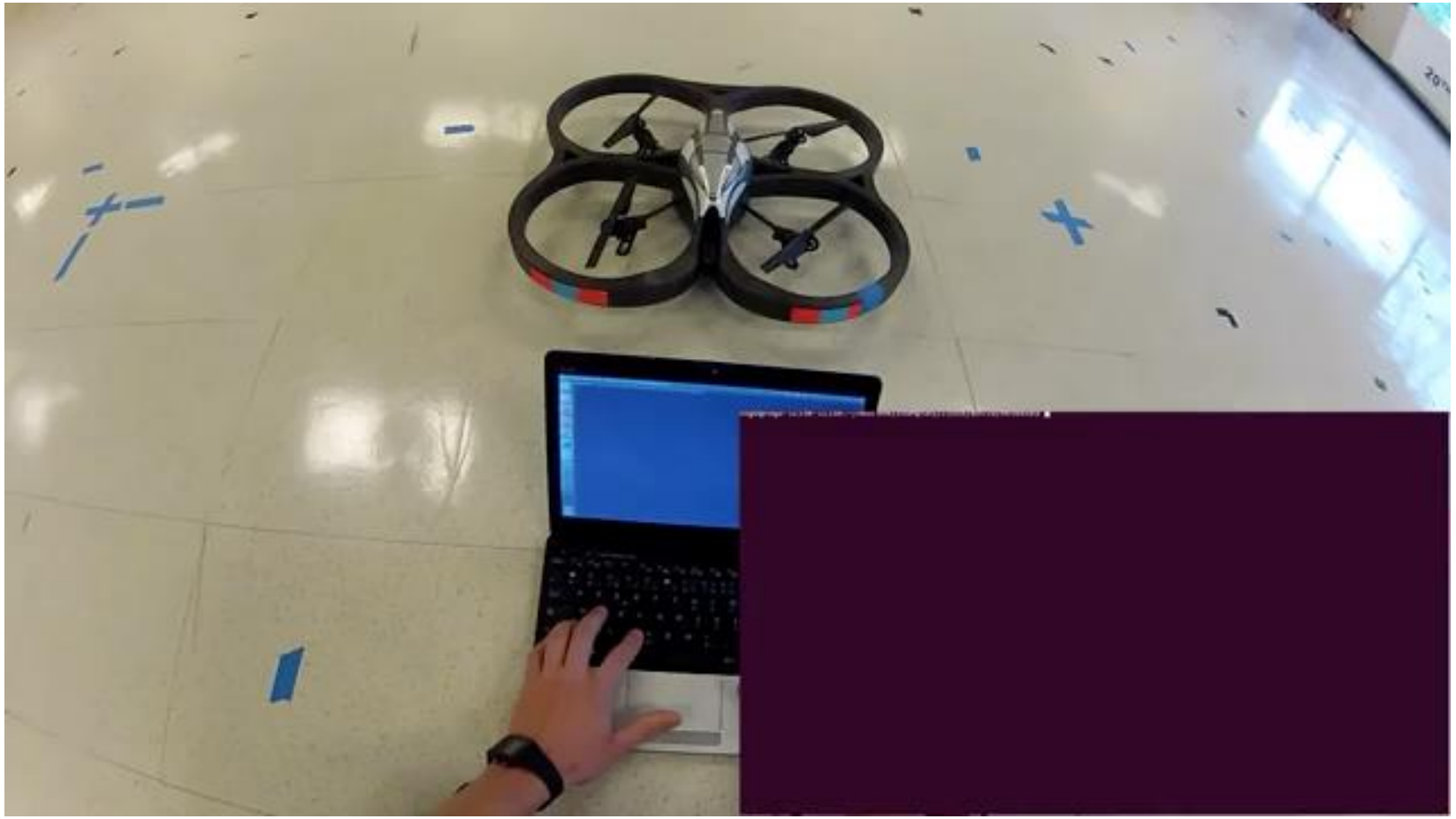
KLT Tracker

- cost function: sum of squared intensity differences between template and window
- optimization technique: gradient descent
- model learning: no update / last frame / convex combination

- attractive properties:
  - fast
  - easily extended to image-to-image transformations with multiple parameters

Slides from Jiri Matas 2016

Optical Flow in Robotics



https://youtu.be/C95bngCOv9Q

Mean Shift Theory

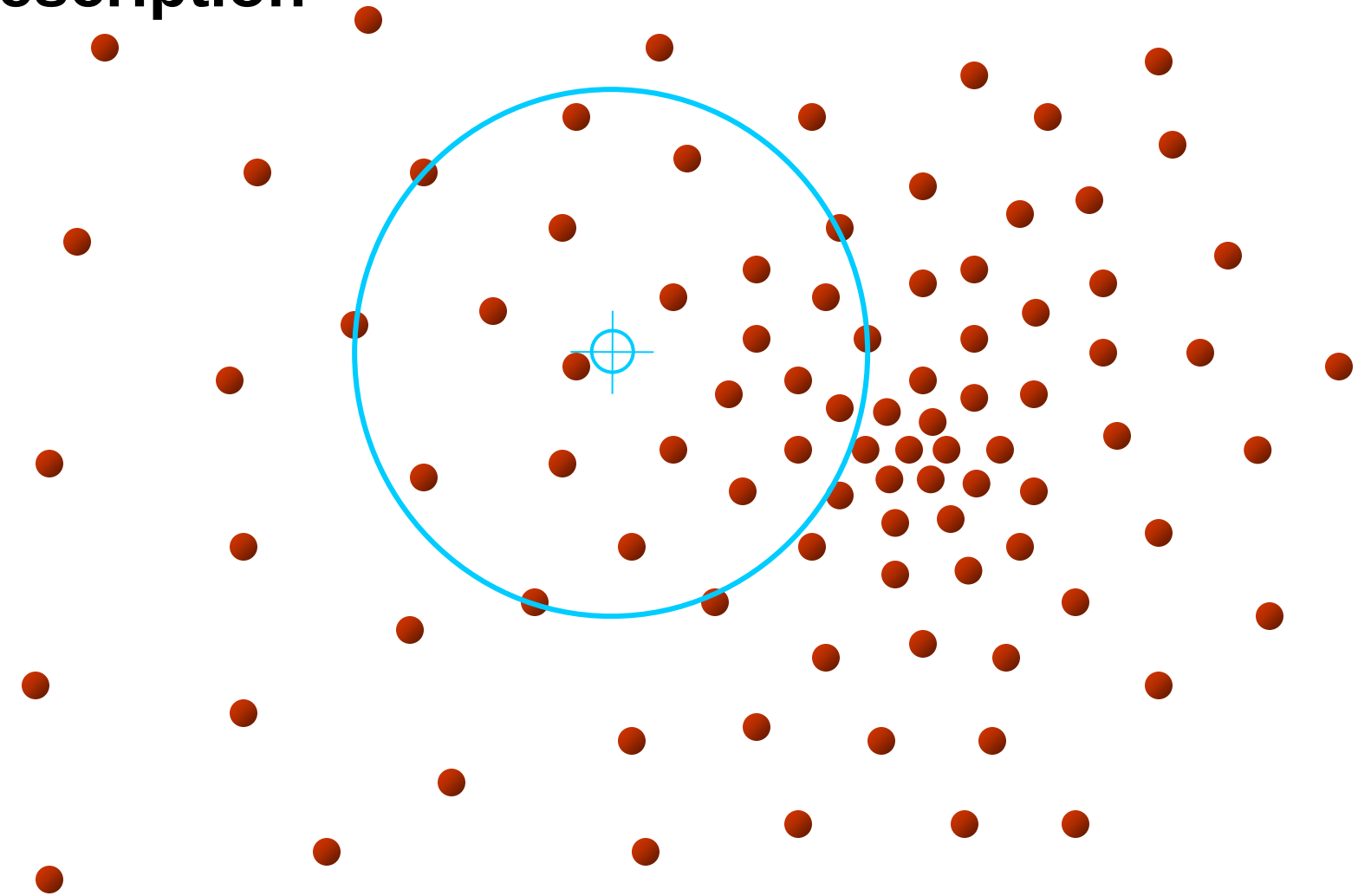
Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer.
"Real-Time Tracking of Non-Rigid Objects Using Mean Shift."

Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000. Vol. 2. IEEE, 2000.

