Tracking

Dictionary:

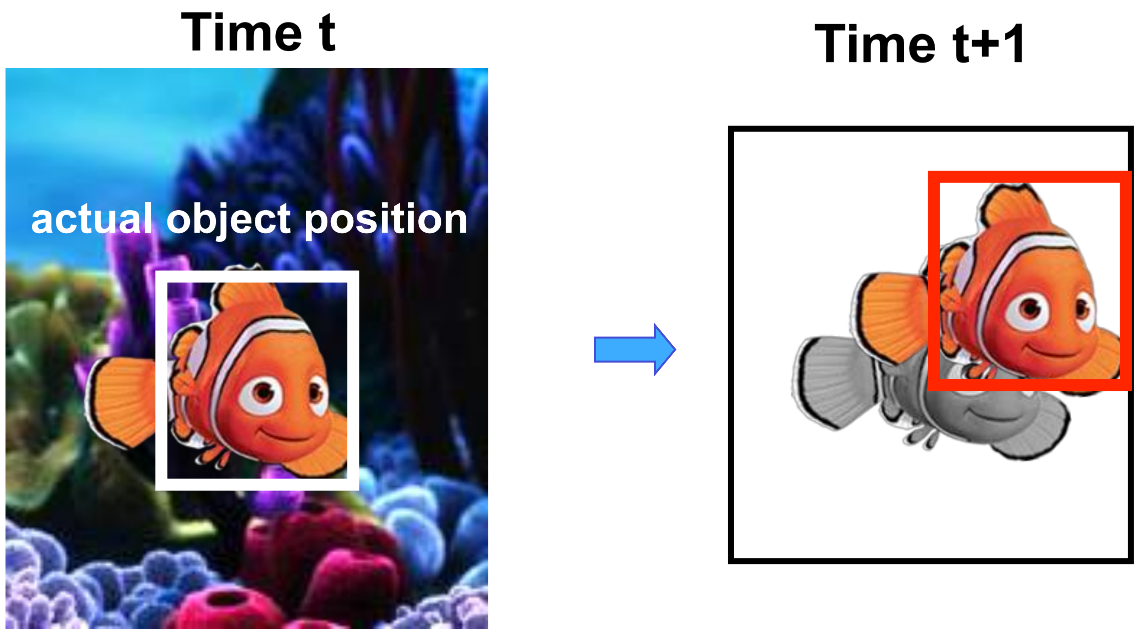
- [noun] “The pursuit (of a person or animal) by following tracks or marks they left behind”
- [verb] “Observe or plot the moving path of something (e.g., to track a missile)”

What does it mean in Computer Vision?

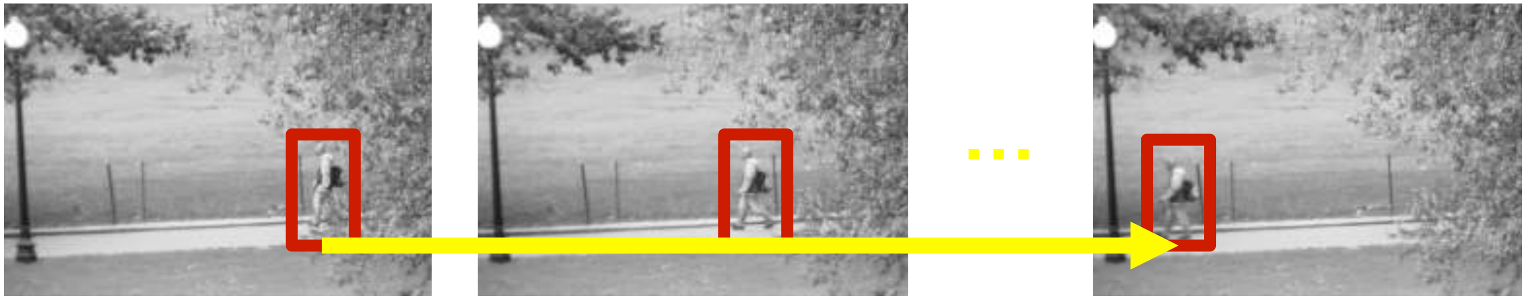
Many thanks to: H. Grabner, L. van Gool, and V. Ferrari for some of the slides & videos.

Computer Vision

What is Tracking



LOCALIZE "*IT*" IN THE NEXT FRAMES



Why do we need it

What is tracking for you? Why do you think it is relevant and may be important?

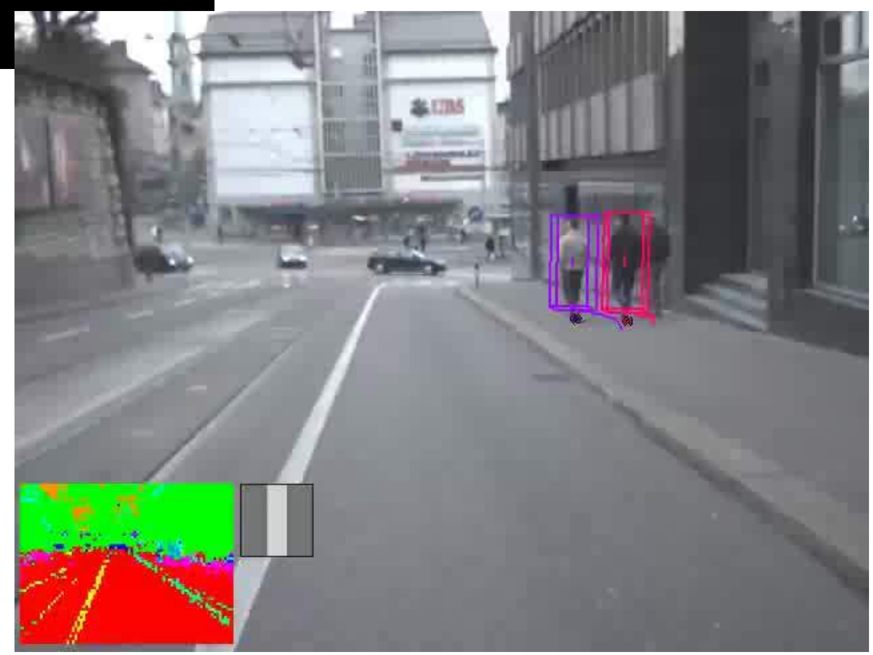
Where could it be useful, in real-life application/engineering scenarios?

Task: "List <u>applications</u> you can think of on a piece of paper"