Won the Best Paper Award CVPR 2000

Slides from Ukrainitz & Sarel

Intuitive Description

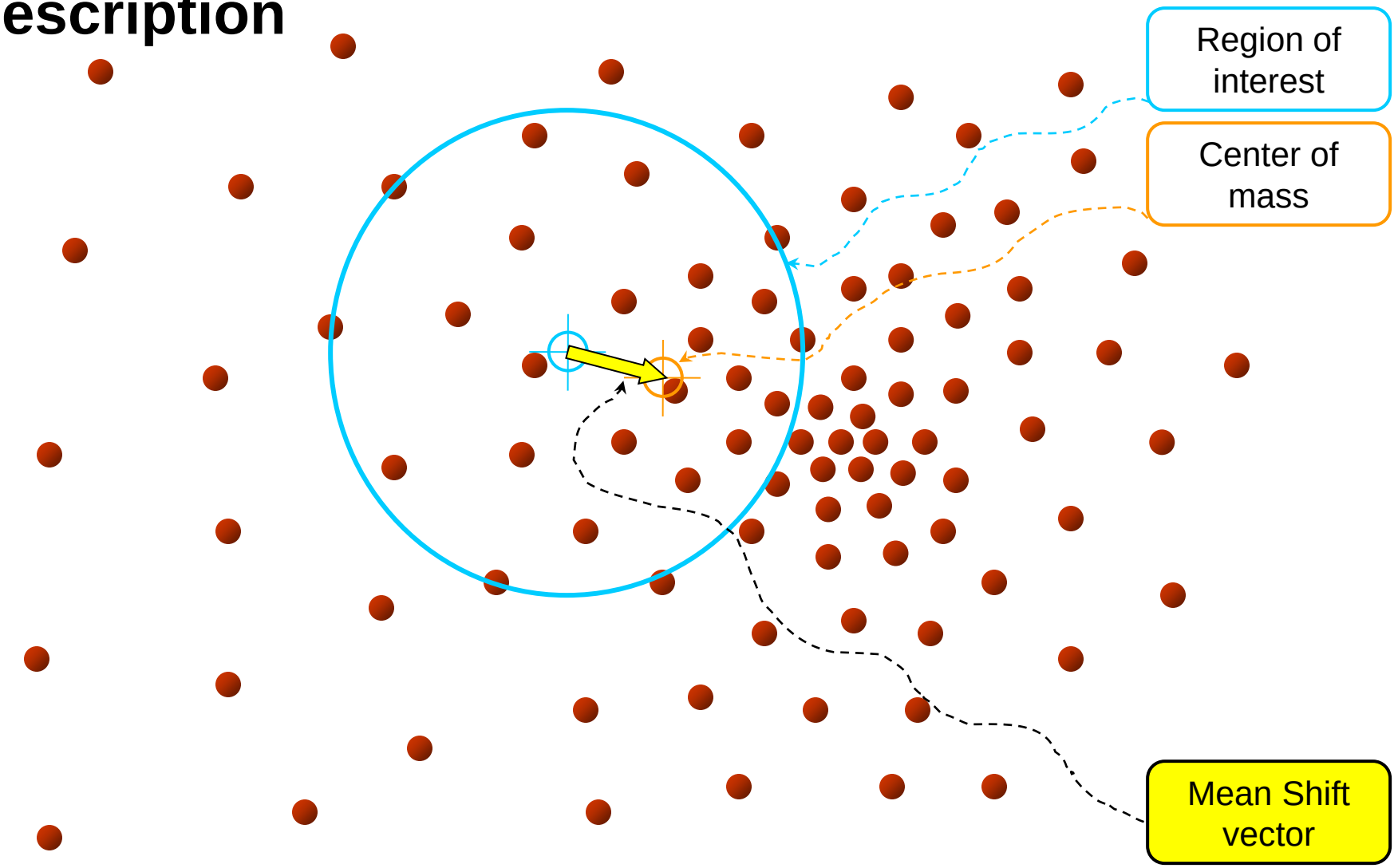


Objective : Find the densest region

Distribution of identical billiard balls

Slides from Ukrainitz & Sarel

Intuitive Description

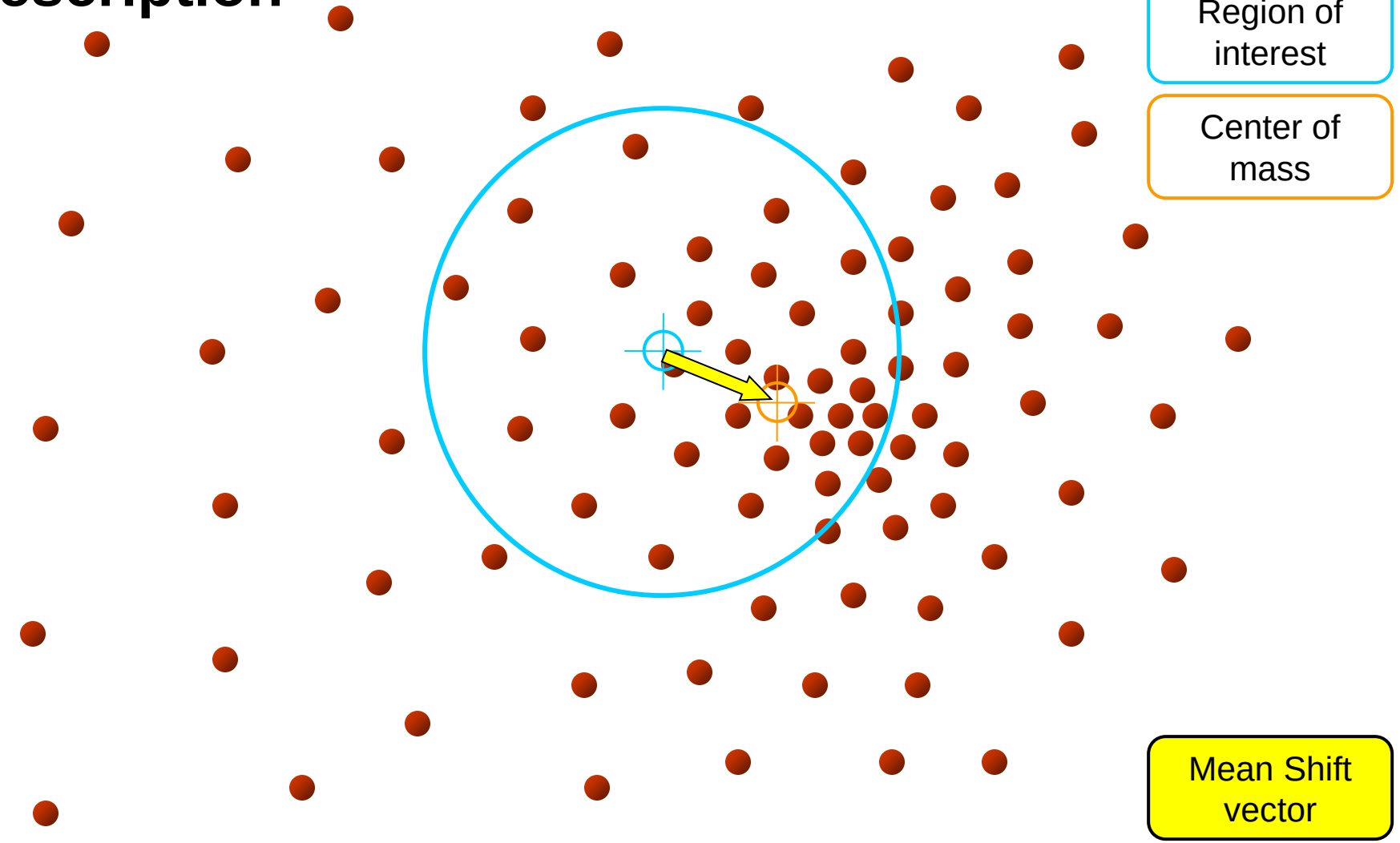


Objective : Find the densest region

Distribution of identical billiard balls

Slides from Ukrainitz & Sarel

Intuitive Description

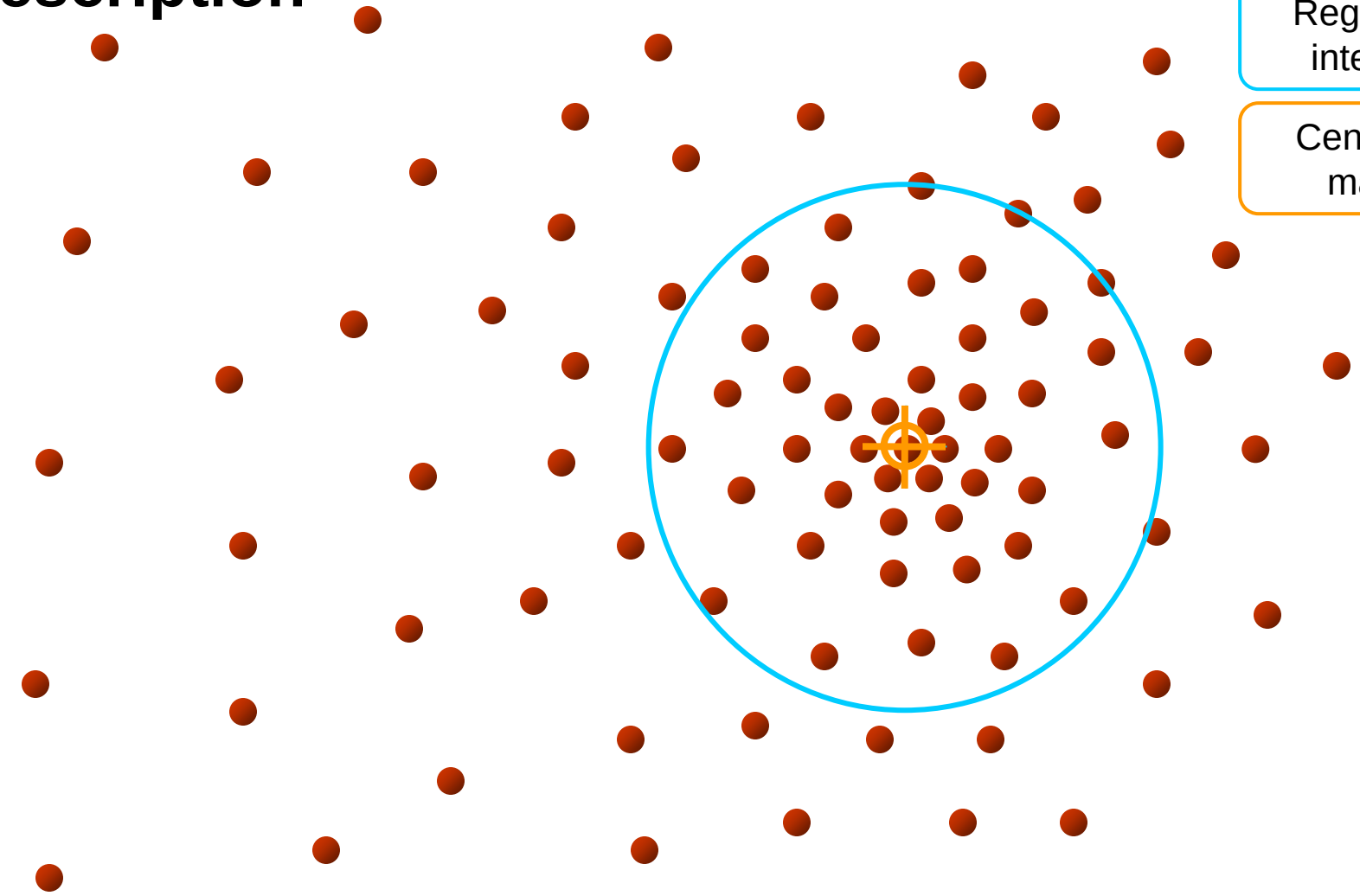


Objective : Find the densest region

Distribution of identical billiard balls

Slides from Ukrainitz & Sarel

Intuitive Description