Discuss in groups of 3-4

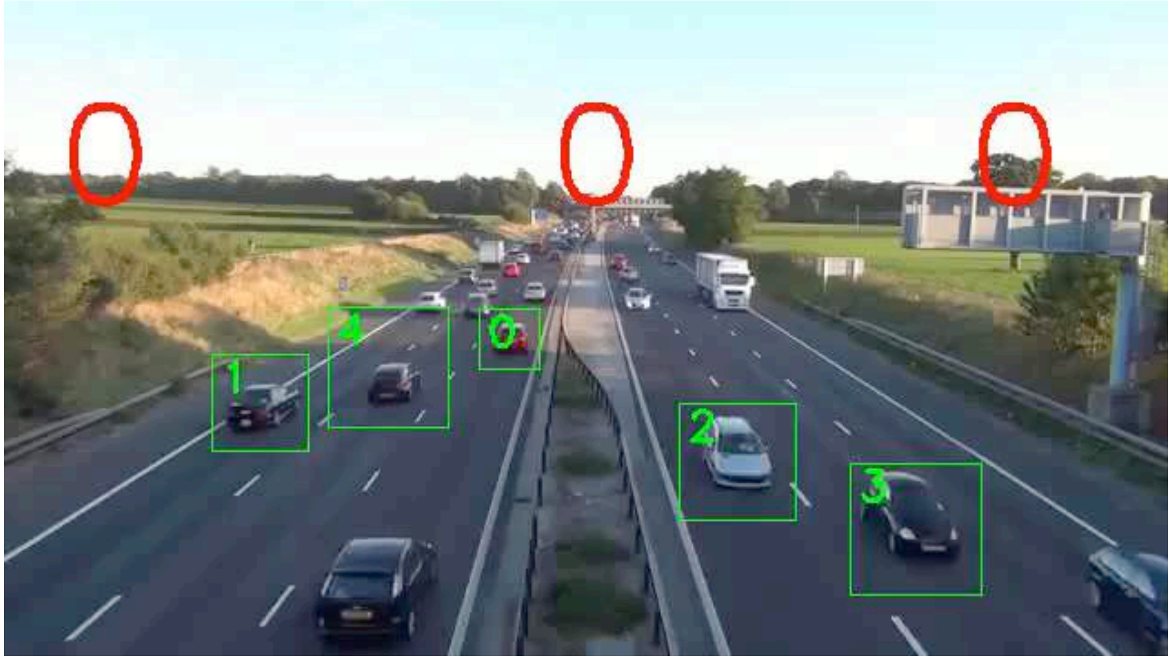
Computer Vision

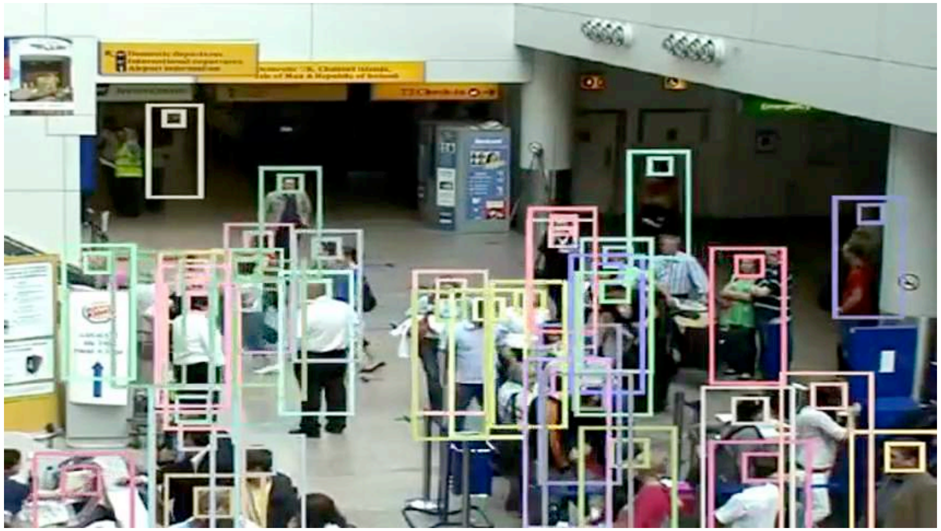
Autonomous Driving



Computer Vision

Surveillance, Safety, Security



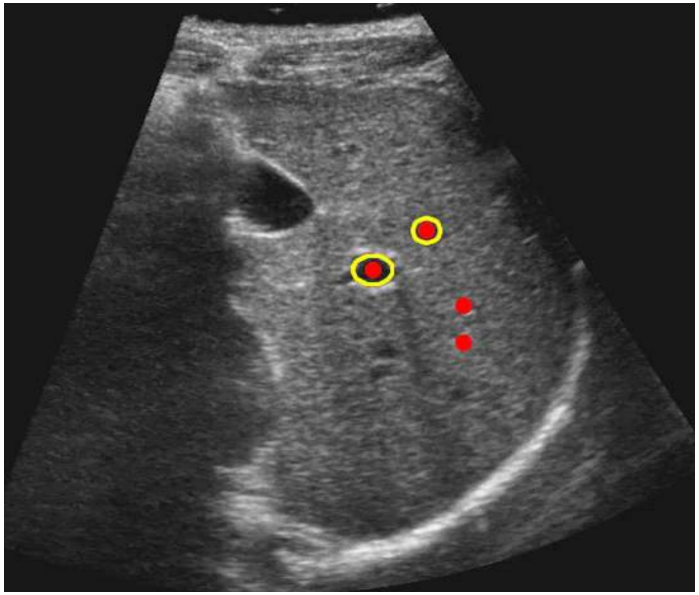


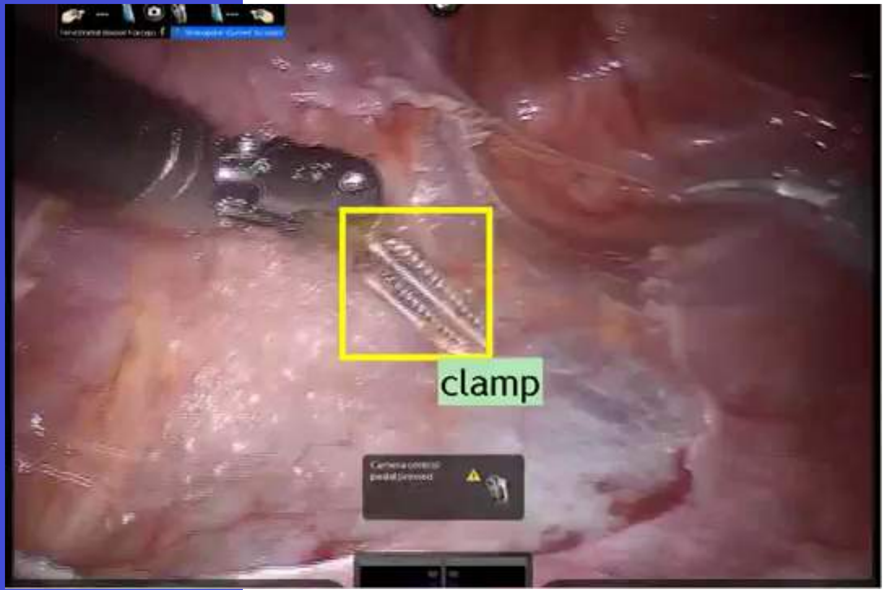
Applications: VR/AR glasses

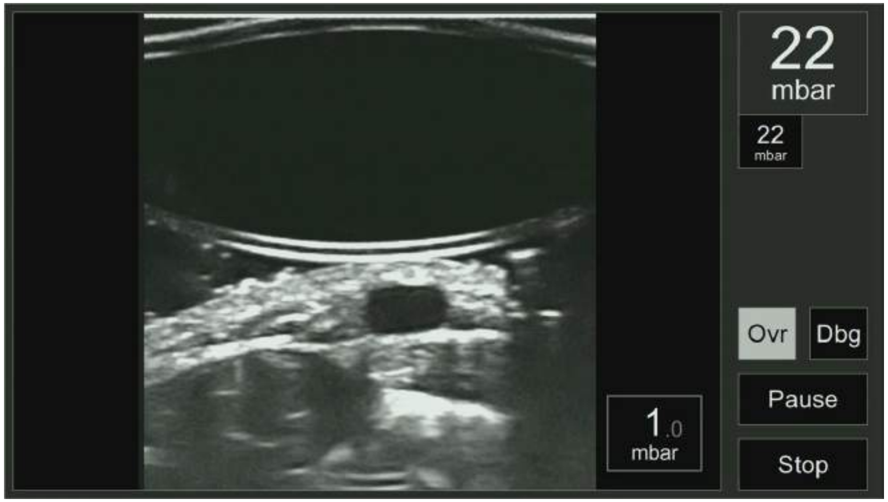
Microsoft HoloLens

Computer Vision

Medical Guidance

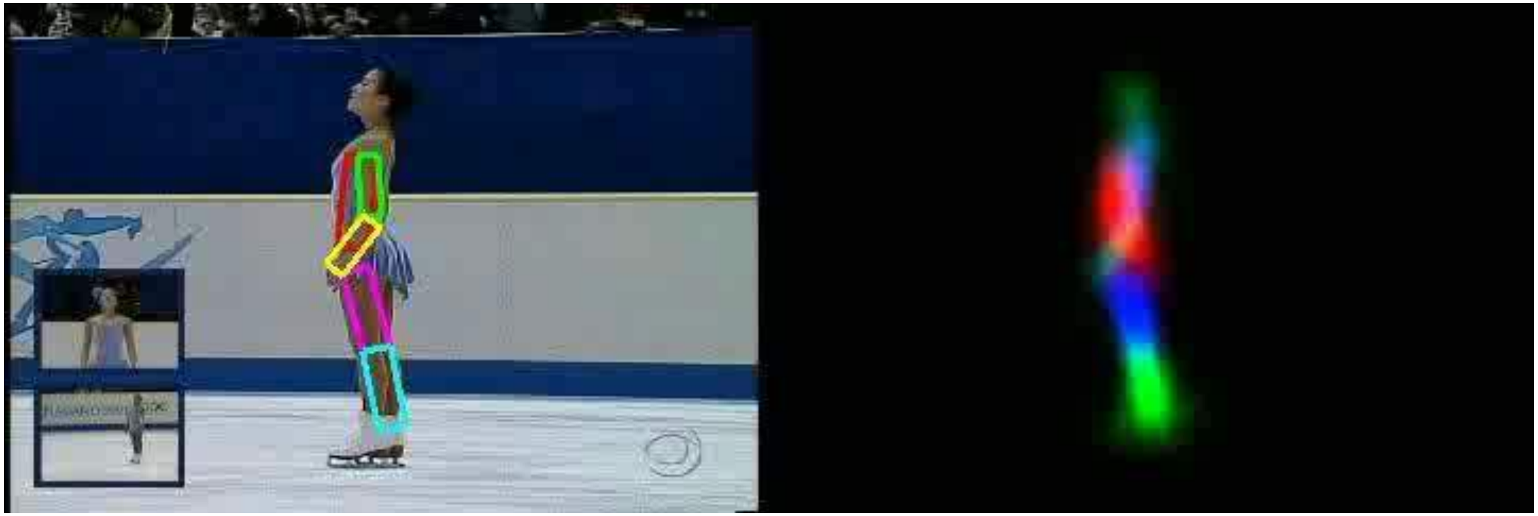






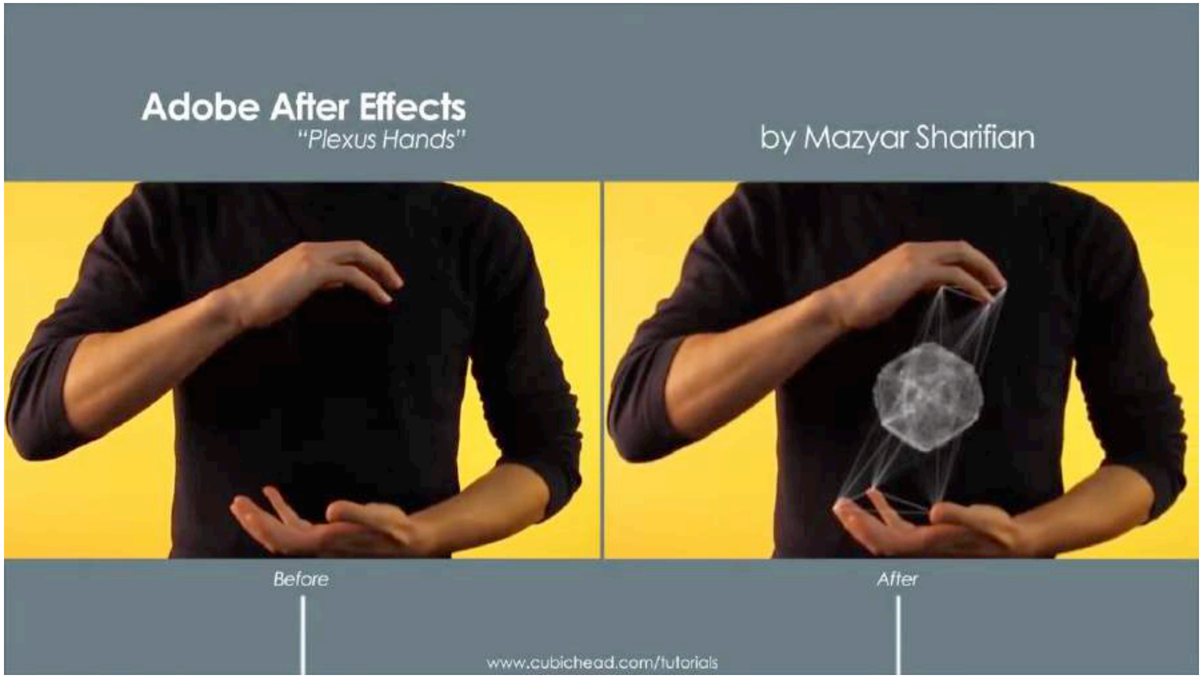
Computer Vision

Sports



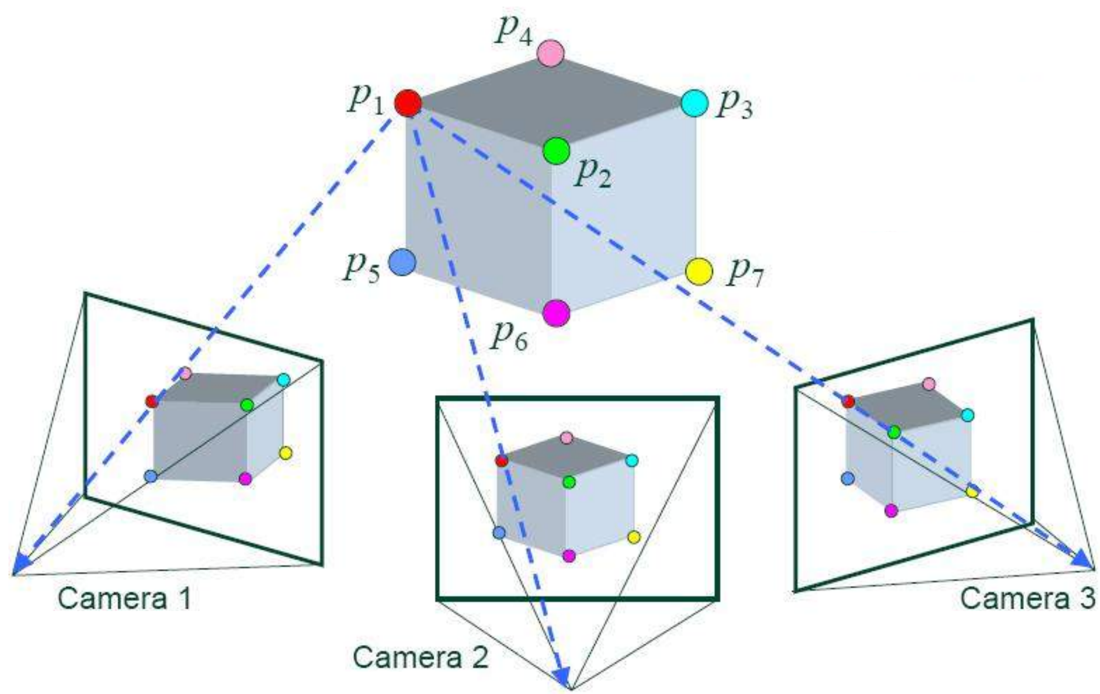
Computer Vision

Video Editing

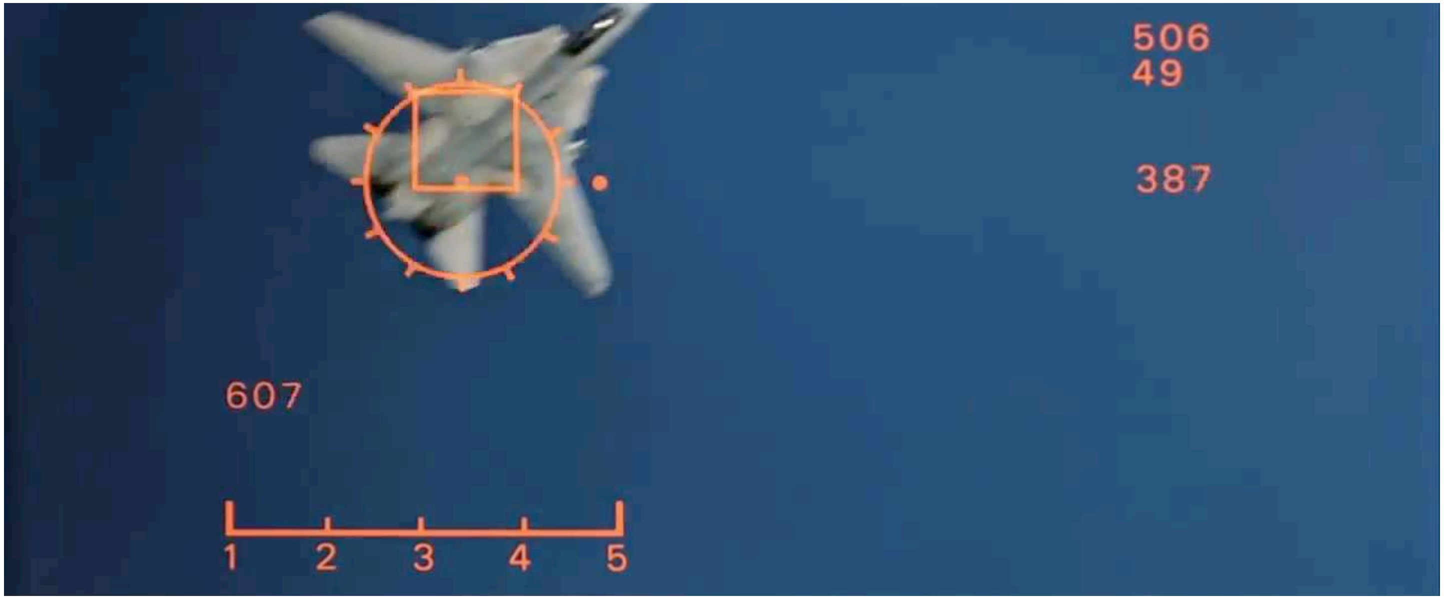


SfM: Structure from Motion

- Tracked Points gives correspondences

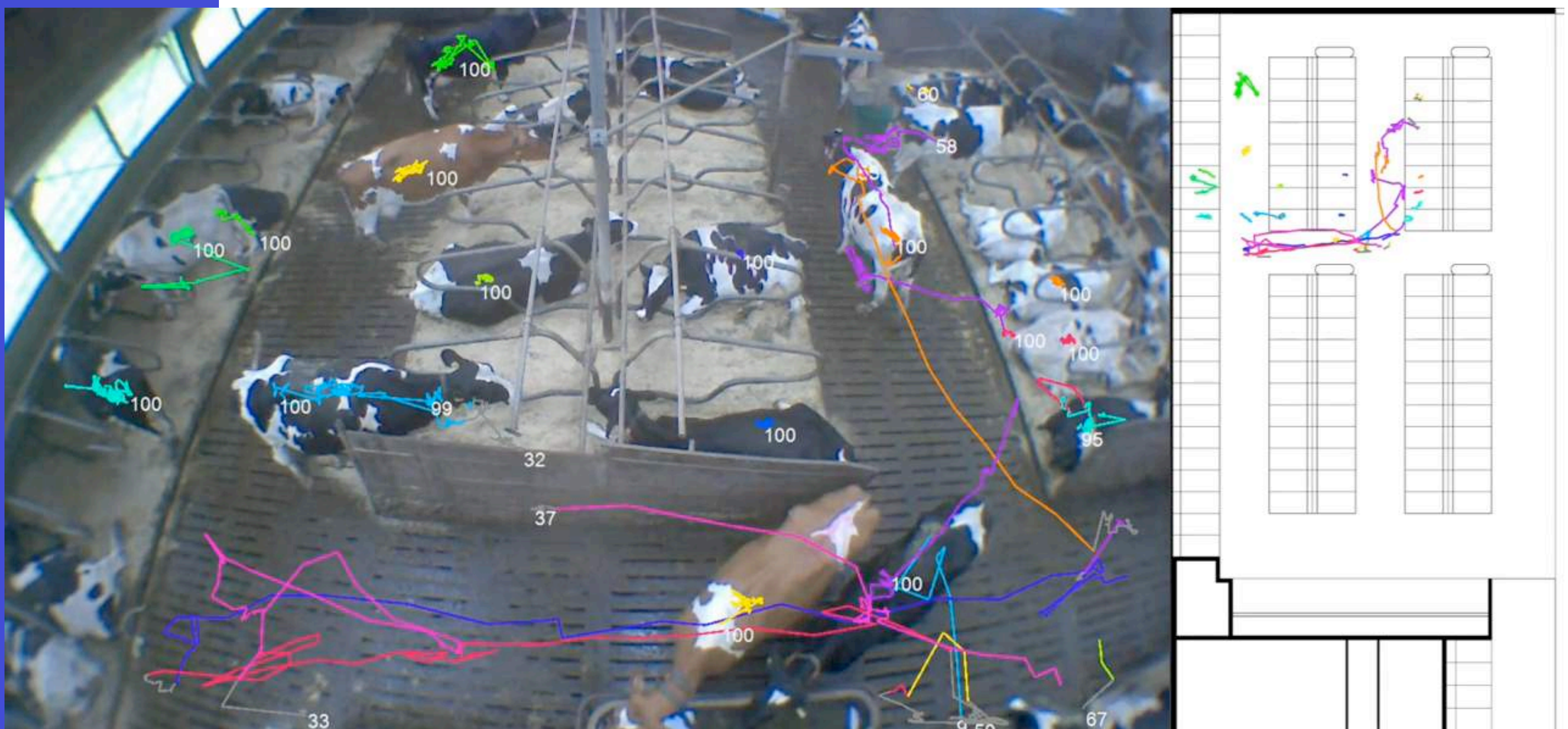


Defense



"Top Gun"

Of course, “very importantly” The Cow Tracker



Applications

- Structure-from-Motion
- Autonomous Driving
- Gesture/Action Recognition
- Augmented Reality
- Navigation
- Safety and Security
- Medical Targeting / Guidance
- Motion Compensation
- …

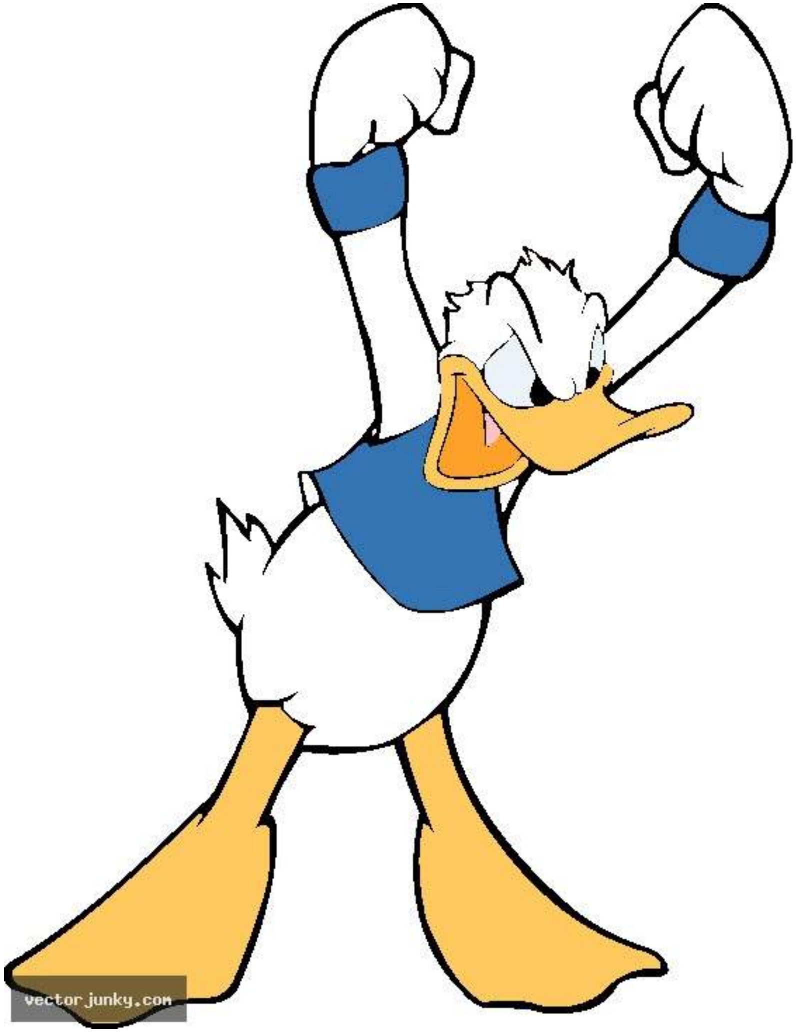
You will be able to:

1. Determine applications of tracking and identify problems solvable by tracking
2. Analyze what methods could work in a practical scenario / situation
3. Assess potential limitations / pitfalls of particular approaches and scenarios
4. **Propose an optimal tracking solution**

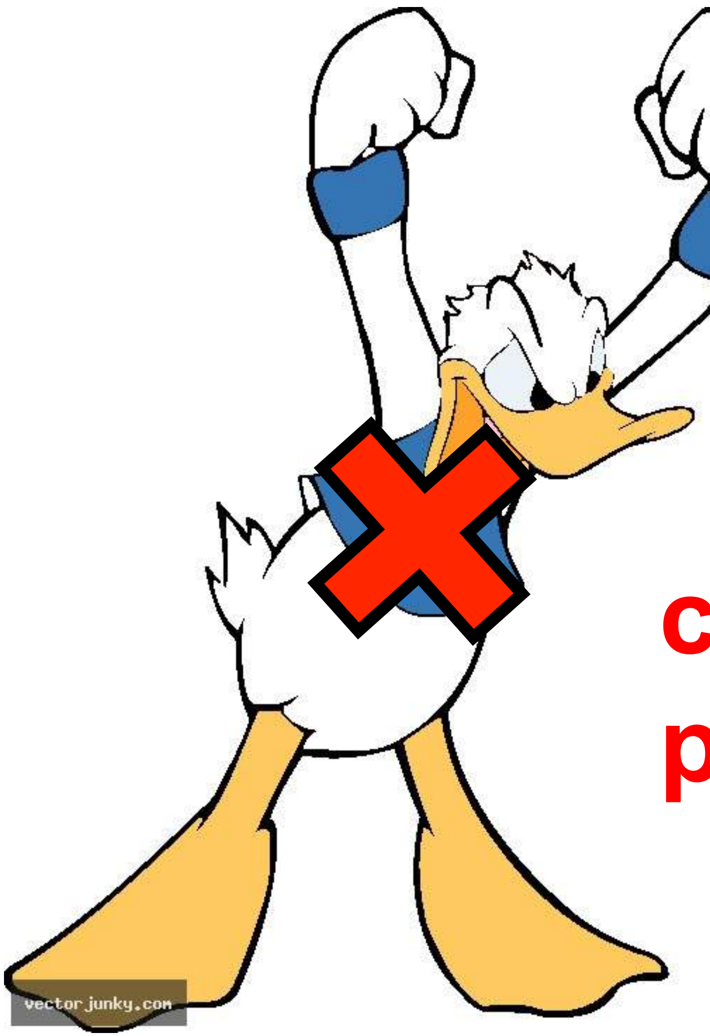
How will we get there:

- (some) common tracking methods
- Few particular keywords & implementation
- What not: details of all individual implementations; cf. "how to google"

What to track?



What to track?

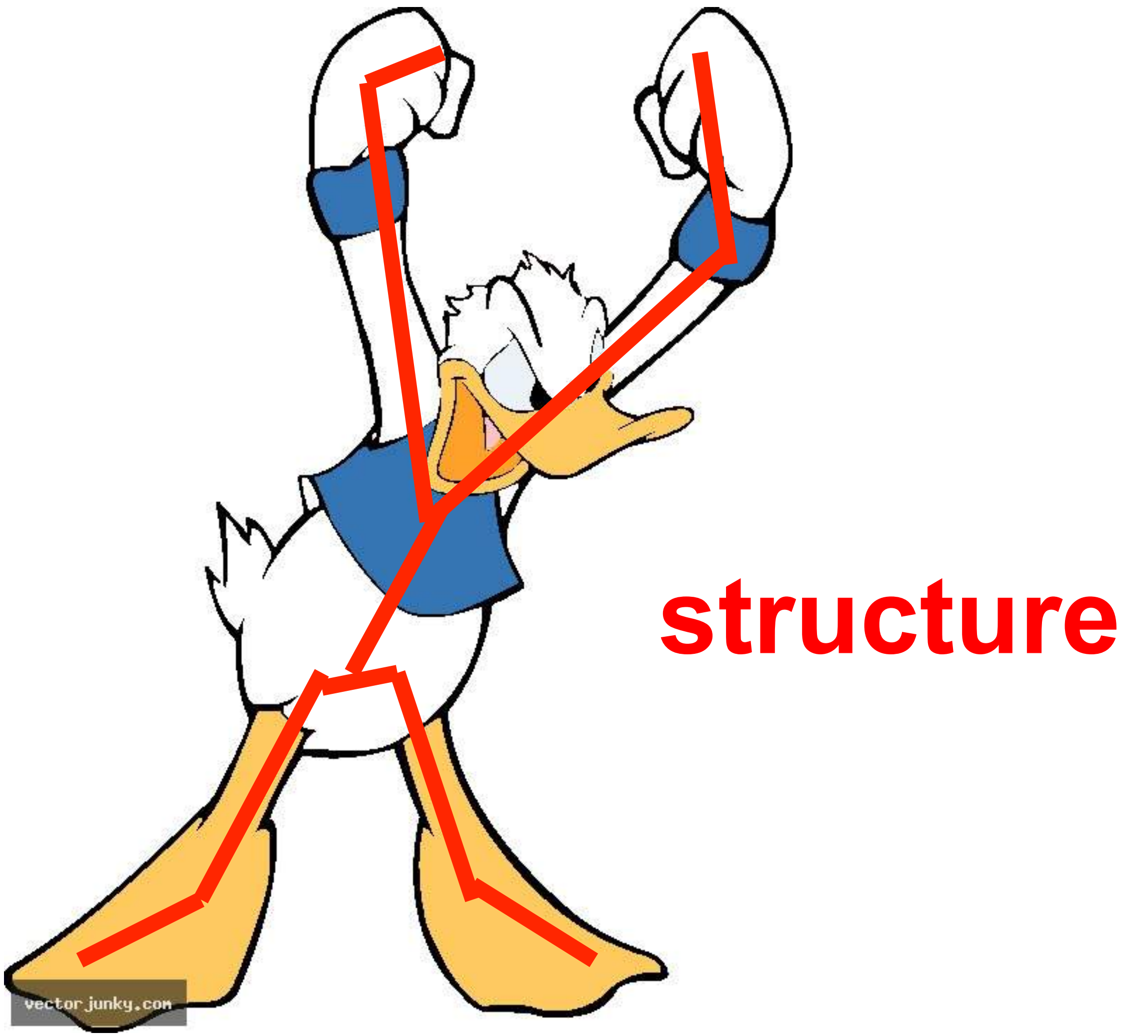


center point

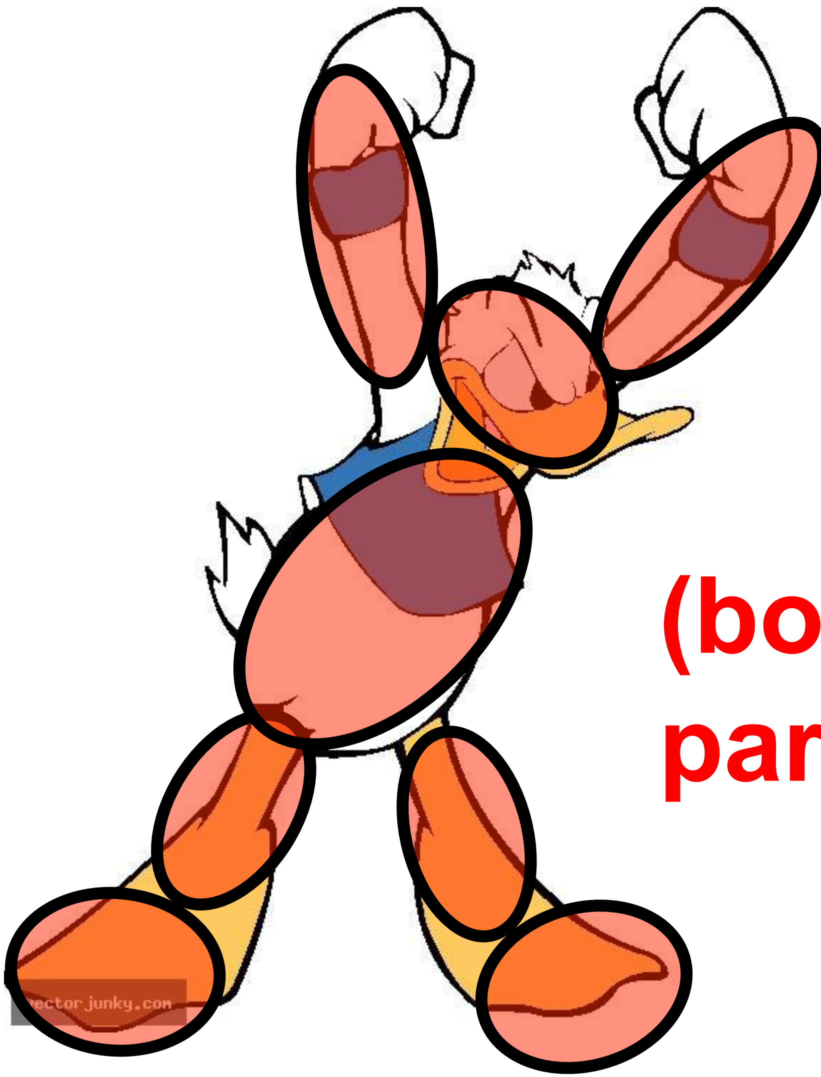
What to track?

multiple points

What to track?