Objective : Find the densest region

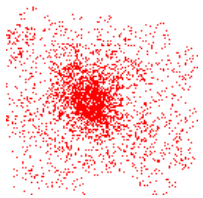
Distribution of identical billiard balls

Slides from Ukrainitz & Sarel

What is Mean Shift ?

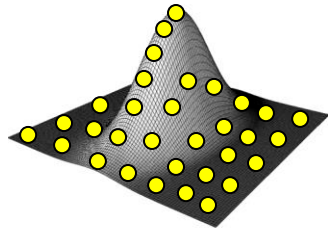
A tool for:
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in R^N

PDF in feature space
- **Color space**
- **Scale space**
- **Actually any feature space you can conceive**



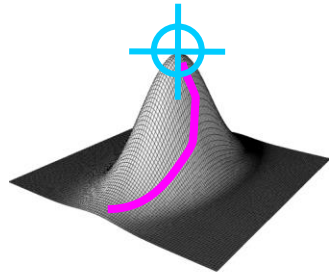
Data

Non-parametric Density Estimation



Discrete PDF Representation

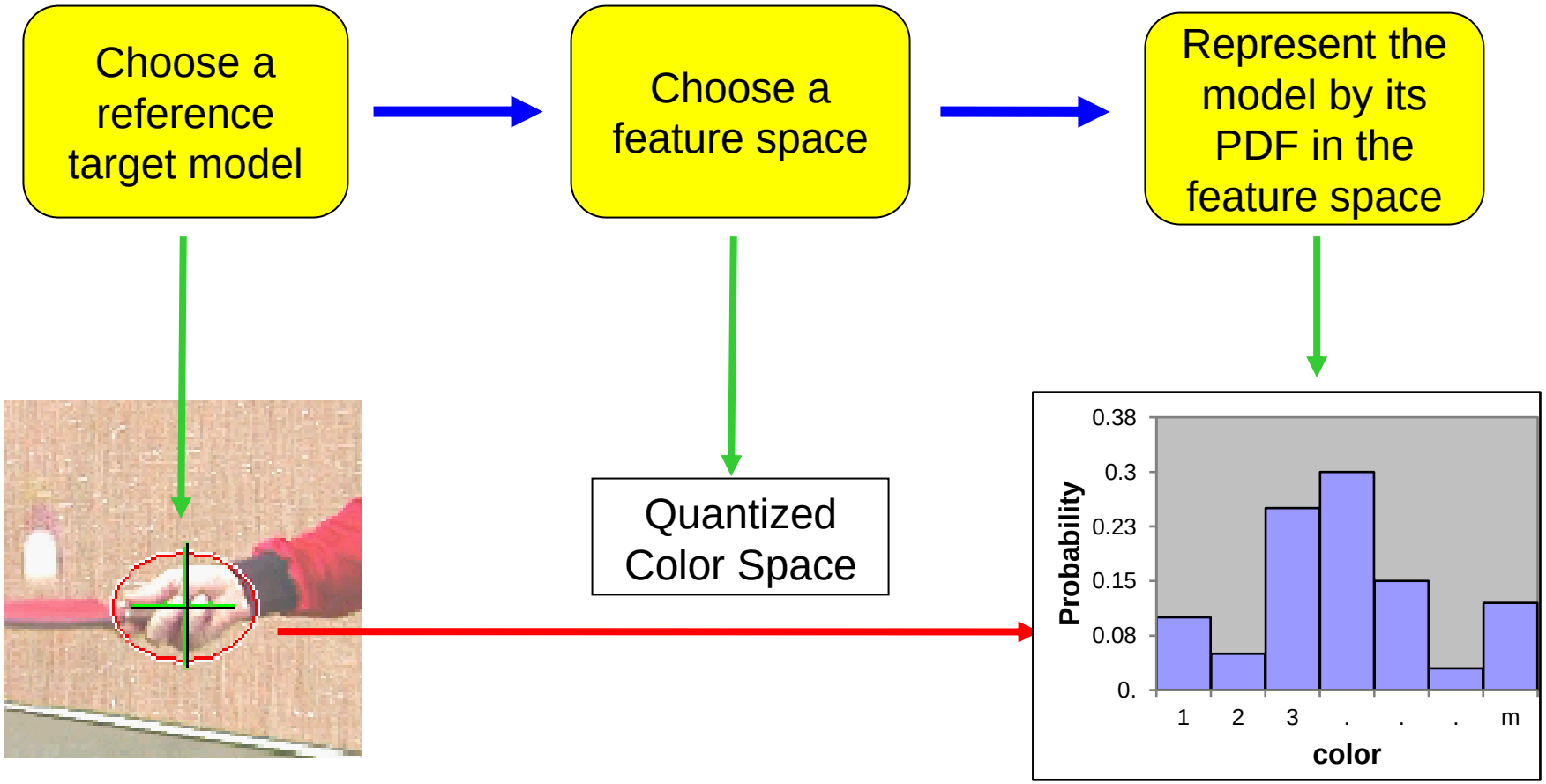
Non-parametric Density **GRADIENT** Estimation (Mean Shift)