What to track?



(body) parts

Computer Vision

What to track?

What to track?

Approaches

(i) Feature tracking

generic

corners, blob/contours, regions, …

(ii) Model-based tracking

application-specific

face, human body, …

Tracking Requirements

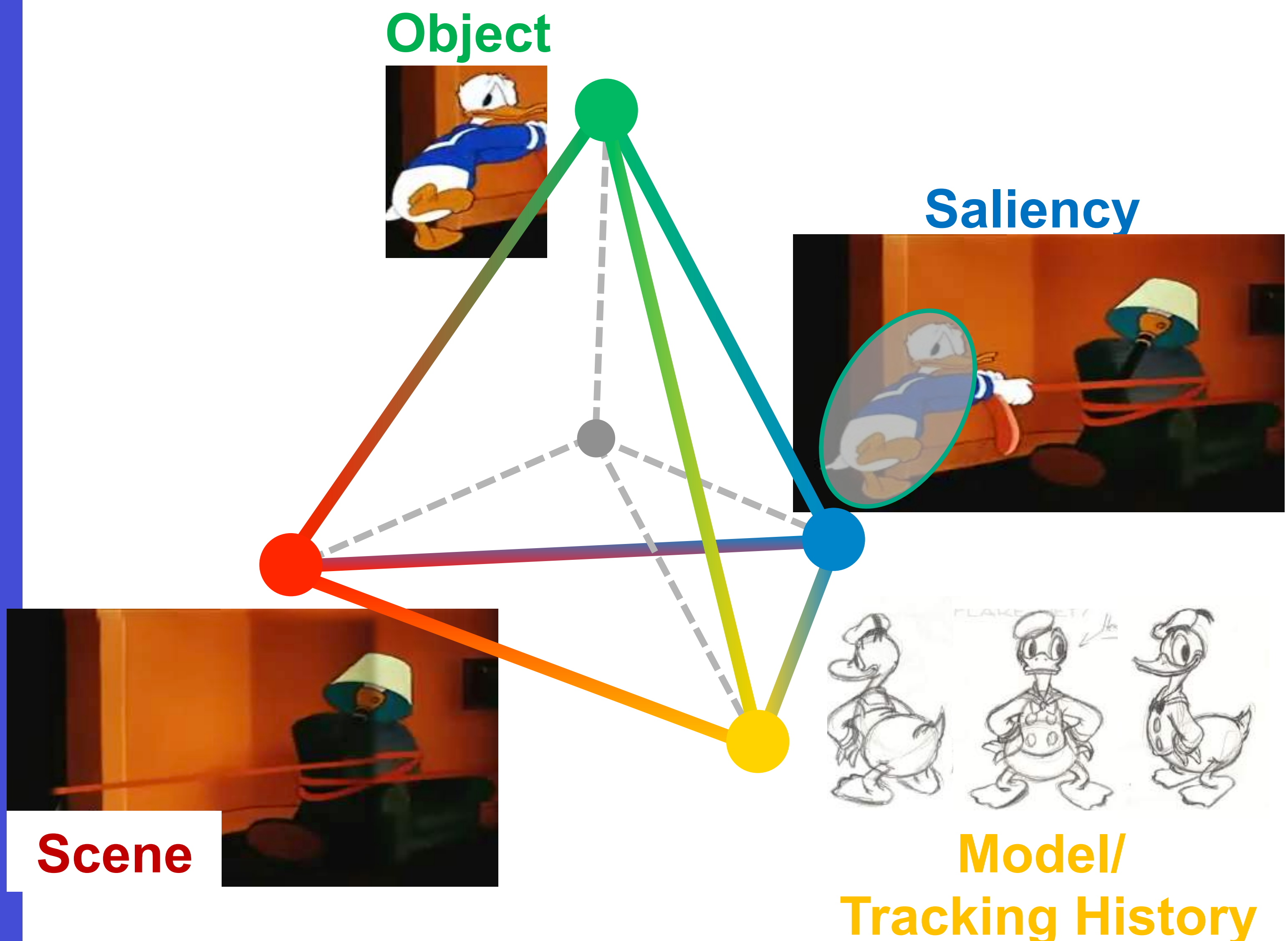
- Strongly depends on the **application**!

Robust, Accurate, Fast,…

- Constrain the tracking task!

Information about the object, dynamics, …

Tracking Cues



Motion as a Cue

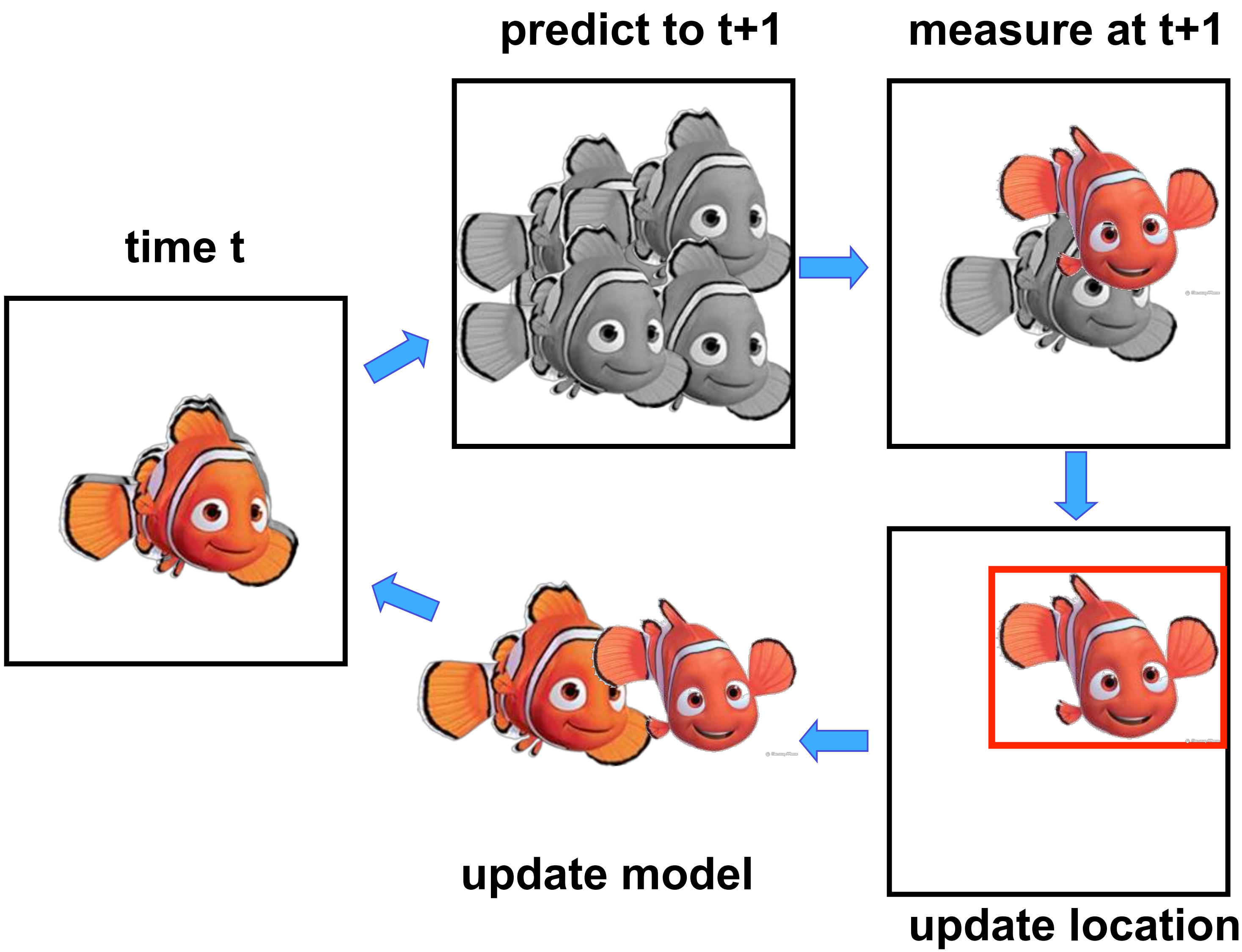
Computer Vision

Motion as a Cue

Motion as a Cue

- Eye perceptive to temporal changes (gradients)
- “Event based camera”

General Tracking Loop



Which strategy to use?

Depends, No single solution

Some rule-of-thumb suggestions:

- If you can alter the “object” to be tracked,

➔ <u>modify/add tracking info</u>

e.g. optical IR markers, mark with patterns, etc

- If object is fixed/known, but modification not possible/ desired ➔ <u>Utilize known info</u>

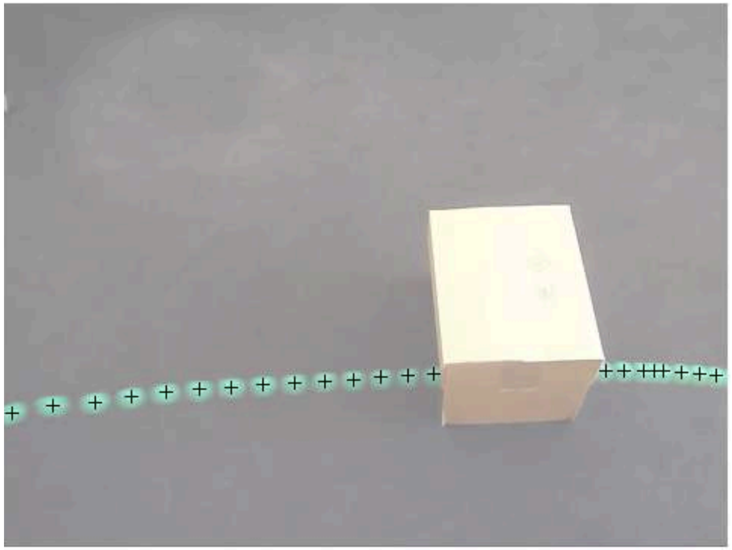
  e.g. use a template image and/or known object features

- If object unknown/variable object, but resides in a known (static) environment ➔ utilize this!
- If none above, simply follow from initial image/location

Tracking v.s. segmentation/localization:
Key difference is TEMPORAL consistency

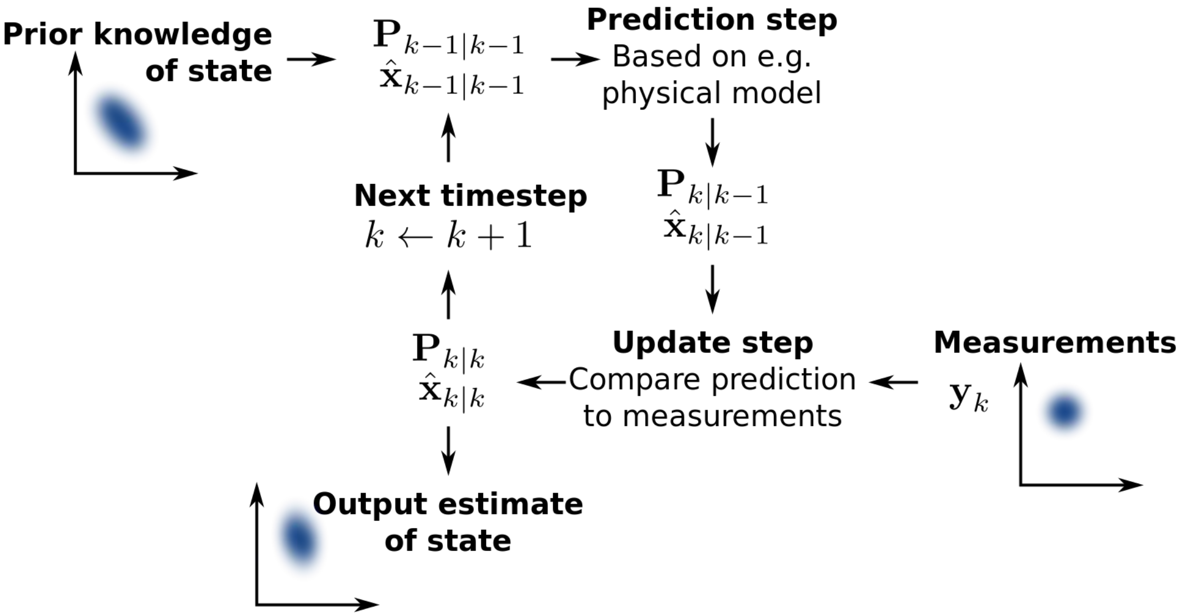
Trajectory
(Temporal Filtering)

Temporal Filtering/Predictions

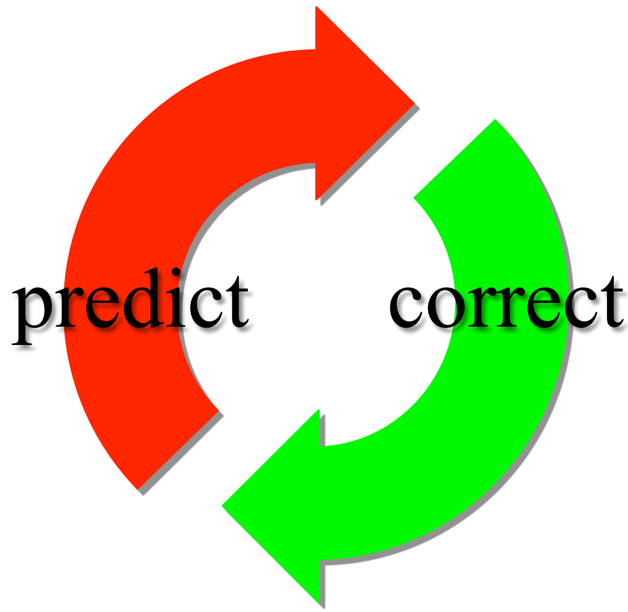


- To predict location
- To reduce noise
- To disambiguate multiple objects

<u>Kalman Filtering</u>

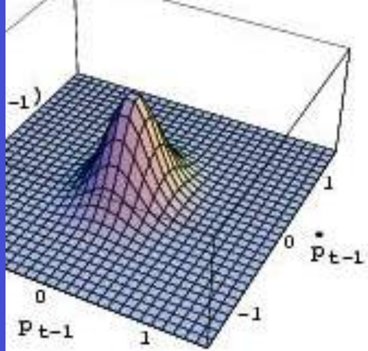


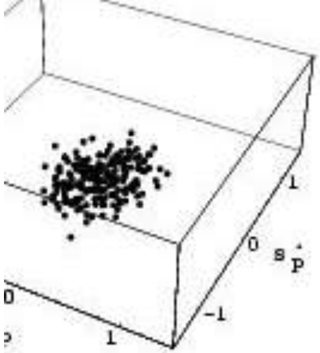
Steps of Tracking



- Recap: Particle filtering
  - Tracking can be seen as the process of propagating the posterior distribution of state given measurements across time.