PDF Analysis

Slides from Ukrainitz & Sarel

Mean-Shift Object Tracking - Target Representation

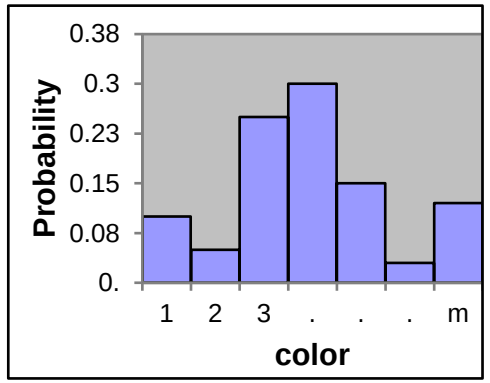


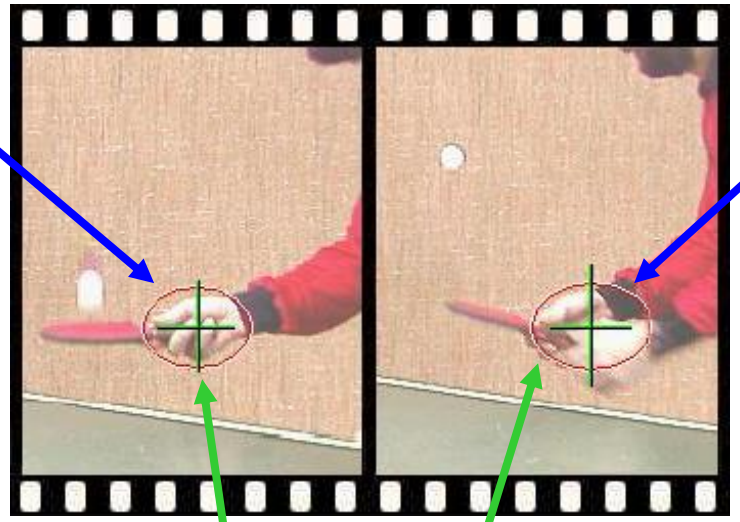
Slides from Ukrainitz & Sarel

Kernel Based Object Tracking, by Comaniniu, Ramesh, Meer

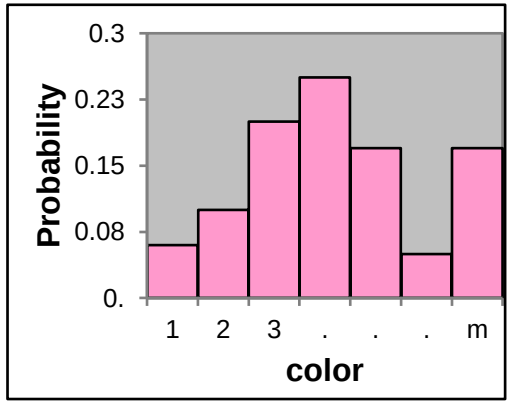
Mean-Shift Object Tracking - PDF Representation

Target Model
(centered at 0)





Target Candidate
(centered at y)



Similarity Function:

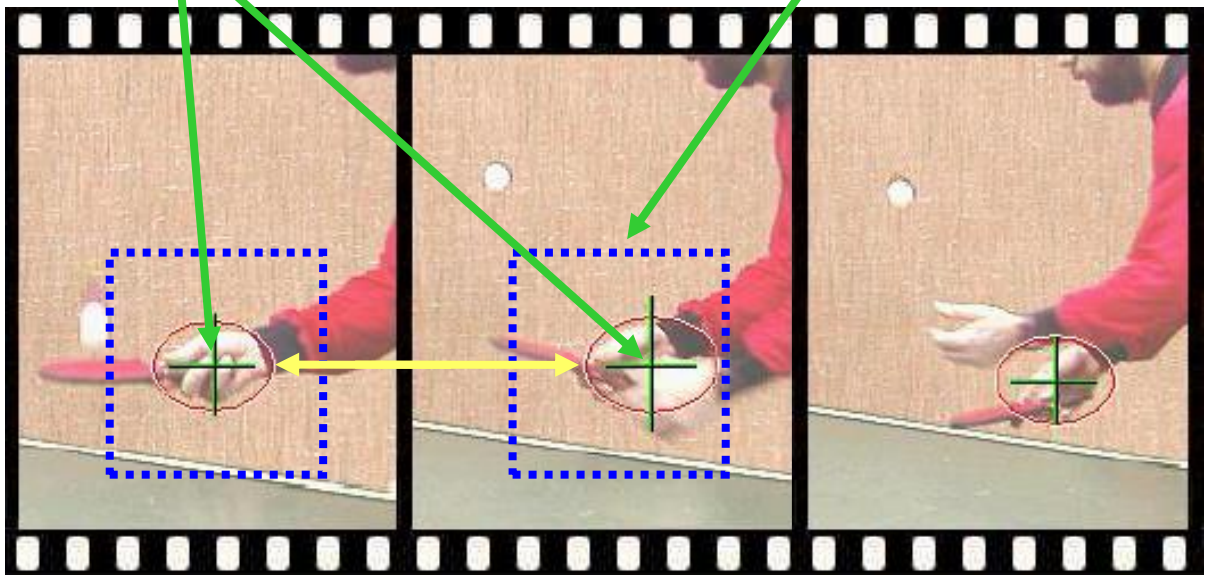
$$f(y) = f\left[\vec{q}, \vec{p}(y)\right]$$

Mean-Shift Object Tracking - Target Localization

Start from the position of the model in the current frame

Search in the model's neighborhood in next frame

Find best candidate by maximizing a similarity func.



… Current frame …

Repeat the same process in the next pair of frames

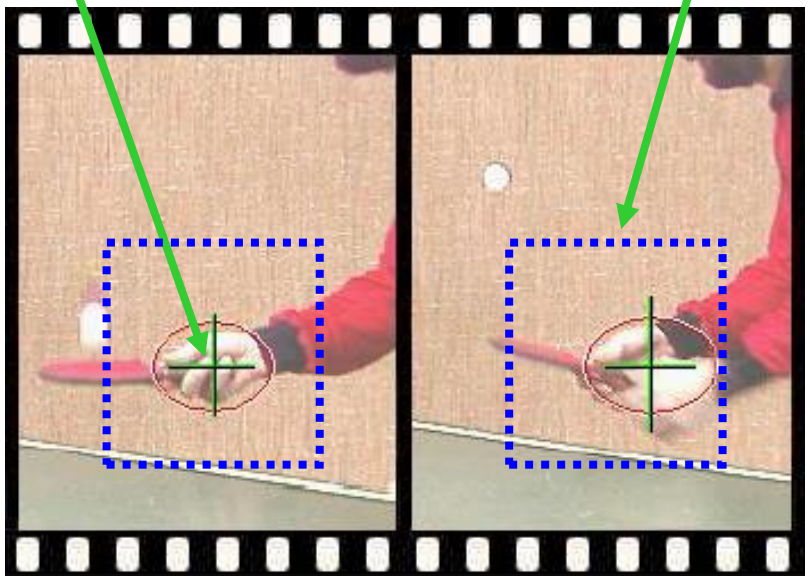
Slides from Ukrainitz & Sarel

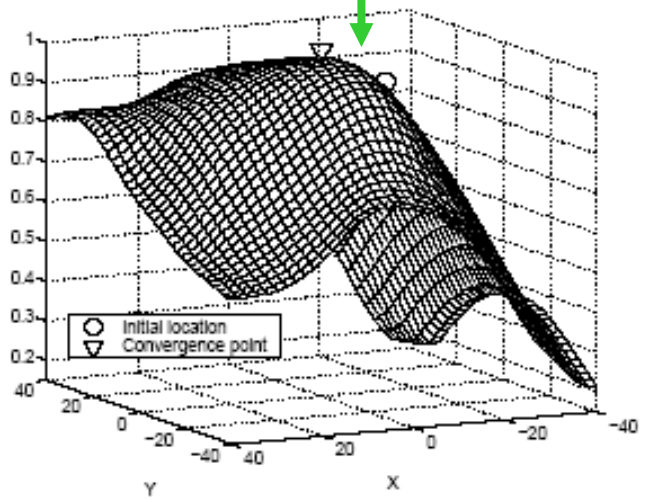
Mean-Shift Object Tracking - Target Localization

Start from the position of the model in the current frame

Search in the model's neighborhood in next frame

Find best candidate by maximizing a similarity func.





Slides from Ukrainitz & Sarel

Mean-Shift Pros/Cons

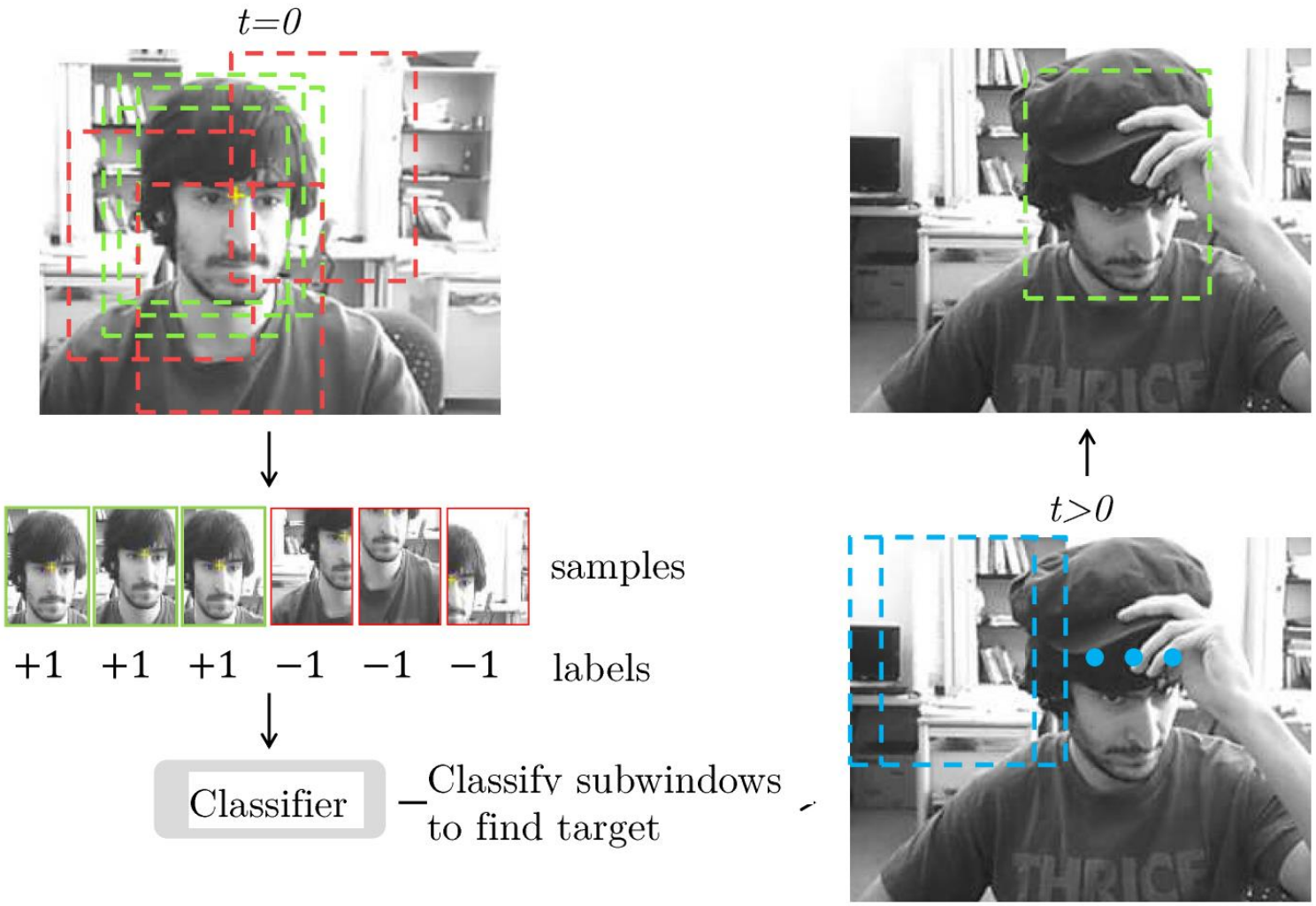
- Low computational cost (easily real-time)
- Surprisingly robust
  - Invariant to pose and viewpoint
  - Often no need to update reference color model
- Invariance comes at a price
  - Position estimate prone to fluctuation
  - Scale and orientation not well captured
  - Sensitive to color clutter (e.g., teamates in team sports)
- Local search by gradient descent
- Problems:
  - abrupt moves
  - occlusions

Slides from Jiri Matas 2016

Mean-Shift Example

https://youtu.be/RG5uV_h50b0

Discriminative Tracking: Tracking by Detection



Slides from Jiri Matas 2016

Connection to Correlation

- Let's have a linear classifier with weights $\mathbf{w}$

$$y = \mathbf{w}^T \mathbf{x}$$

- During tracking we want to evaluate the classifier at subwindows $\mathbf{x}_i$:

$$y_i = \mathbf{w}^T \mathbf{x}_i$$

- Then we can concatenate y_i into a vector $\mathbf{y}$ (i.e. response map)

- This is equivalent to **cross-correlation** formulation which can be computed efficiently in Fourier domain

$$\mathbf{y} = \mathbf{x} \circledast \mathbf{w}$$

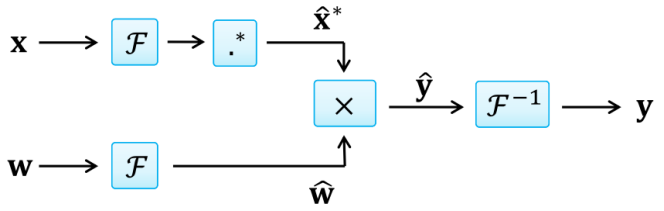
- Note: Convolution is related; it is the same as cross-correlation, but with the flipped image of $\mathbf{w}$ (P → d).

The Convolution Theorem

"Cross-correlation is **equivalent** to an **element-wise product** in Fourier domain"

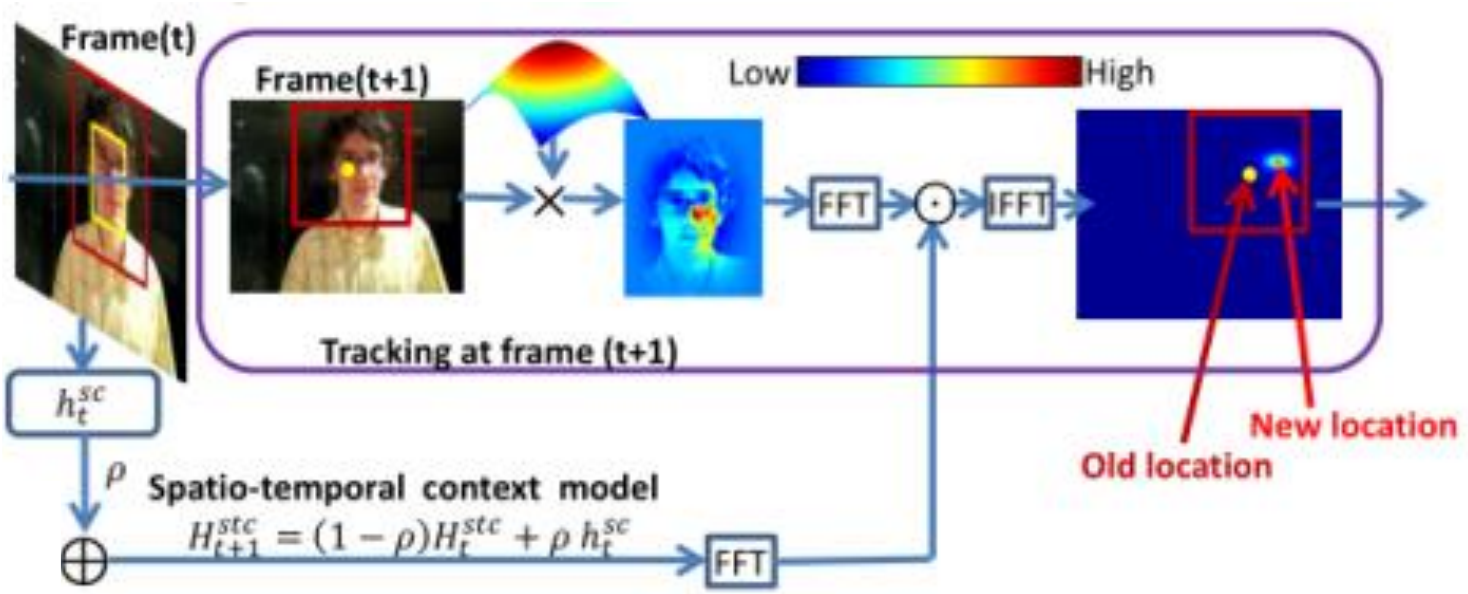
$$\mathbf{y} = \mathbf{x} \circledast \mathbf{w} \iff \hat{\mathbf{y}} = \hat{\mathbf{x}}^* \times \hat{\mathbf{w}}$$

- In practice:

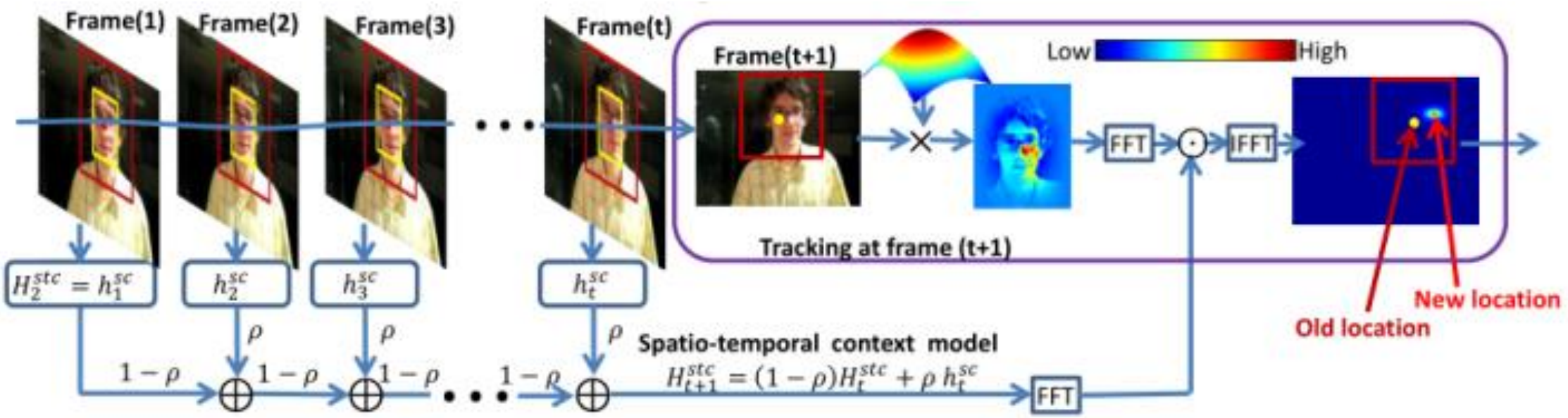


Slides from Jiri Matas 2016

Overall Framework



Overall Framework



Minimum Output Sum of Squared Error (MOSSE) based Correlation Filter

Slides from Jiri Matas 2016

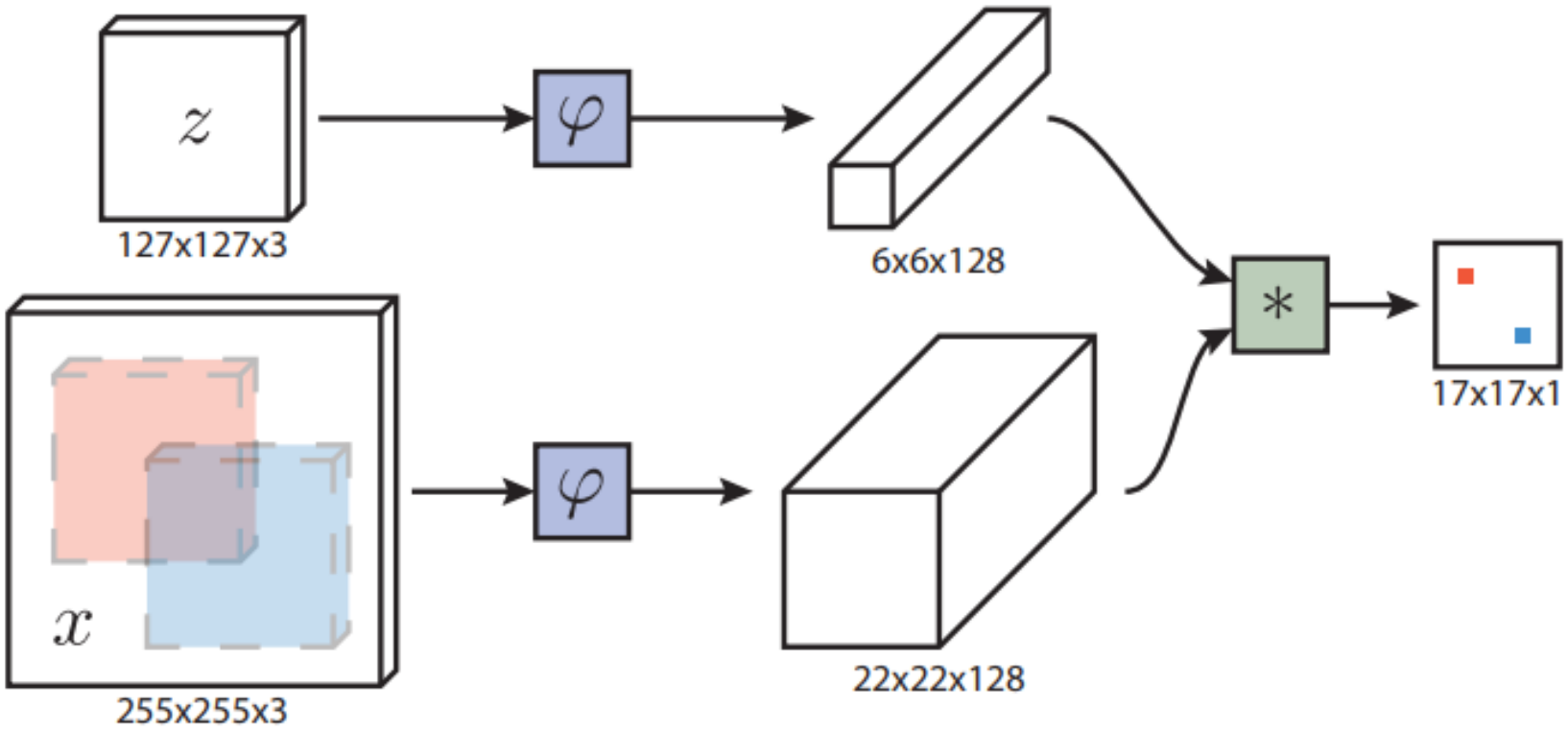
A Nice Summary for Tracking API in OpenCV

- https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/

Deep Learning based Tracking

Supervised Tracking

- **Paper : Fully-Convolutional Siamese Networks for Object Tracking**



Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A. and Torr, P.H., 2016. Fully-convolutional siamese networks for object tracking. ECCV Workshop.

Supervised Tracking

- **Training pairs**

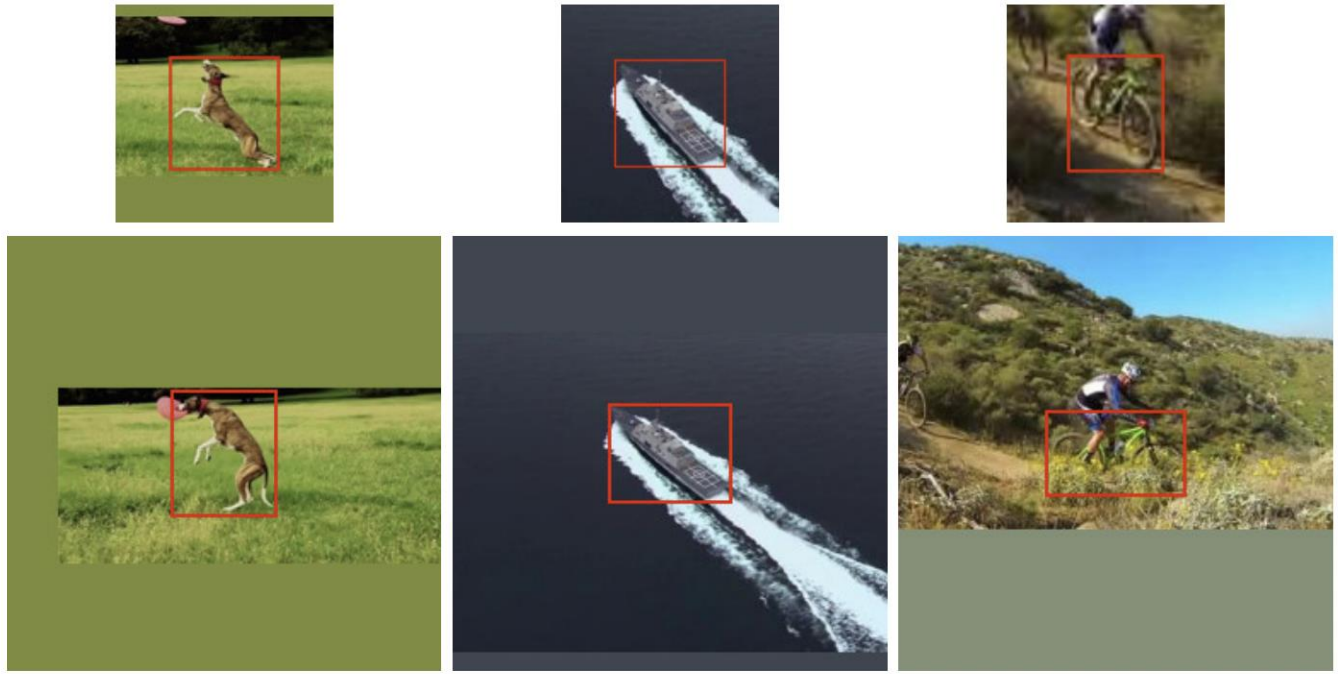
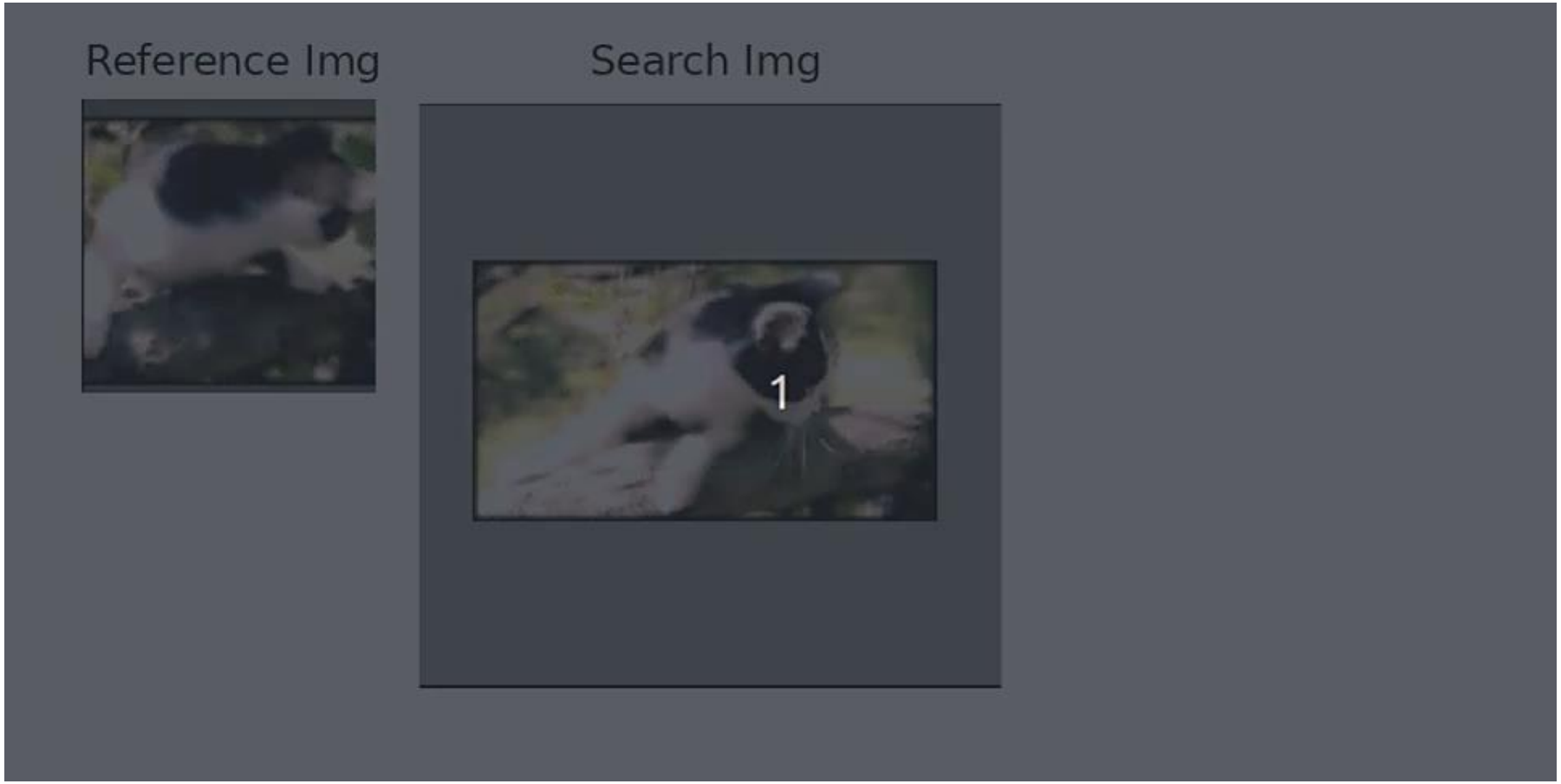