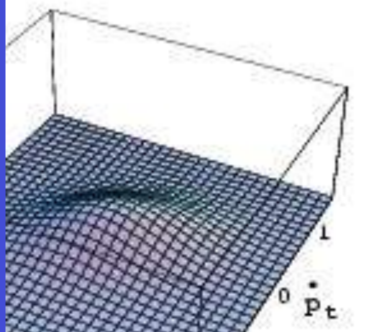
Particle Filter

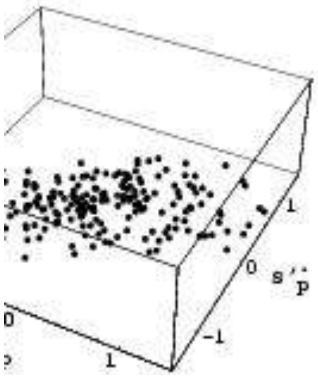




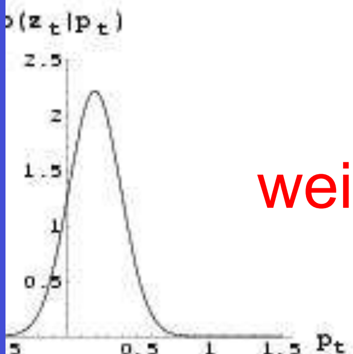
$$p(p_{t-1}, \dot{p}_{t-1} \mid z_{t-1})$$

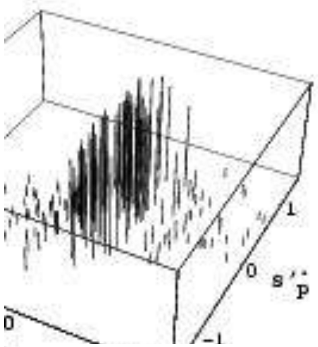
prediction





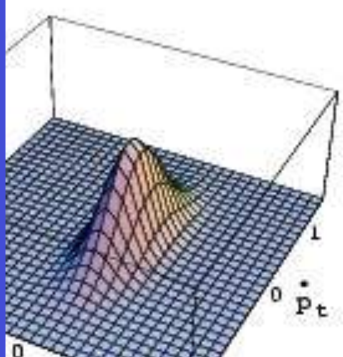
$$p(p_t, \dot{p}_t \mid z_{t-1})$$

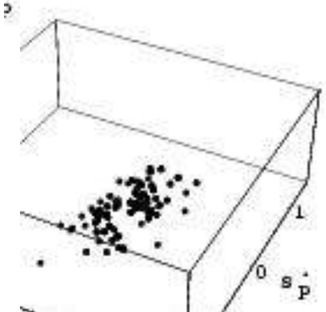




weighing with $\boxed{p(z_t \mid p_t)}$

update





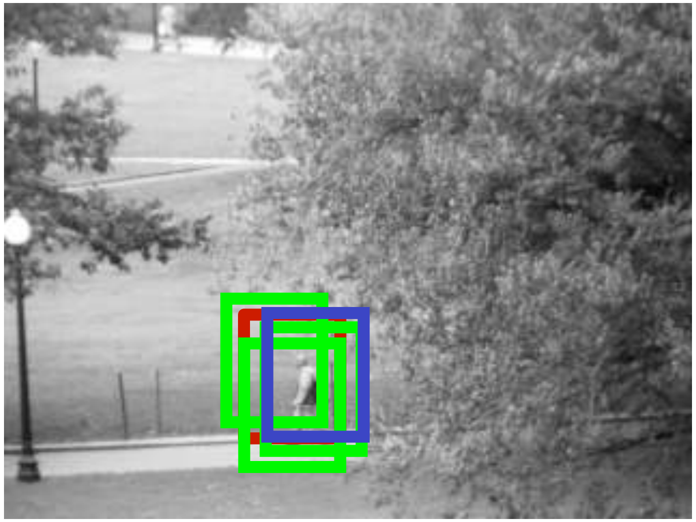
$$p(p_t, \dot{p}_t \mid z_t)$$

CONDENSATION

Traditional/Simple Tracking

t=1

initialization



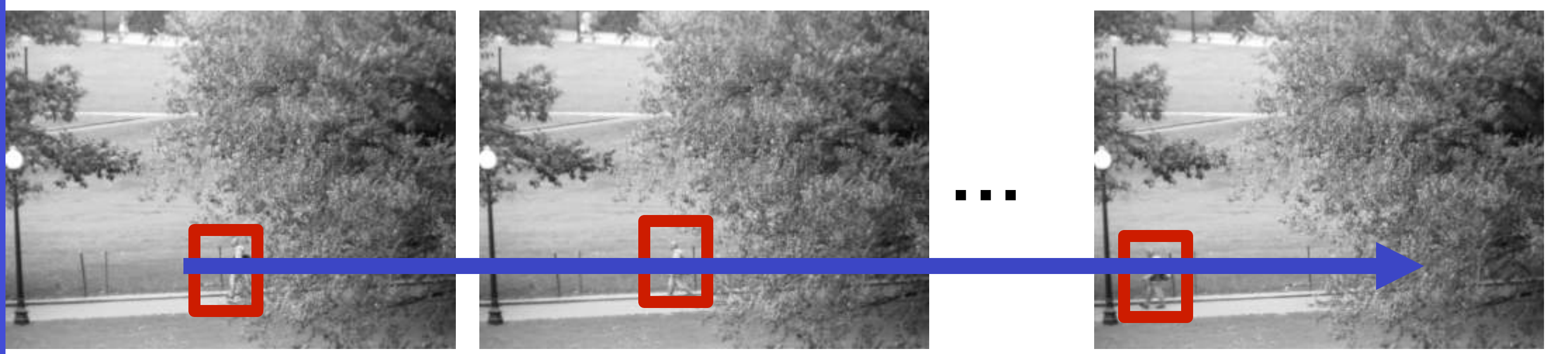
t=2

position in prev. frame

candidate new positions
(e.g., dynamics)

best new position
(e.g., max color similarity)

Tracking-by-Detection



detect object(s) independently in each frame

associate detections over time into *tracks*

Outline

Feature

- Region Tracking
- Point Tracking
- Template Tracking

Model

- Tracking-by-Detection
  - of a specific target
  - of the object class
- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)

Region Tracking

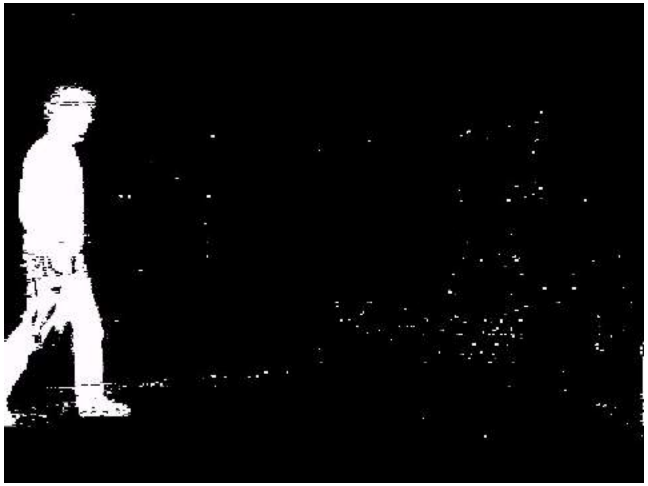
(and Mean Shift Algorithm)

Background Modeling

Input

-

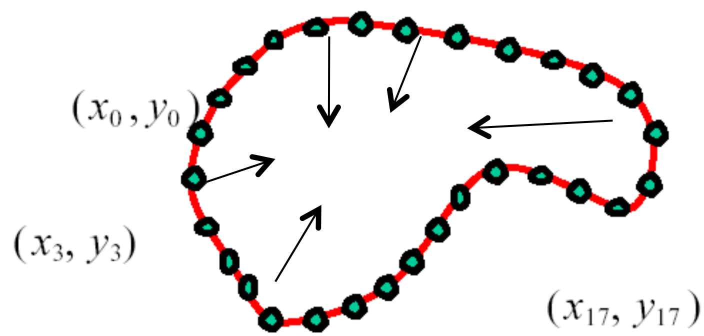
Background Model

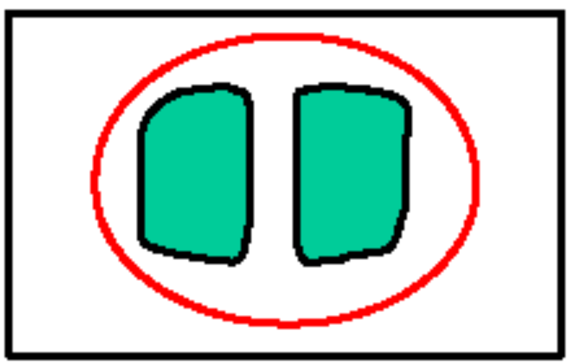


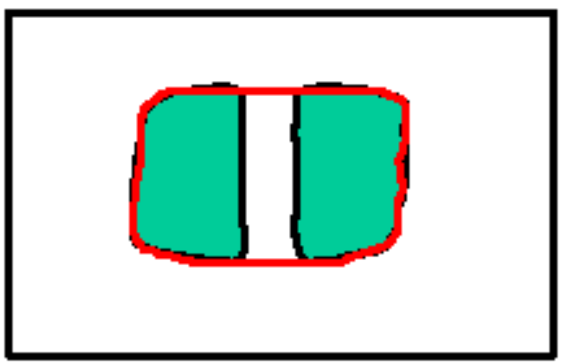
Moving Foreground Blobs (Objects)

Deformable models

- One option: Fit deformable curves





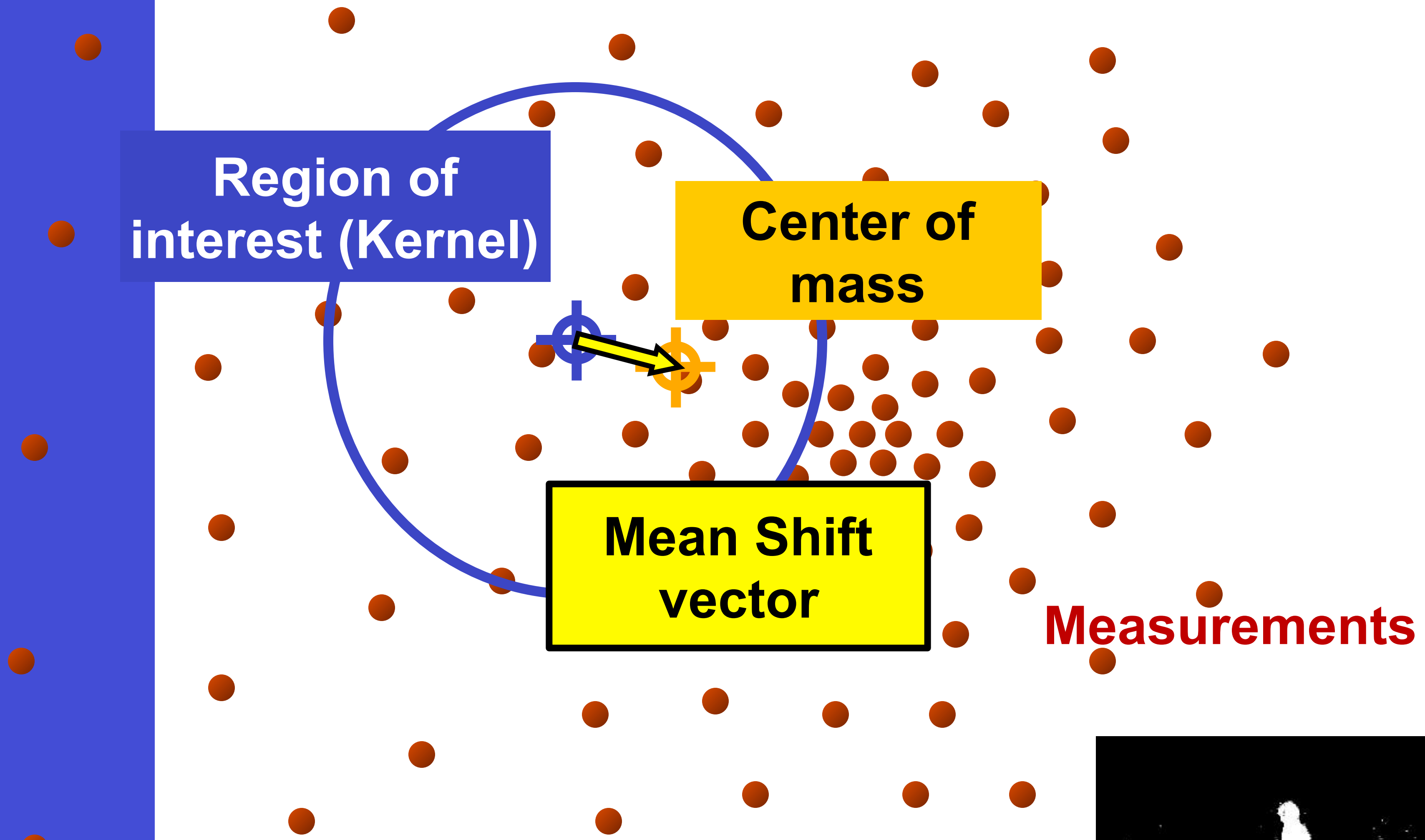


Mean Shift Tracking

- The mean shift tracker tracks a region, with a prescribed (color) distribution
- The similarity between the tracked region and the target region is maximized, through evolution towards higher density in a parameter space
- Typically this search only takes a few iterations

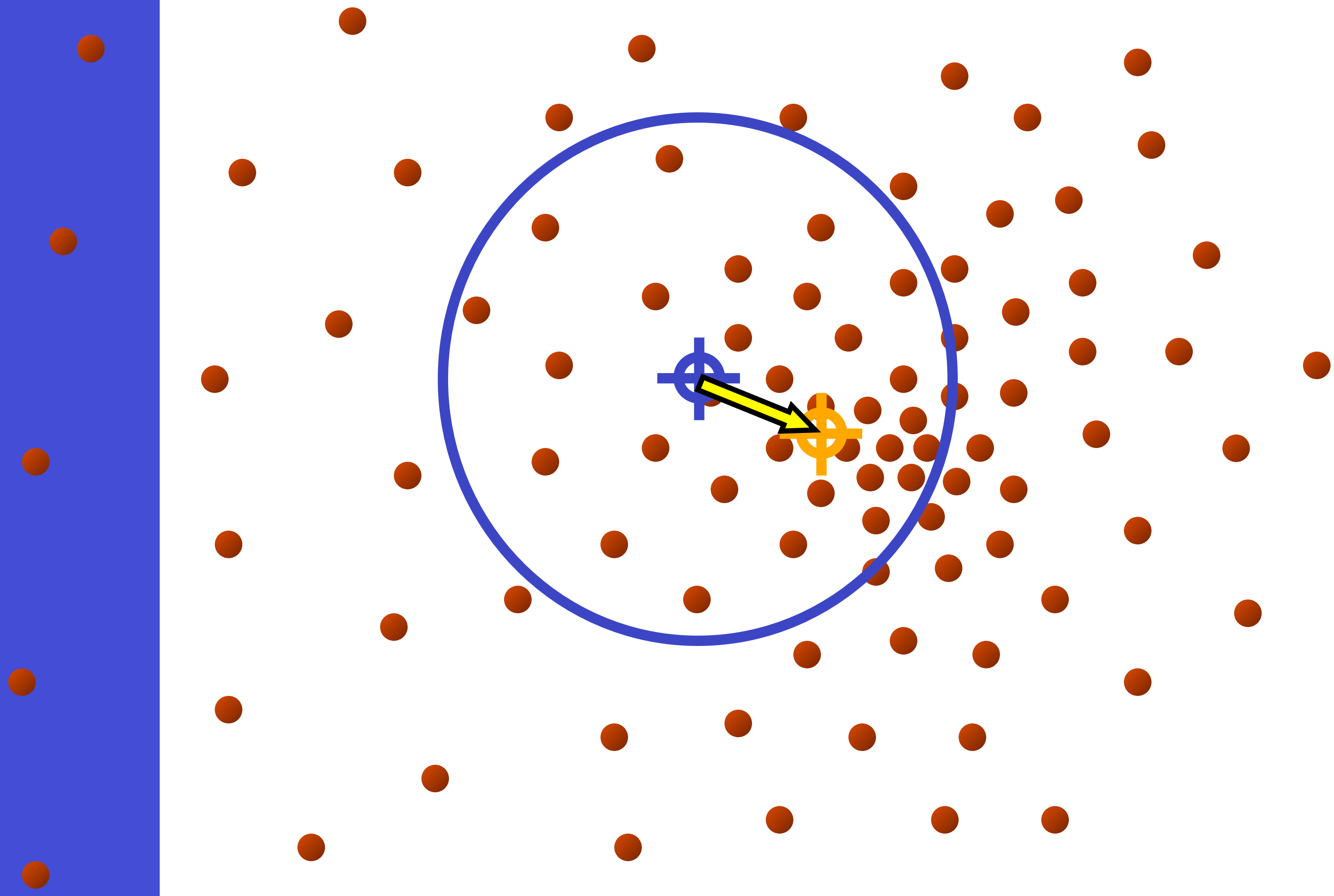
[Comaniciu and Meer, ICCV'99]