Fig. 2: Training pairs extracted from the same video: exemplar image and corresponding search image from same video. When a sub-window extends beyond the extent of the image, the missing portions are filled with the mean RGB value.

Supervised Tracking

- **Visualizations**



Supervised Tracking

- **Training loss**

ℓ: Binary Cross Entropy Loss (Logistic Loss)

$$L(y, v) = \frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{D}} \ell(y[u], v[u])$$

$$y[u] = \begin{cases} +1 & \text{if } k\|u - c\| \leq R \\ -1 & \text{otherwise} \end{cases}.$$

y: ground truth label at pixel u
v: prediction value at pixel u

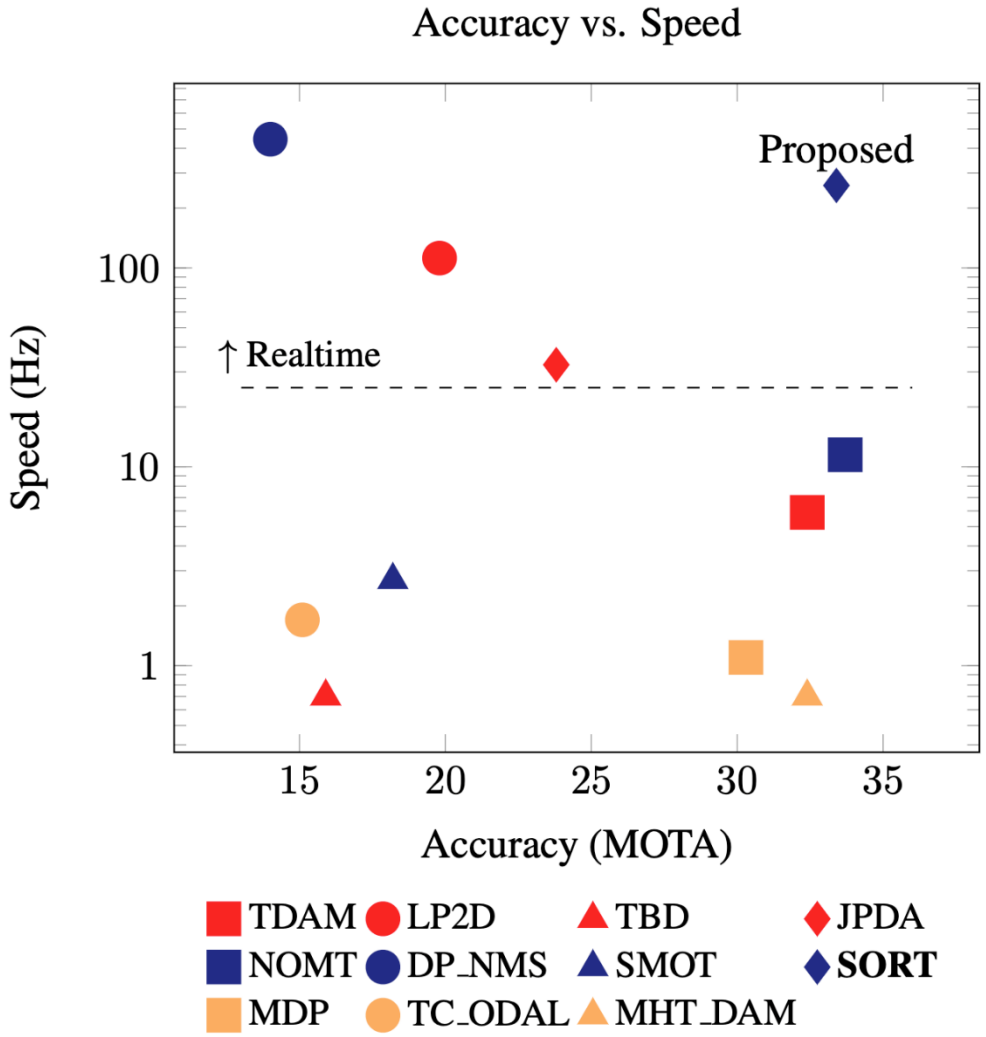
+1 only if pixel u is with R pixels from the ground truth center c

Supervised Tracking

- **Video**

CVPR 2019 Deeper and Wider Siamese Networks for Real-Time Visual Tracking

Simple Online & Realtime Tracking (SORT) for Multi-Object Tracking (MOT)



Key Ideas

1. Object tracking does not need to be complicated.

2. A track-by-detection framework – detect objects first, then assign.

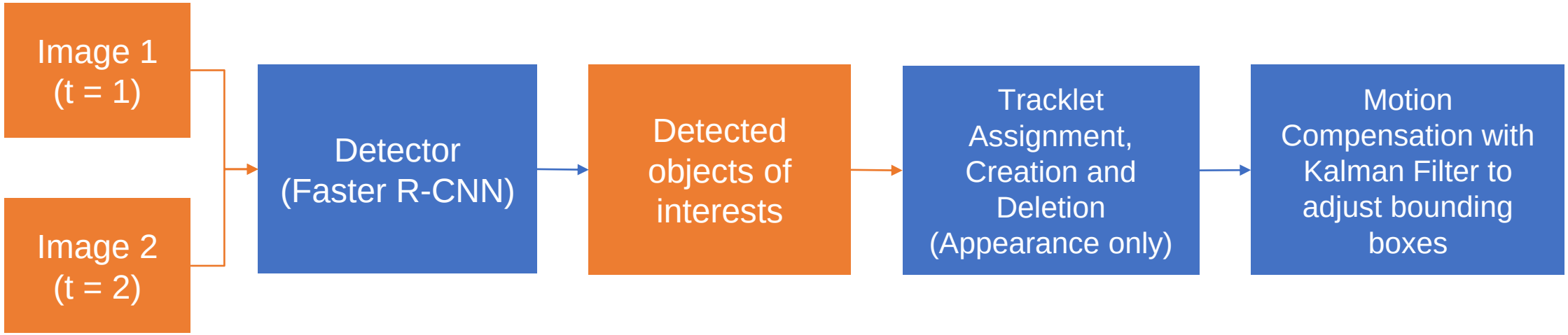
3. Key advantage of SORT is *speed!* Runs at ~260Hz / ~20x faster than baselines.