Meanshift Tracking



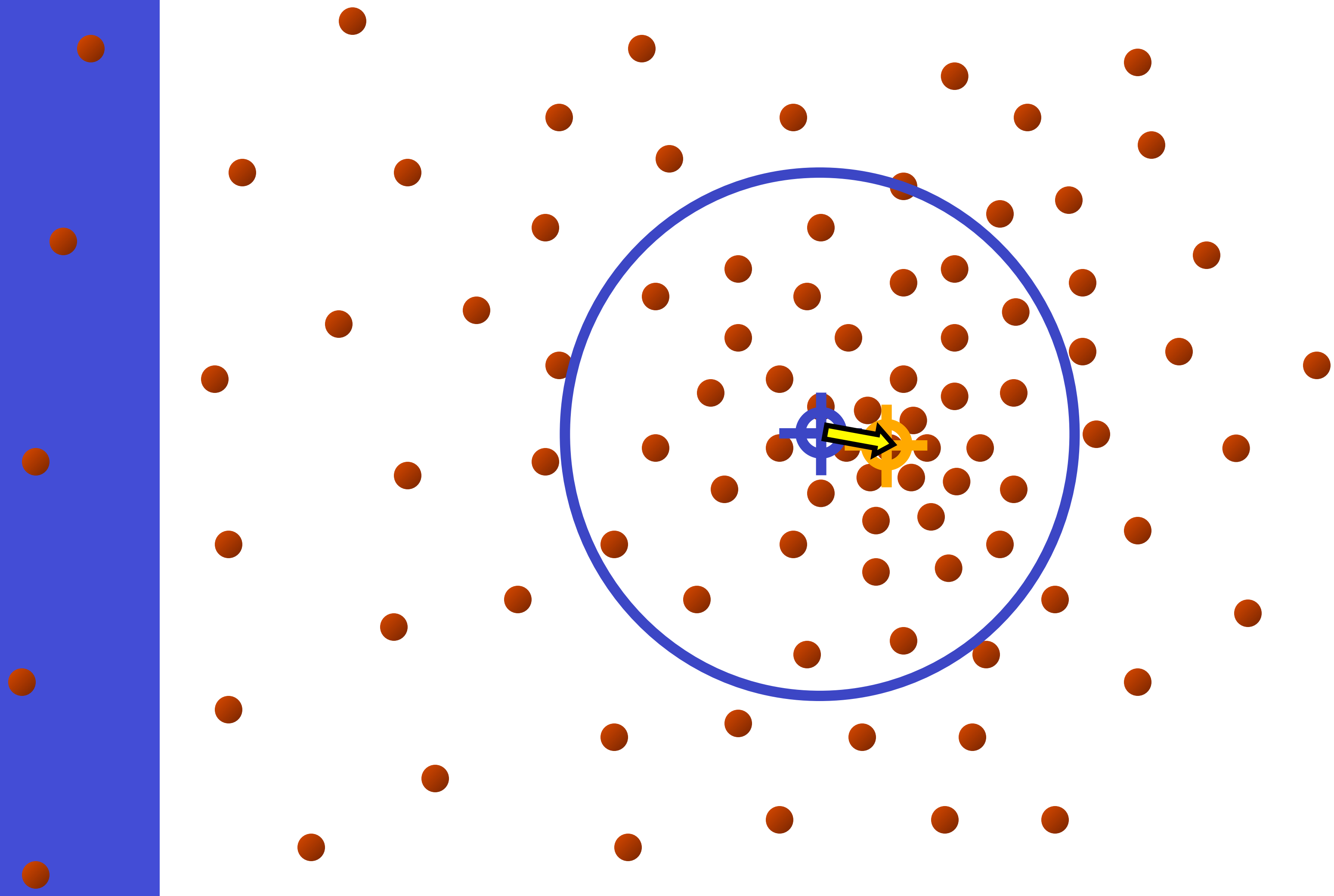
Computer Vision

Intuitive Description

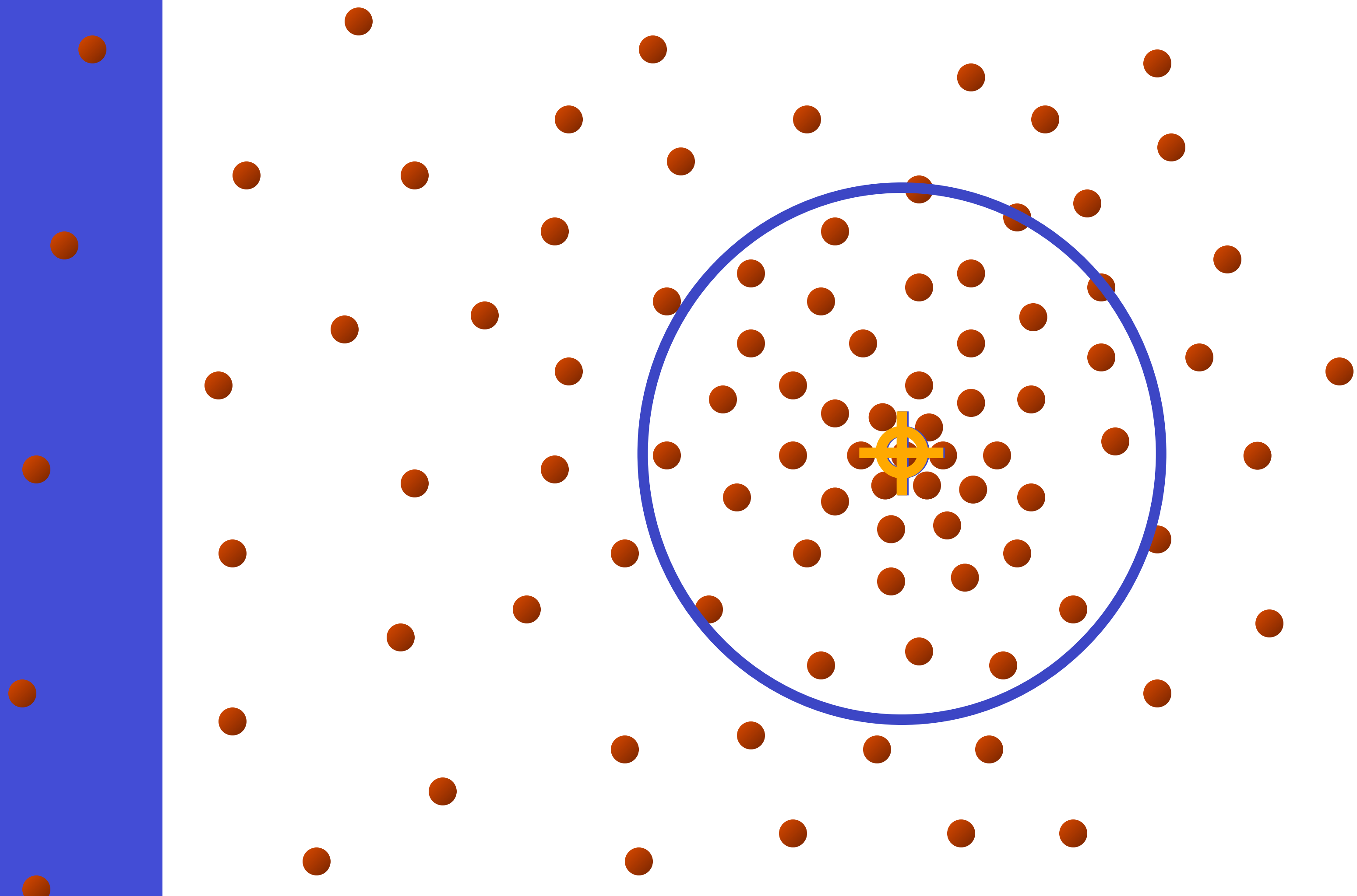


Computer Vision

Intuitive Description

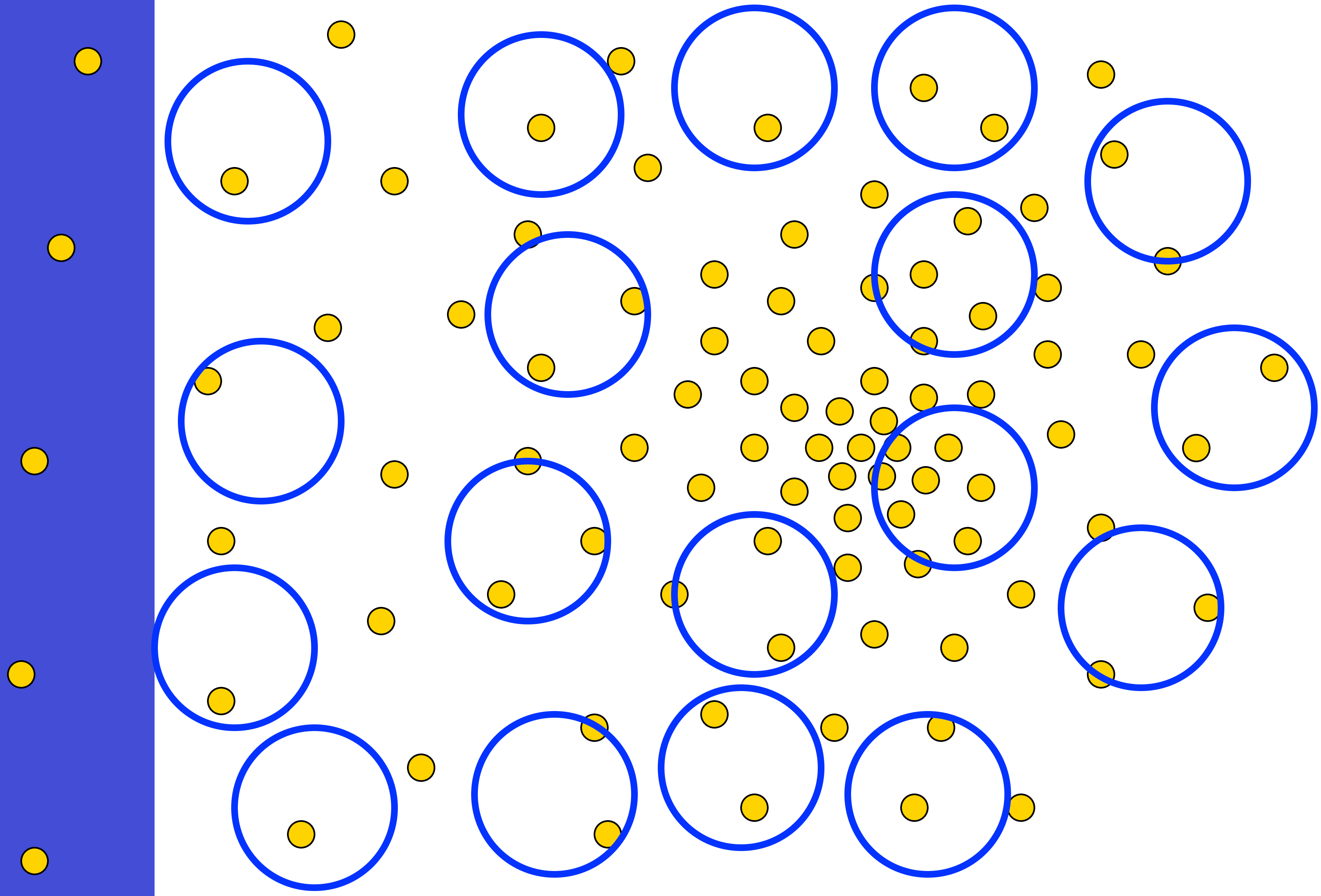


Intuitive Description



Computer Vision

Intuitive Desciption



Computer Vision

Example: Safety Monitoring

Outline

Feature

- Region Tracking
- Point Tracking
- Template Tracking

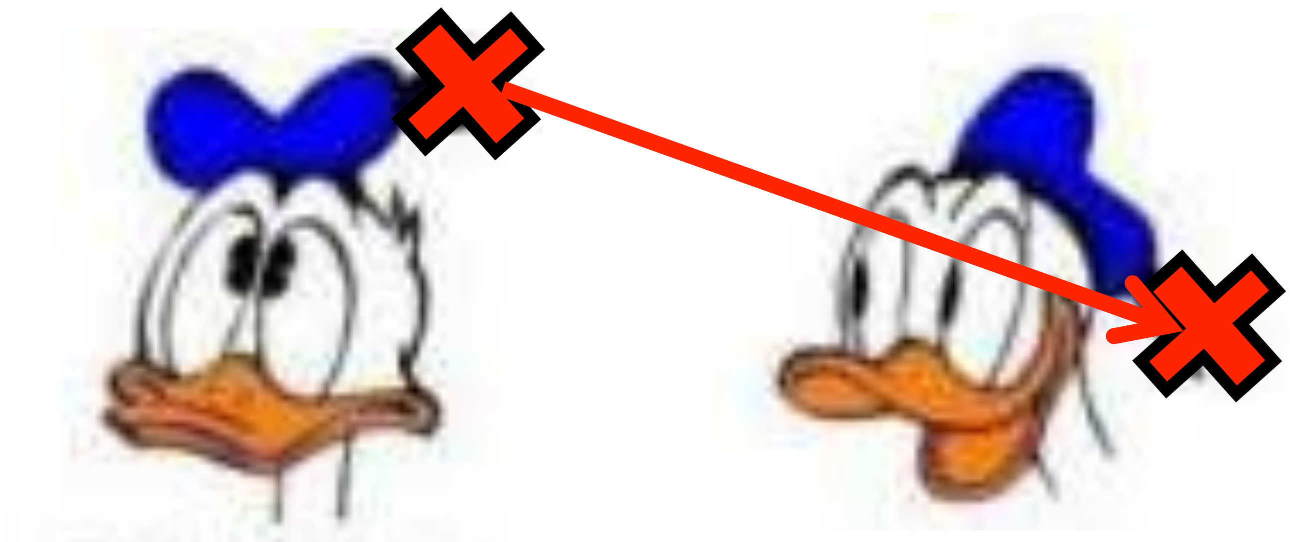
Model

- Tracking-by-Detection
  - of a specific target
  - of the object class
- Model-based Body Articulation
- On-line Learning

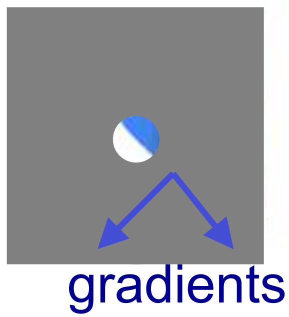
- Misc (preventing drift, context, issues)

Point Tracking
(and Aperture Problem)

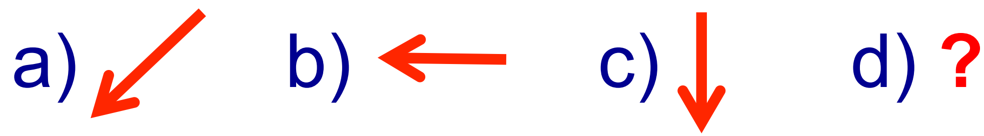
Estimate Optimal Transformation



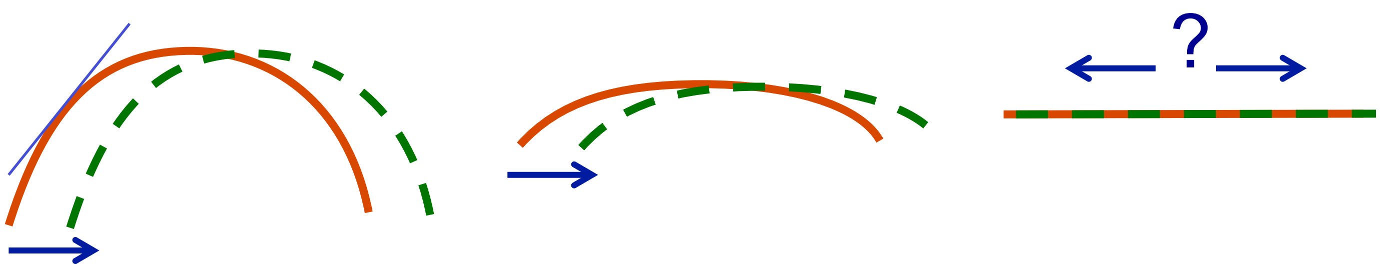
When can we estimate motion?



Q1. Which direction is the pattern behind the circular hole moving?

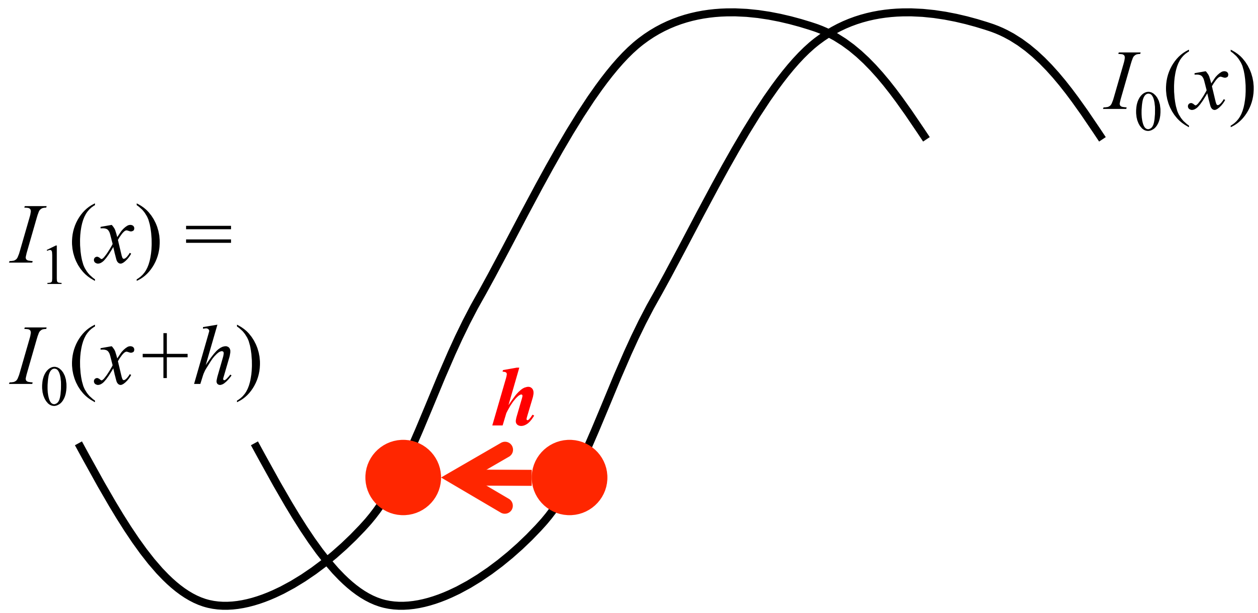


Q2. 1D motion: We cannot determine the direction of motion from red to green line on the right. Why not?



Q3. Any similarity/connection between Q1 & Q2?

Sum of Squared Differences



$$E(h) = [I_0(x+h) - I_1(x)]^2$$

Computer Vision

Displacement

$$E(h) = [\, I_0(x+h) - I_1(x) \,]^2$$

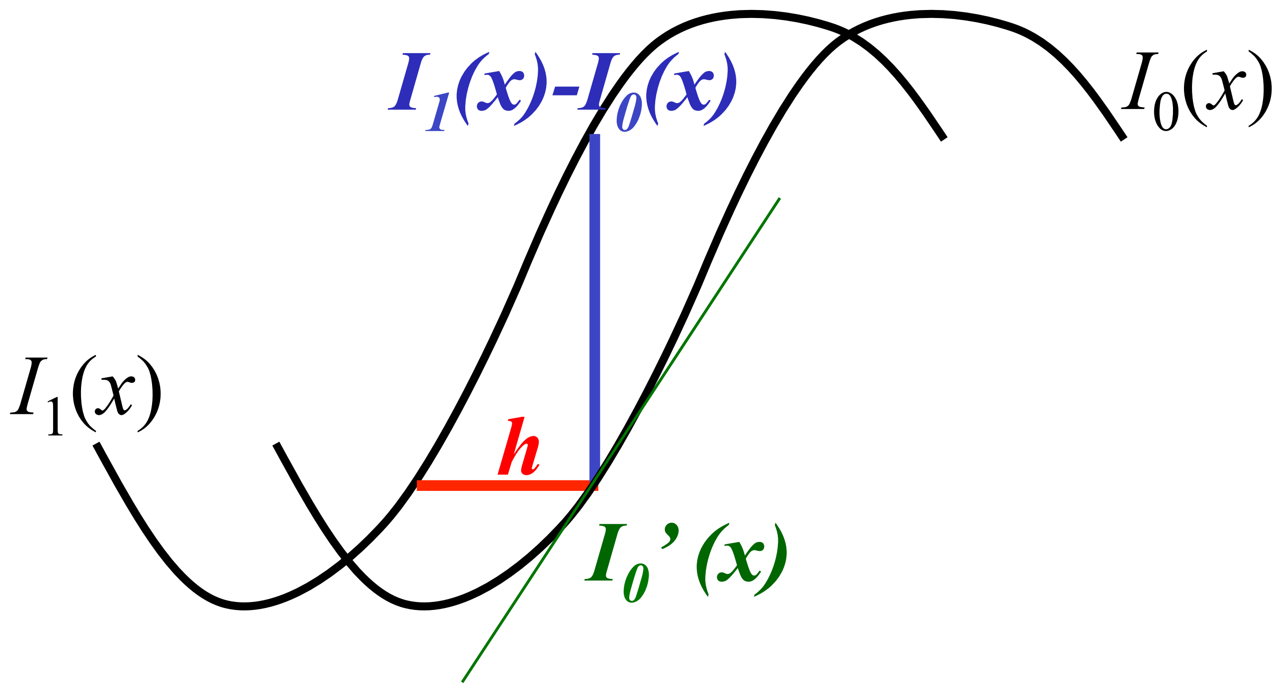
$$E(h) \approx [\, I_0(x) + h I_0'(x) - I_1(x) \,]^2$$

$$\frac{\partial E}{\partial h} \approx 2\, I_0'(x)\, [\, I_0(x) + h I_0'(x) - I_1(x) \,] = 0$$

$$h \approx \frac{I_1(x) - I_0(x)}{I_0'(x)}$$

Computer Vision

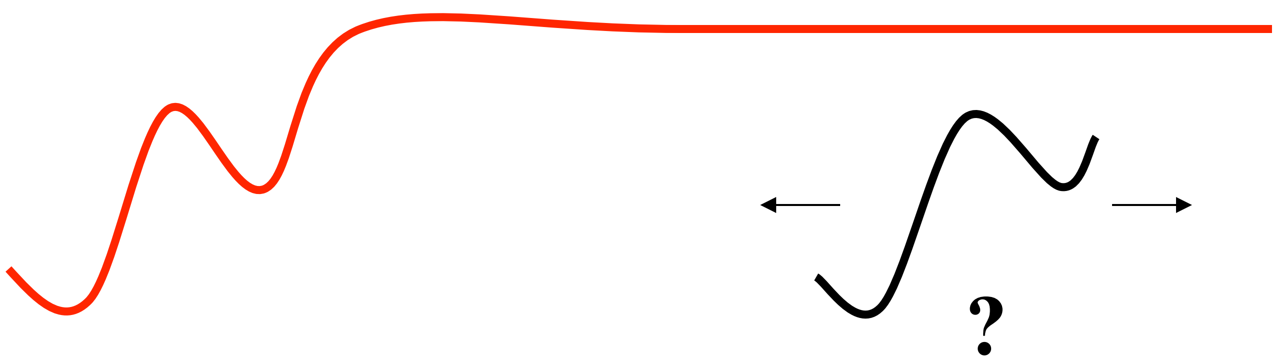
Intuition



$$h \approx \frac{I_1(x) - I_0(x)}{I_0'(x)}$$

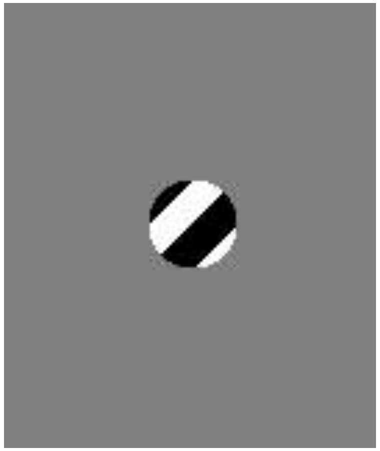
Problem 1: Zero Gradient

$$h \approx \frac{I_1(x) - I_0(x)}{I_0'(x)}$$

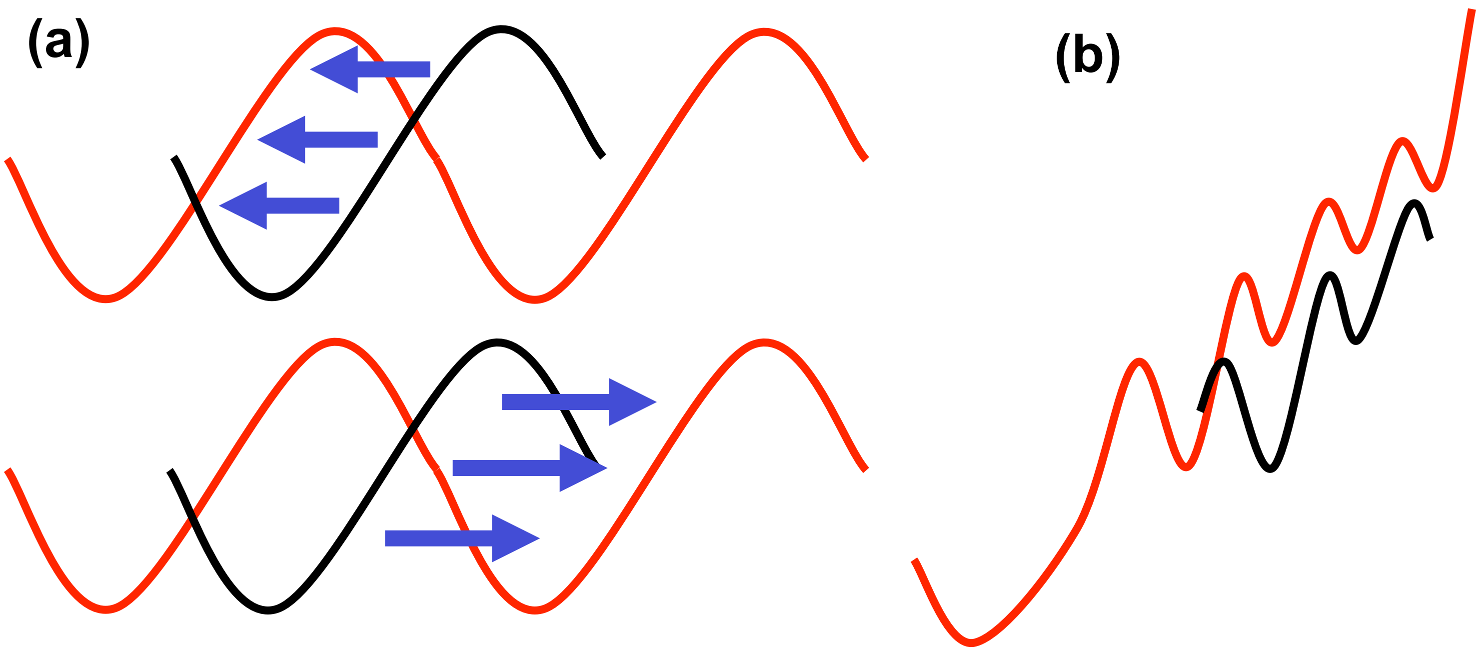


Problem 1: "Aperture problem"

- Tracking needs gradients in all possible directions to be well defined
- If no gradient along one direction, we cannot determine relative motion in that axis

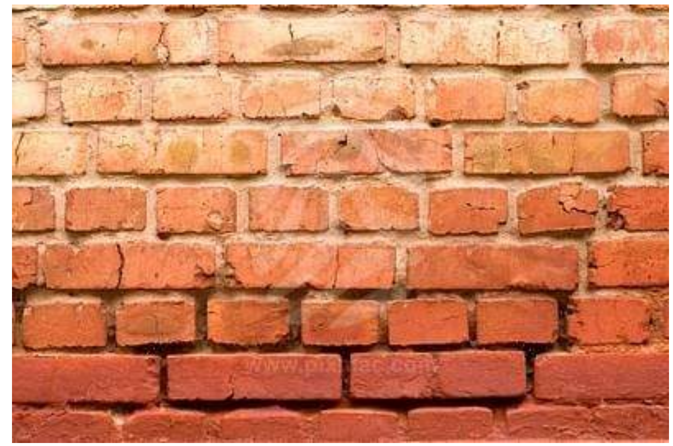


Problem 2: Local Minima

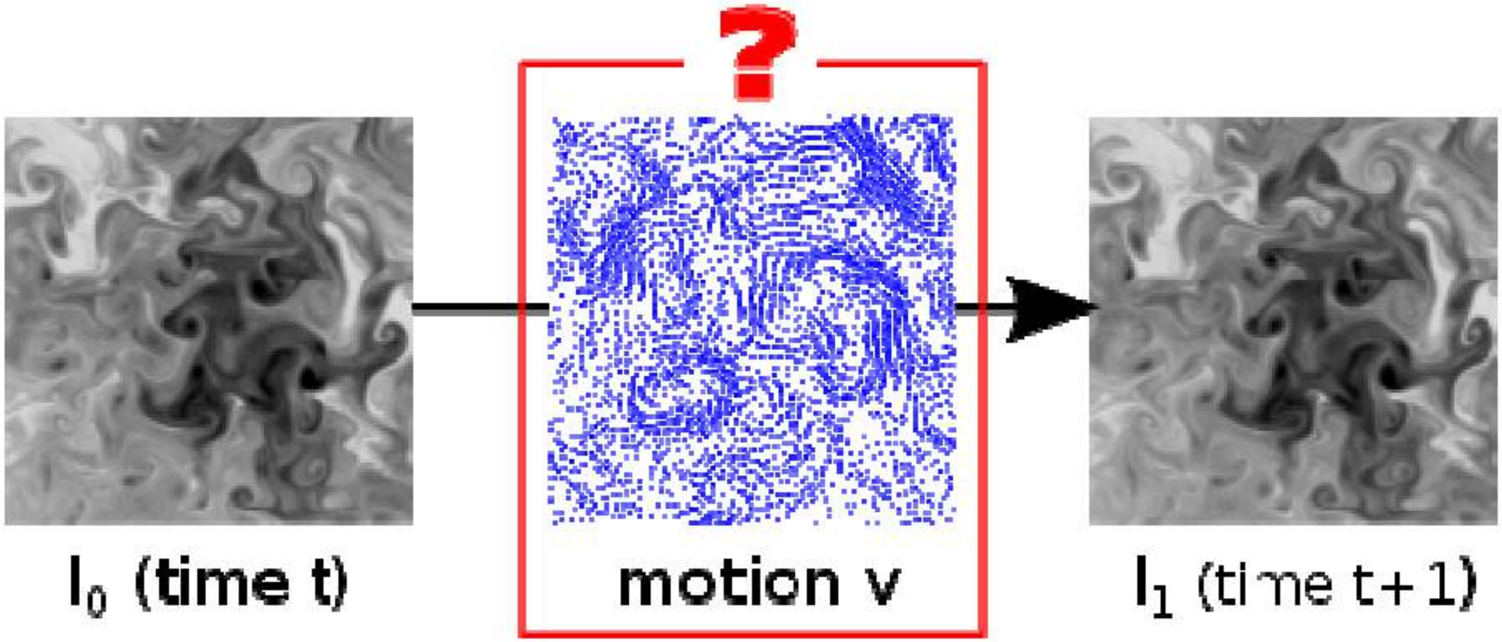


- Motion to closest minimum has to be assumed
- Indirect result: Frame-rate should be faster than motion of half-wavelength (Nyquist rate)
- Nonconvex regions may indicate multiple sols

Problem 2: Local Minima



Recall: Optical Flow



- OF recovers (smooth) motion everywhere
- Least-squares regularization: Horn-Schunk makes smooth spatial change assumption
- In contrast, tracking seeks a single motion!

Recall: Optical Flow

$$\boxed{I_x u + I_y v + I_t = 0}$$

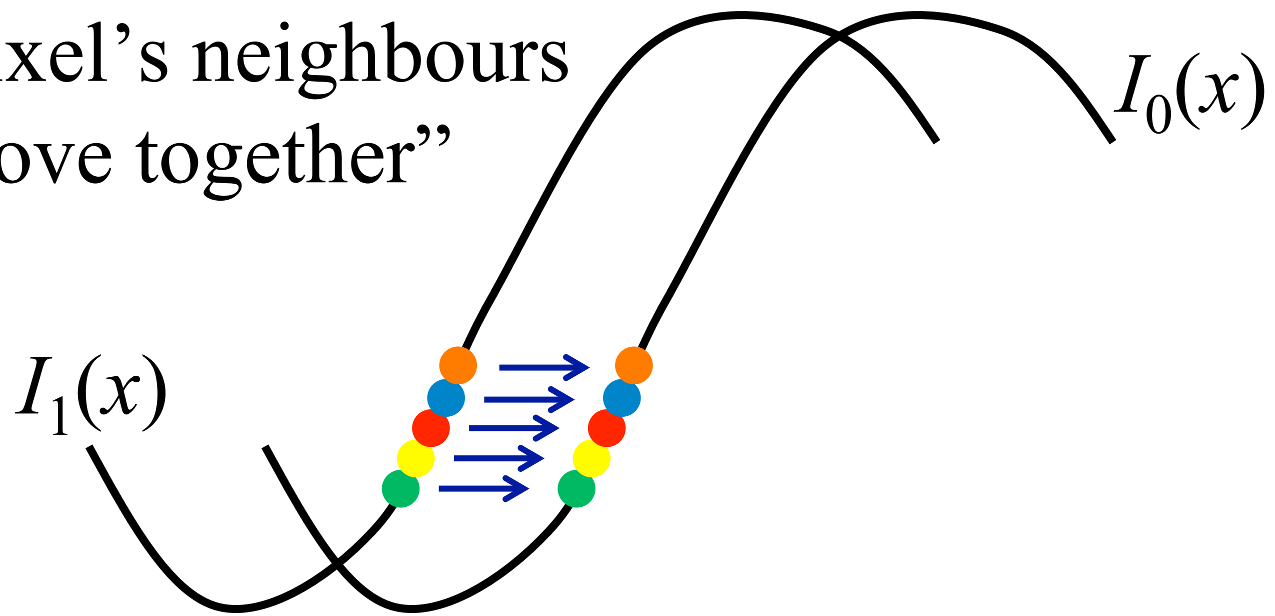
$$I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t}$$

$$u = \frac{dx}{dt}, \quad v = \frac{dy}{dt}$$

1 equation in 2 unknowns

Treating Aperture Problem in Tracking

- Get additional info to constrain motion:
  - OF: Smoothly regularize in space
  - Tracking: Assume single motion for a region
- Spatial coherence constraint: "A pixel's neighbours all move together"



Computer Vision

Least Squares Problem: Single motion with multiple equations

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

Over determined System of Equations

$$\underset{25\text{x}2}{A} \; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Pseudo Inverse

$$\underset{2\text{x}2}{(A^T A)} \; \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

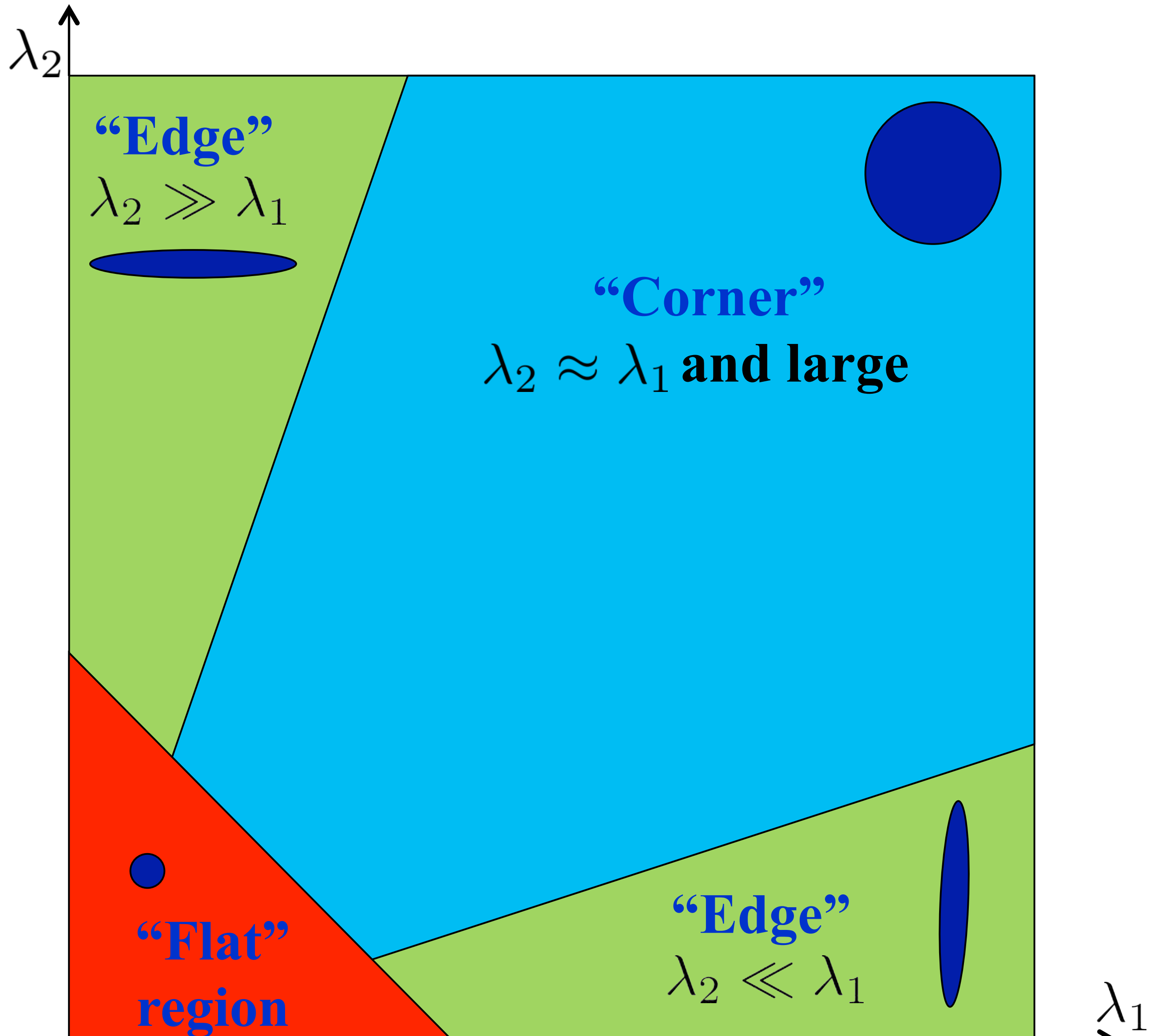
Eigenvectors of A^TA

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

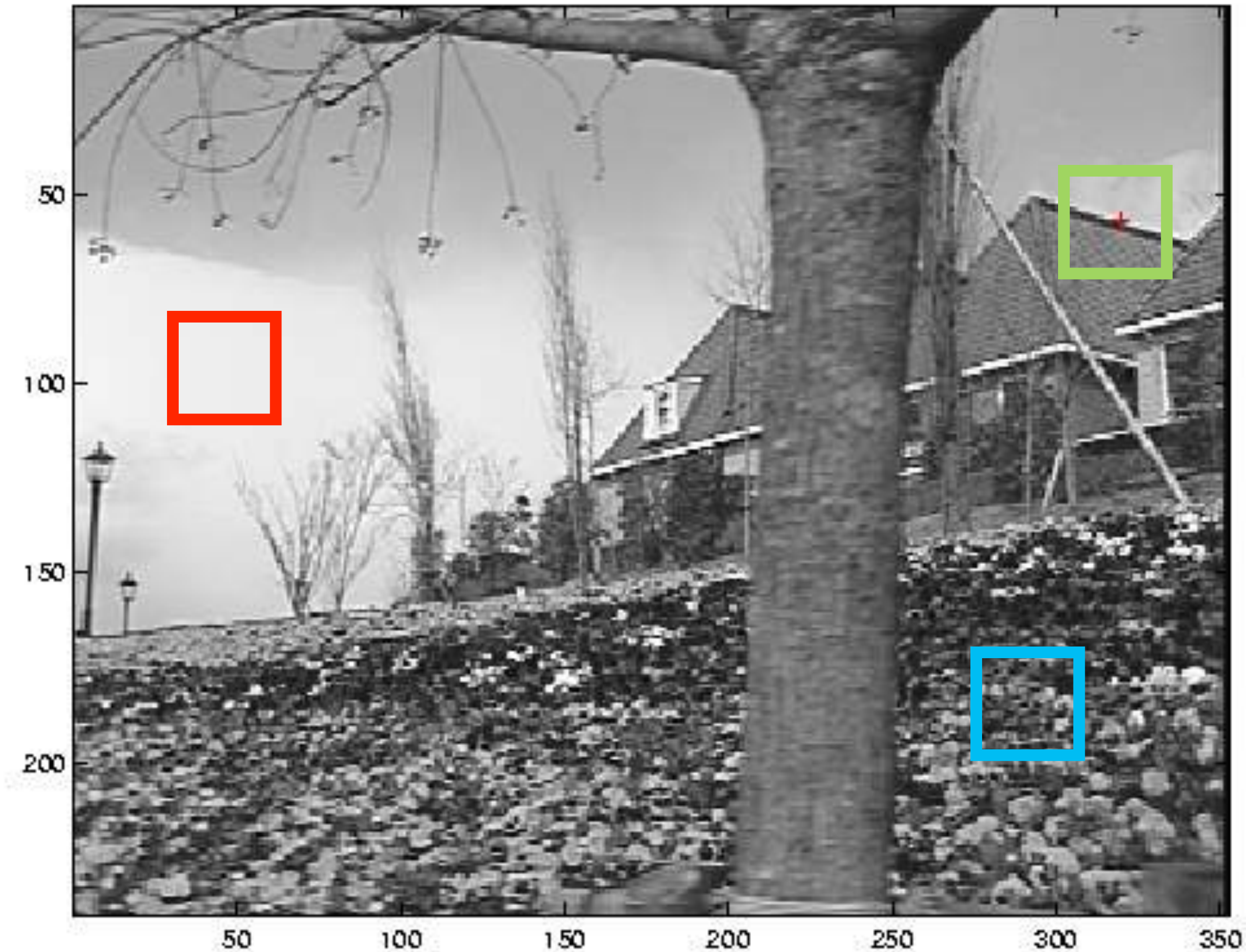
- Haven't we seen an equation like this before?

- Recall Harris corner detector!
- Thus, "good image features are also good for tracking"

Interpreting the Eigenvalues



Samples: Edge / Low Texture / High Texture



Computer Vision

Example

Outline

Feature

- Region Tracking
- Point Tracking
- Template Tracking

Model

- Tracking-by-Detection
  - of a specific target
  - of the object class
- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)

Template Tracking

Lucas-Kanade Template Tracker

- Lucas-Kanade is typically for small patches, e.g. 5x5
- Why not run it for the entire object (for a larger window)

- Locally, translation is sufficient to explain motion; but…

Lucas-Kanade Template Tracker

- Motion is more complex in a larger window

- Nonetheless, we can easily generalize the motion model to other parametric models!