Simple Online & Realtime Tracking, Alex Bewley et al., ICIP 2016

Methodology of SORT

Key Ideas

1. Utilizes a pretrained Faster R-CNN instead of handcrafted descriptor.
2. For motion estimation, uses Kalman Filter for motion compensation (assumes fixed velocity; independent of camera motion)
3. For assignment, uses a simple appearance-matching schema of detect → match via IoU with Hungarian algorithm.
4. Finally, top it off with simple heuristics to create new tracklets and delete out-of-scene tracklets!



Simple Online & Realtime Tracking, Alex Bewley et al., 2016

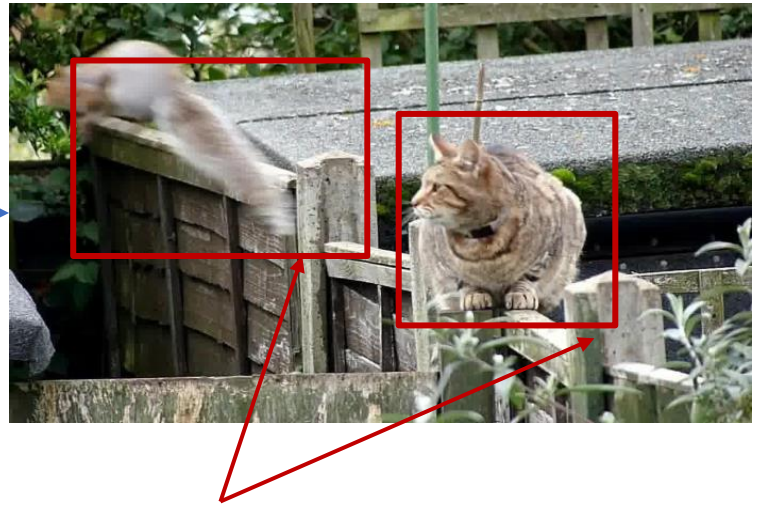
Methodology of SORT - Detection

Image t = 0

Image t = 1

Detector

Unassigned detections



Not that good either!

Simple Online & Realtime Tracking, Alex Bewley et al., 2016

Methodology of SORT – Assignment

Note: Tracklet creation and deletion not shown here

Image t = 0

Unassigned detections

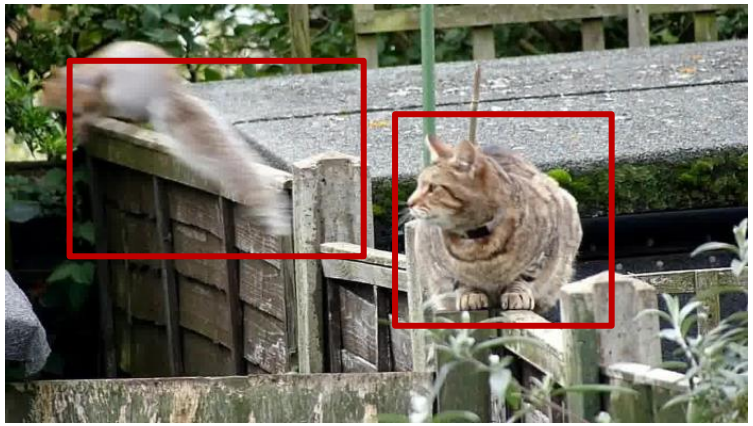


Image t = 1

Extract

Compute IoU Cost Matrix & assign via Hungarian Algorithm

Assigned detections

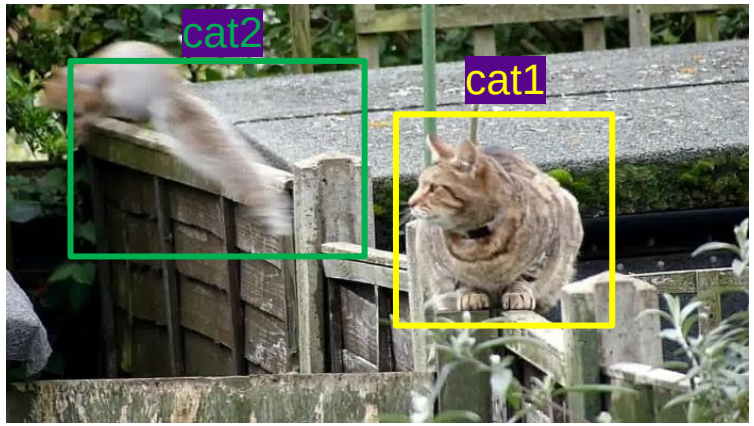


Image t = 1

Simple Online & Realtime Tracking, Alex Bewley et al., 2016

Methodology of SORT – Motion Compensation

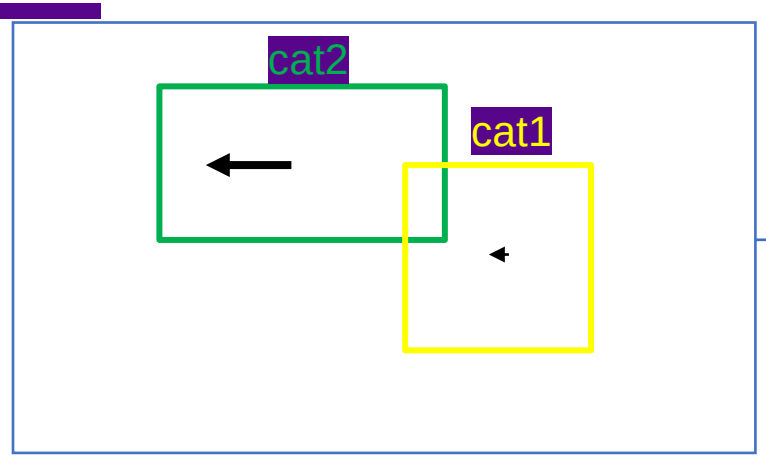


Image t = 0

Previous Bounding Box
State (with velocity)

Assigned detections

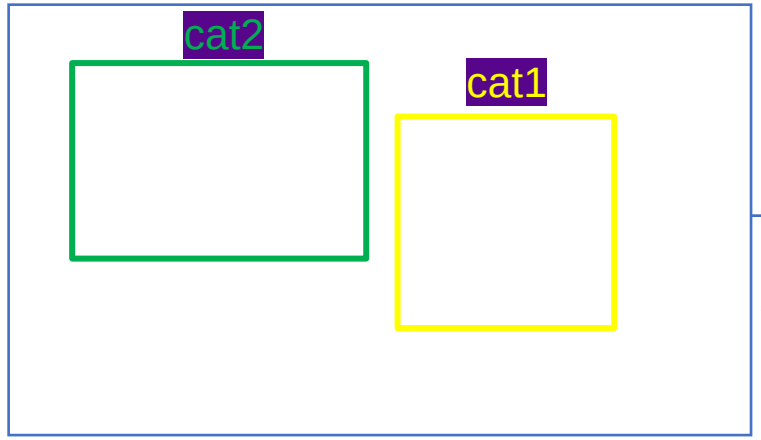


Image t = 1

Bounding Box
State (current detection)

Kalman Filter

Adjusted detections

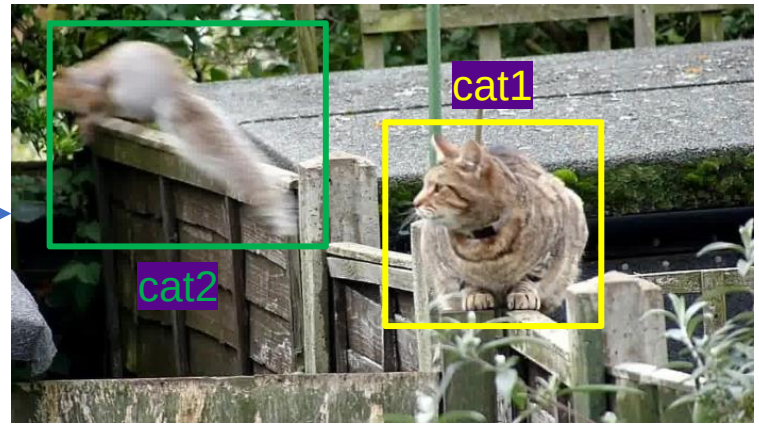


Image t = 1

Simple Online & Realtime Tracking, Alex Bewley et al., 2016

A Demo

Works pretty well!

The formulation naturally handles **short term occlusion** as well thanks to the motion compensation & appearance-based assignment algorithm.

But! It does not handle long term occlusion / long term tracking well! ☹

Can we do better?

Simple Online & Realtime Tracking, Alex Bewley et al., 2016

References for Next Week

- Szeliski 2022
  - Section 6.4

- Forsyth & Ponce 2011
  - Chapter 9

- Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision (pp. 945-953).
- Maturana, D. and Scherer, S., 2015, September. Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 922-928). IEEE.
- Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652-660).