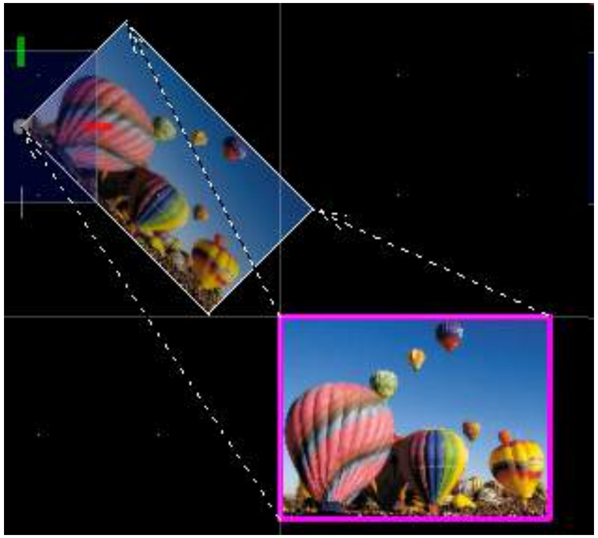
e.g., translation, affine, projective, “warp”

$$E(u,v) = \sum_{x,y} [I(x+u, y+v) - T(x,y)]^2$$

$$E(u,v) = \sum_{x,y} [I(W(x;p)) - T(x,y)]^2$$

Lucas-Kanade Template Tracker

- From Points to templates
- Estimate „optimal“ warp W

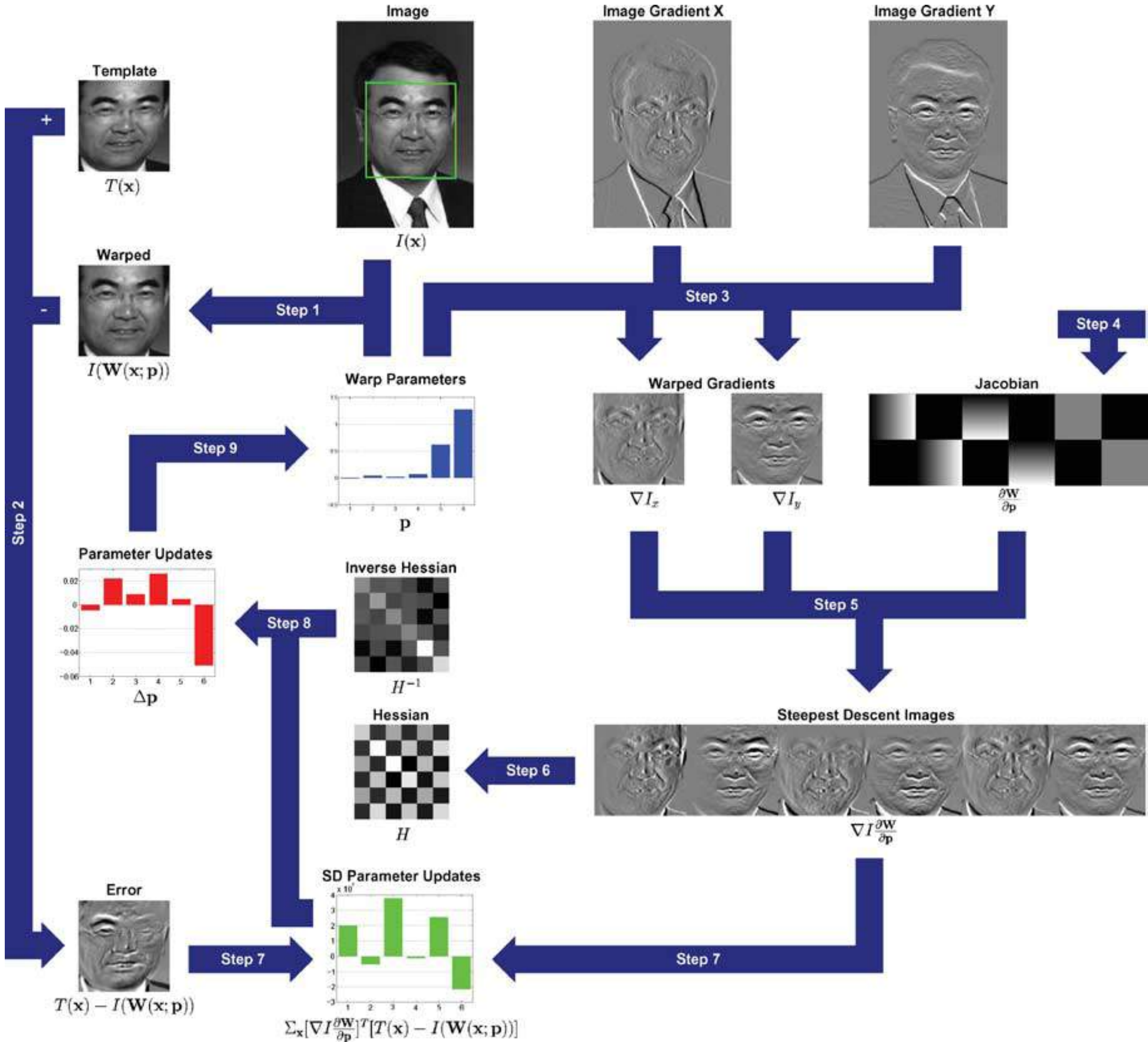


$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]^2$$

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{x})]^2$$

Computer Vision

[Baker & Matthews, IJCV'04, Lucas-Kanade 20 Years On: A Unifying Framework]



Lucas-Kanade Template Tracker

Step 1. Warp I to obtain $I(W([x\ y];P))$

Step 2. Compute the error image $T(x) - I(W([x\ y]; P))$

Step 3. Warp the gradient ∇I with $W([x\ y]; P)$

Step 4. Evaluate $\frac{\partial W}{\partial P}$ at $([x\ y]; P)$ (Jacobian)

Step 5. Compute steepest descent images $\nabla I \frac{\partial W}{\partial P}$

Step 6. Compute Hessian matrix $\sum (\nabla I \frac{\partial W}{\partial P})^T (\nabla I \frac{\partial W}{\partial P})$

Step 7. Compute $\sum (\nabla I \frac{\partial W}{\partial P})^T (T(x,y) - I(W([x,y];P)))$

Step 8. Compute ΔP

Step 9. Update $P \longleftarrow P + \Delta P$

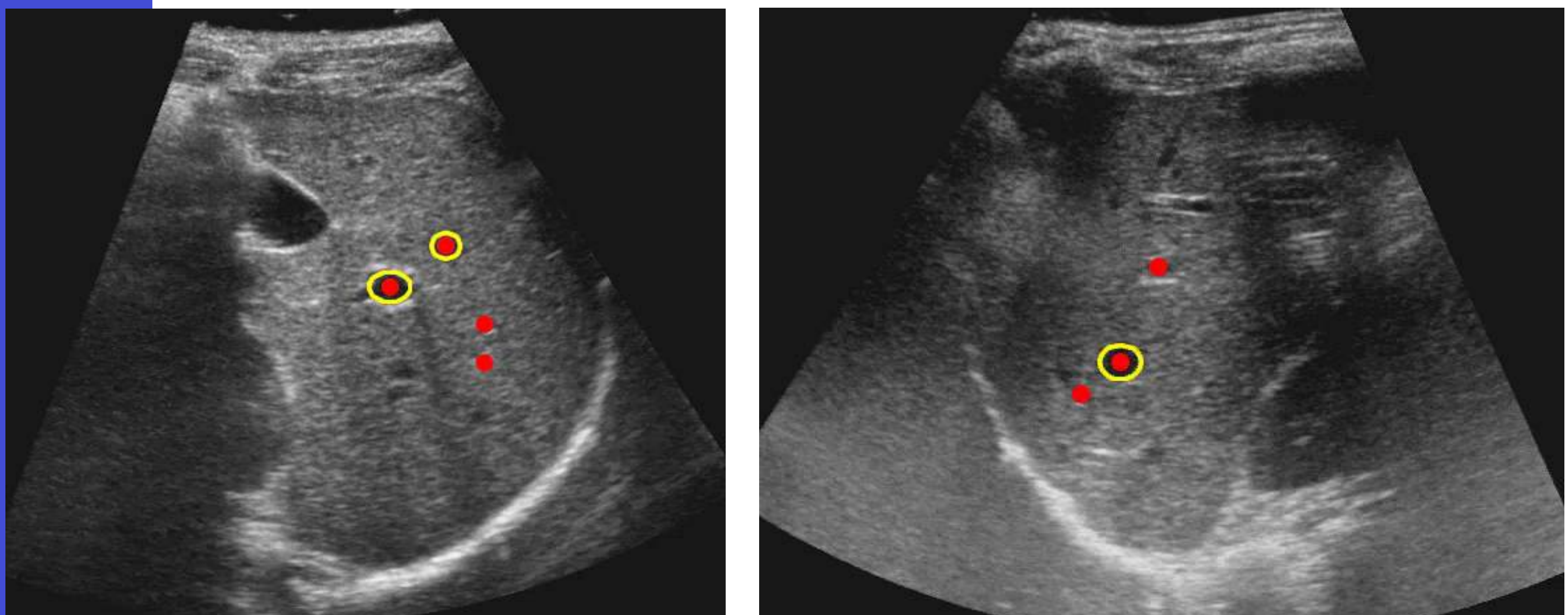
Computer Vision

Example

Computer Vision

Example: Tracking Liver in Ultrasound

[Makhinya and Goksel: **“Motion Tracking in 2D Ultrasound Using Vessel Models and Robust Optic-Flow”**, MICCAI CLUST, 2015]



● Our tracking
+ Manual annotation

Computer Vision

Outline

Feature

- Region Tracking
- Point Tracking
- Template Tracking

Model

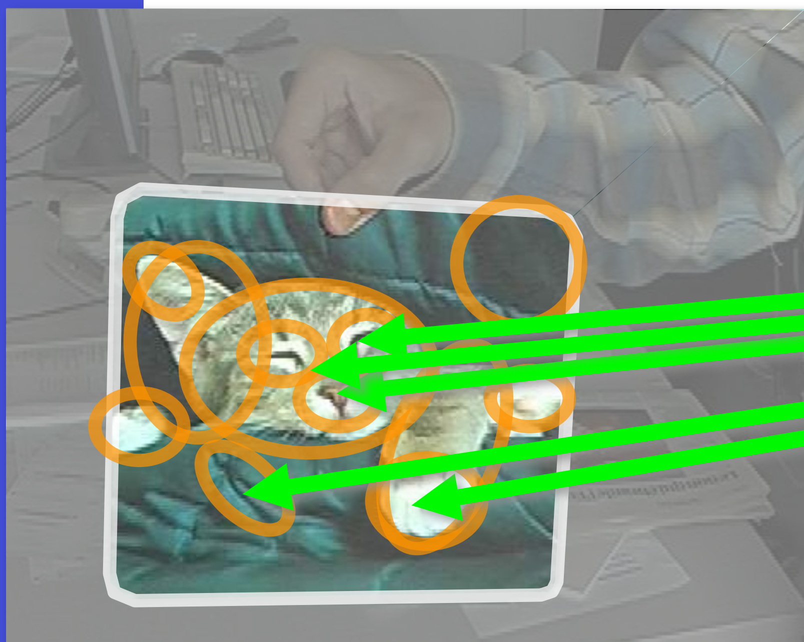
- Tracking-by-Detection
  - of a specific target
  - of the object class
- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)

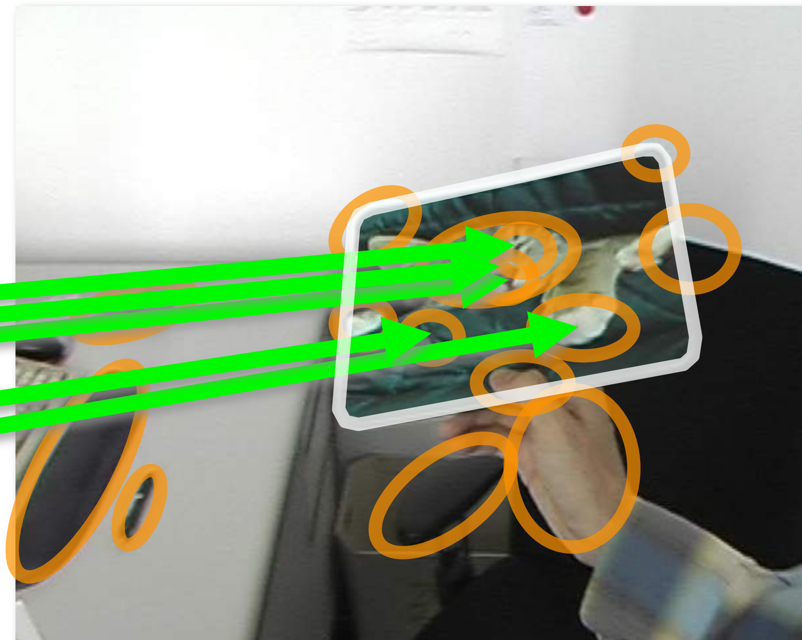
Tracking by Detection

(of a specific target)

3D Object Detection



Reference image(s) of the object to detect



Test image

Computer Vision

3D Object Detection

MathWorks

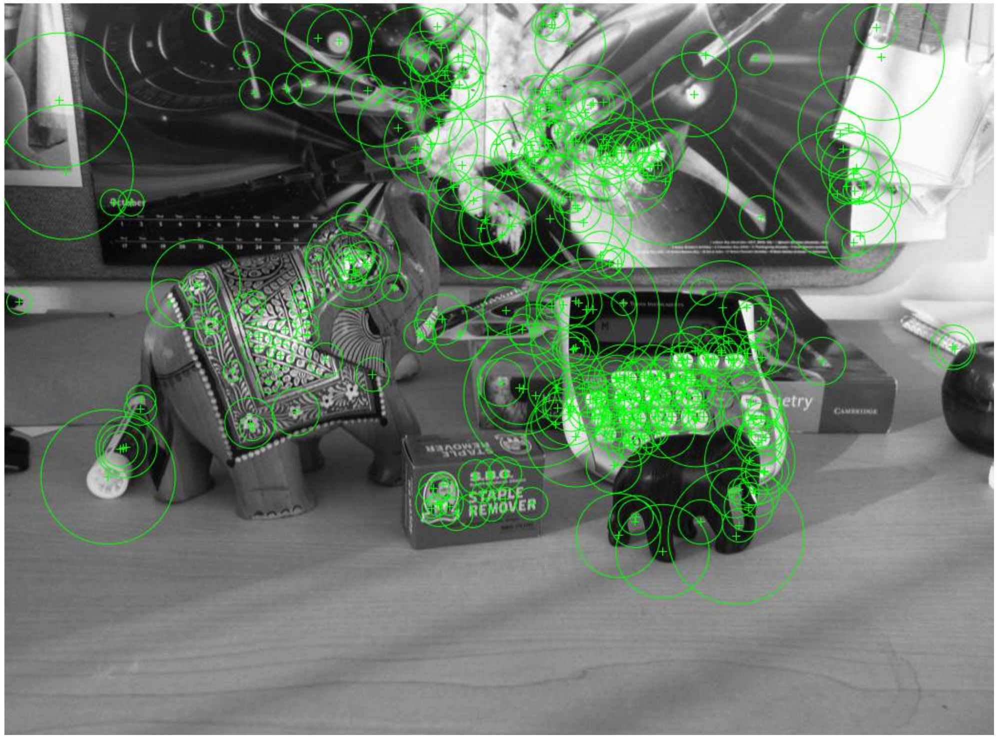
Reference image(s) of the object to detect

Test image

1. Detect Keypoints

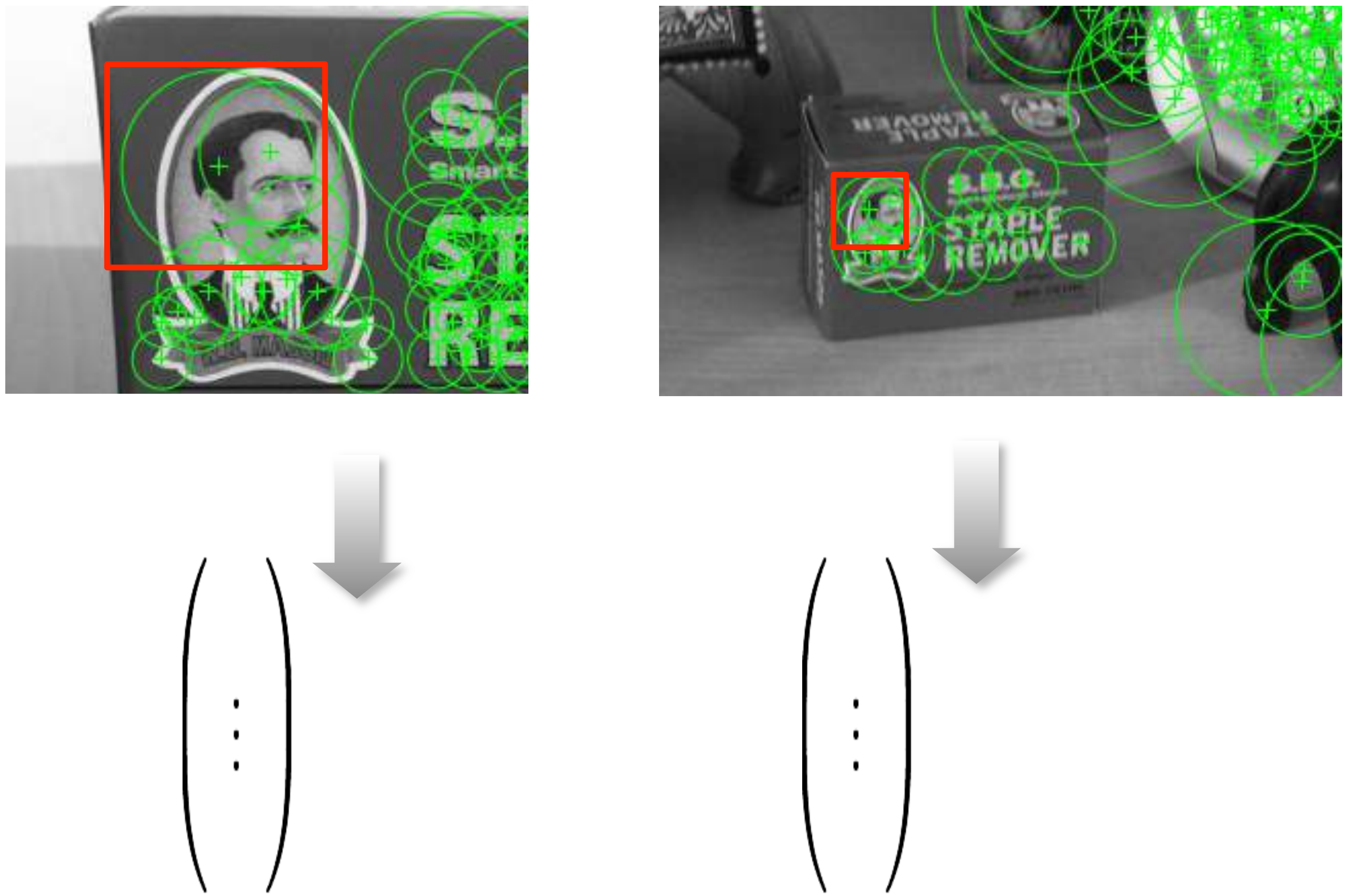
– invariant to scale, rotation, or perspective

100 strongest feature points in the reference image



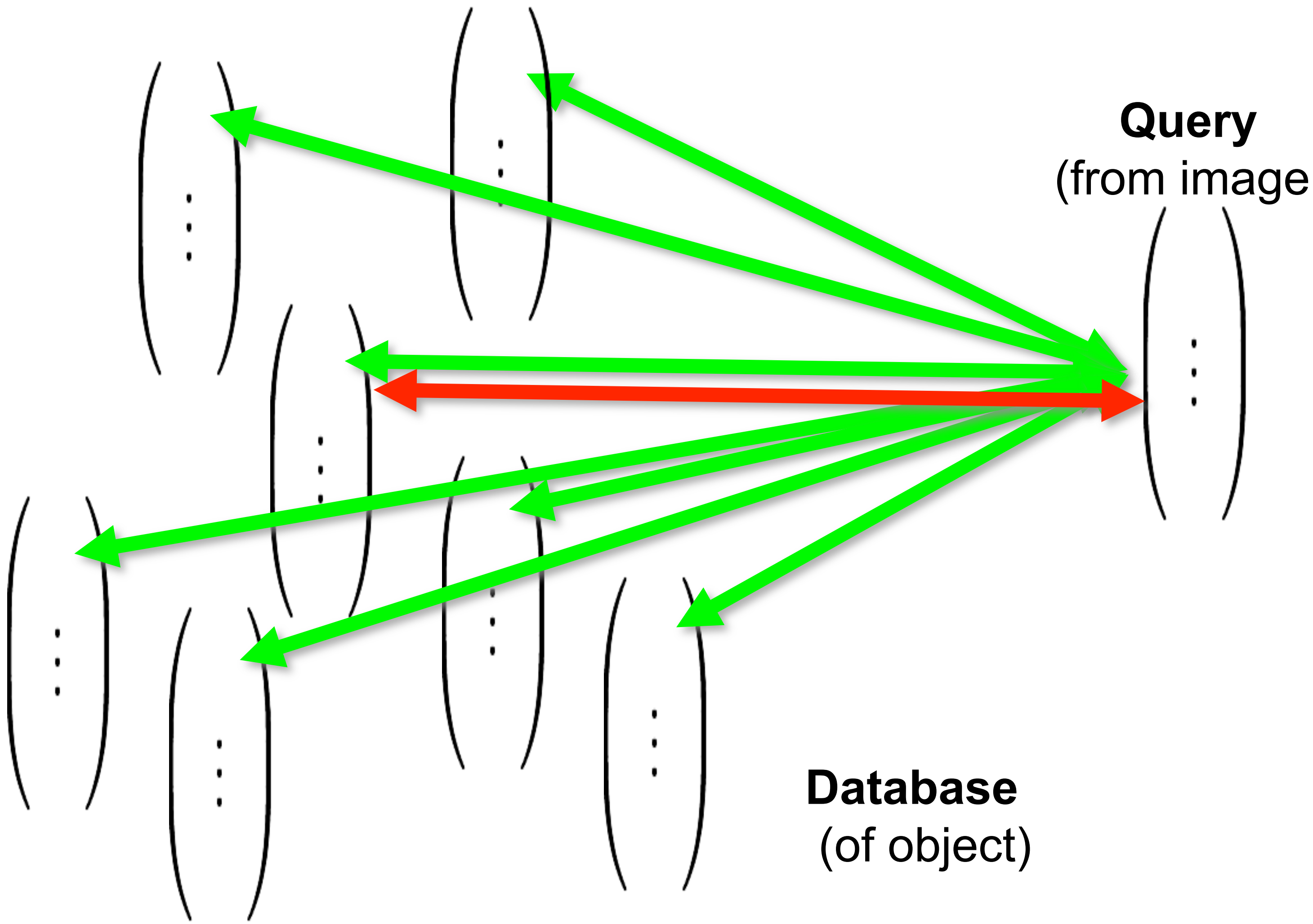
300 strongest feature points in the test image

2. Build Feature Descriptors

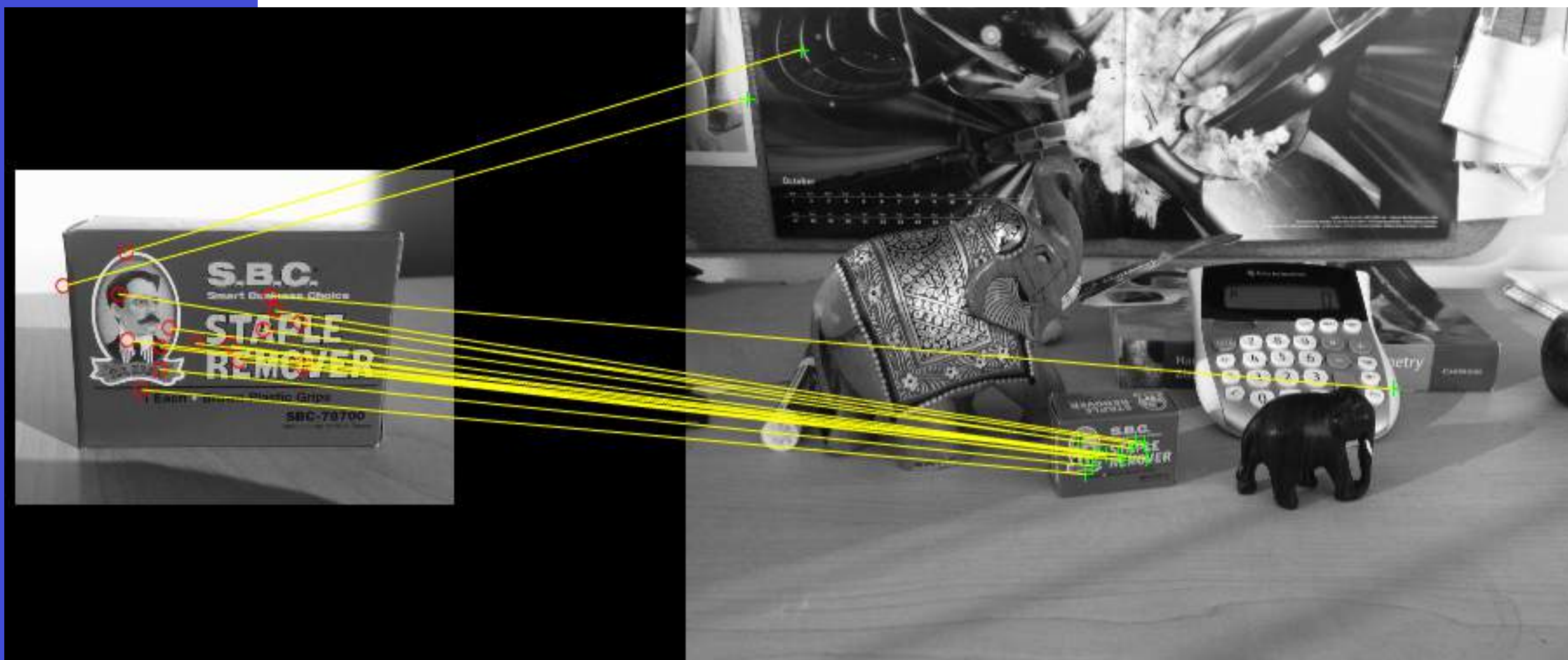


3. Match Keypoint Descriptors

- Search in the Database



3. Search in the Database



Computer Vision

4. Outlier Elimination

Summary

Keypoint Detection

$$\begin{pmatrix} \vdots \end{pmatrix}$$

Keypoint Recognition

Search in the Database

Robust 3D Pose Calculation

(RANSAC)

Geometric verification

Computer Vision

[Wagner et al. ISMAR'08]

Computer Vision

[Wagner et al. '09]

Outline

Feature

- Region Tracking
- Point Tracking
- Template Tracking

Model

- Tracking-by-Detection
  - of a specific target
  - of the object class
- Model-based Body Articulation
- On-line Learning

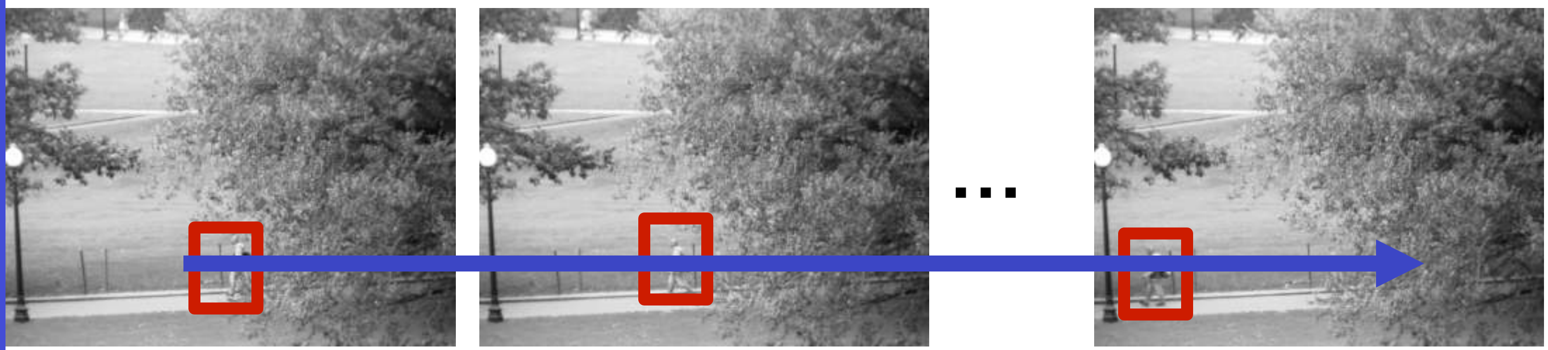
- Misc (preventing drift, context, issues)

Tracking by Detection

(of the object class)

“Multiple Object Tracking”

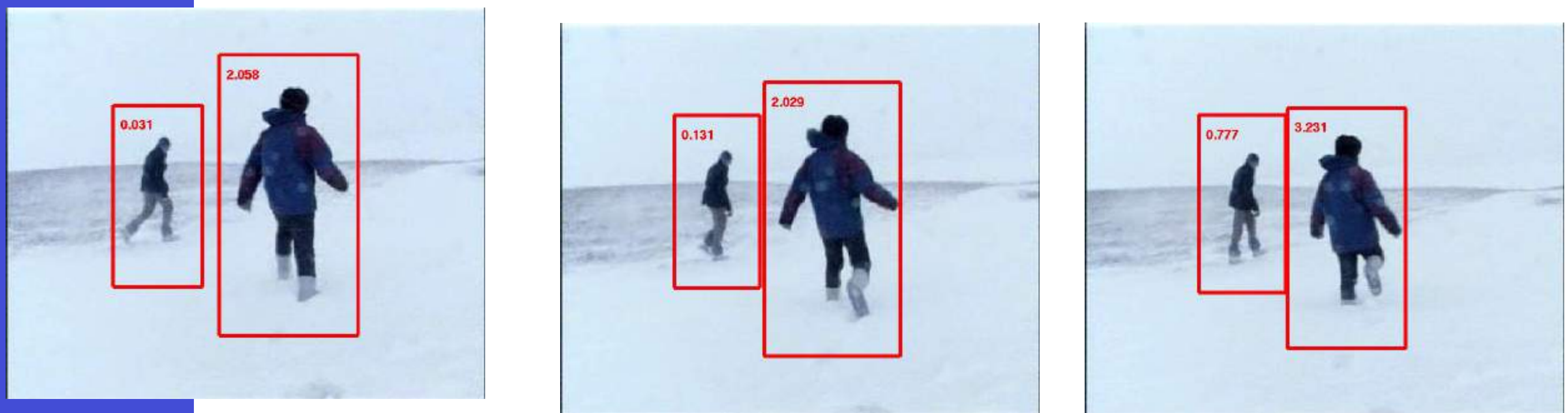
Tracking-by-Detection



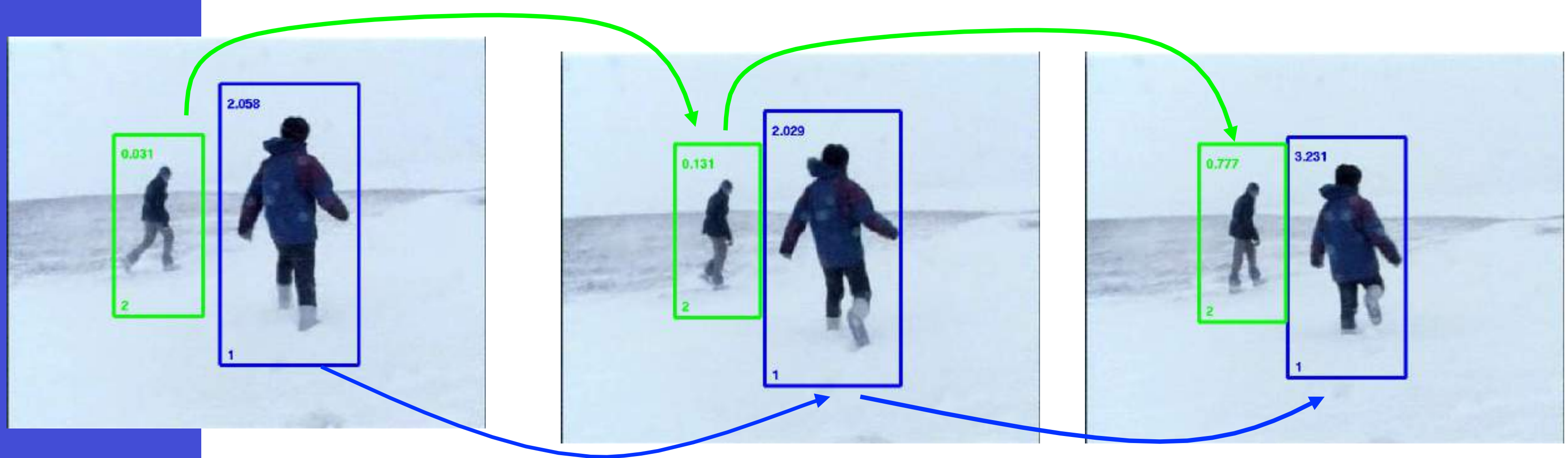
detect object(s) independently in each frame

associate detections over time into *tracks*

Multiple Objects



Frame 1 Frame 5 Frame 9



Computer Vision

Examples: Multiple Object Tracking

How to get the detections?

Persons

Background

Supervised Learning

(Support Vector Machines,
Random Forests,
Neural Networks, ...)

Computer Vision

Using the classifier

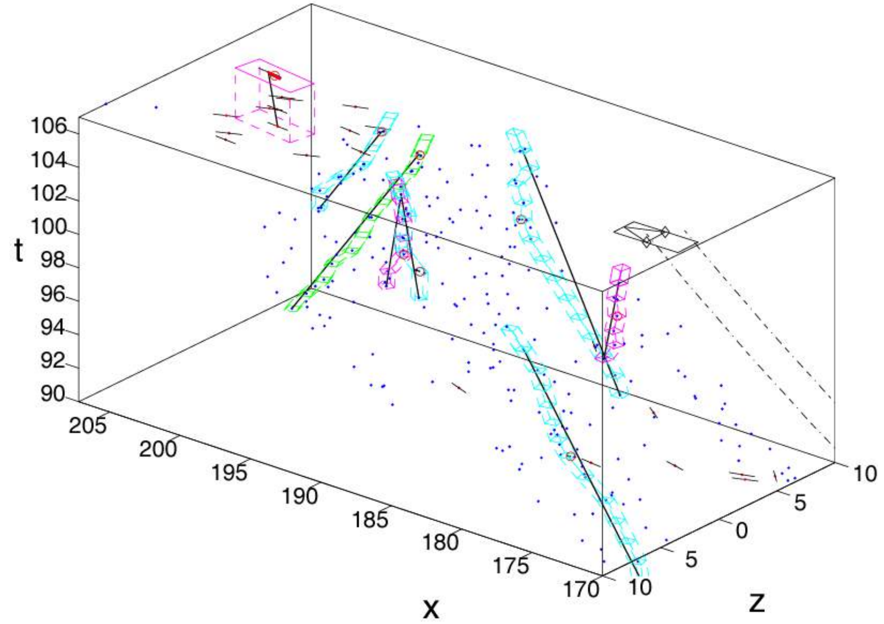
Computer Vision

Space-Time Analysis

- Collect detections in space-time volume

Detections

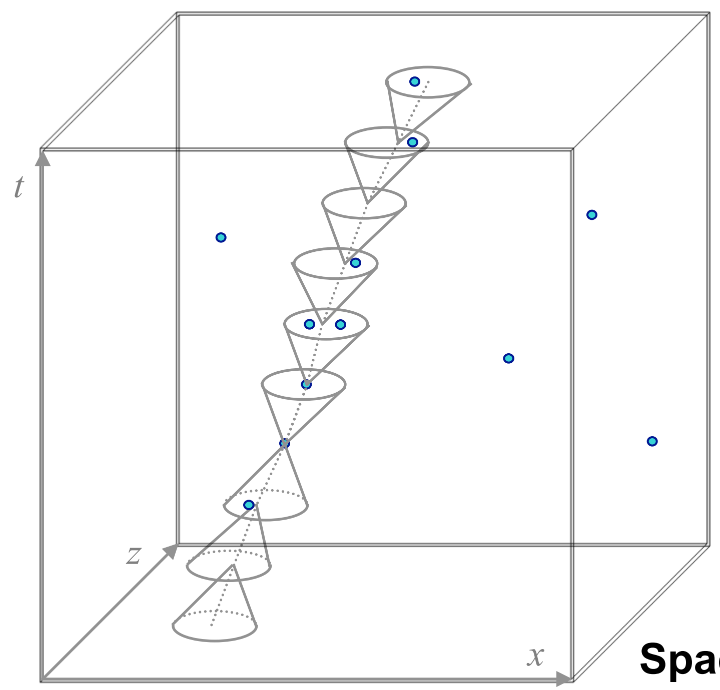
Space Time Volume



[Leibe et al. CVPR'07]

Trajectory Estimation

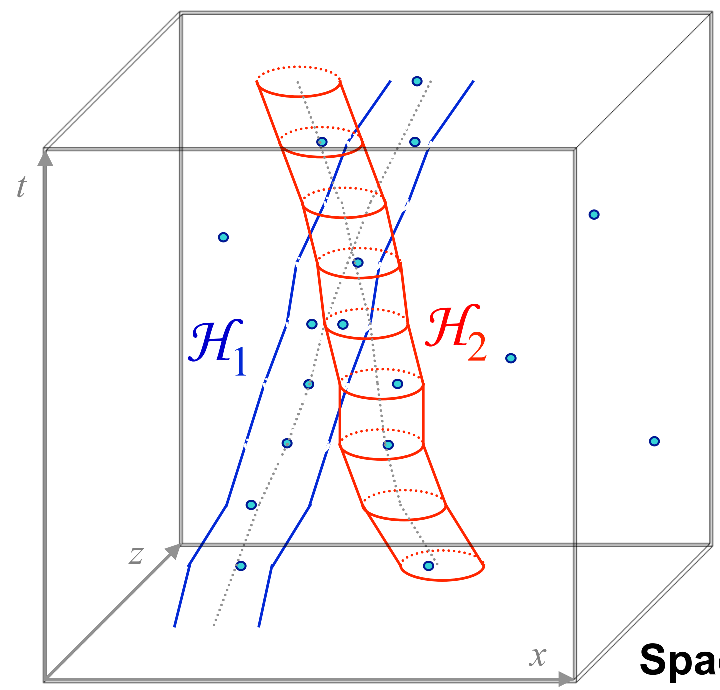
- Trajectory growing and selection



Space Time Volume

Trajectory Estimation

- Trajectory growing and selection



Space Time Volume

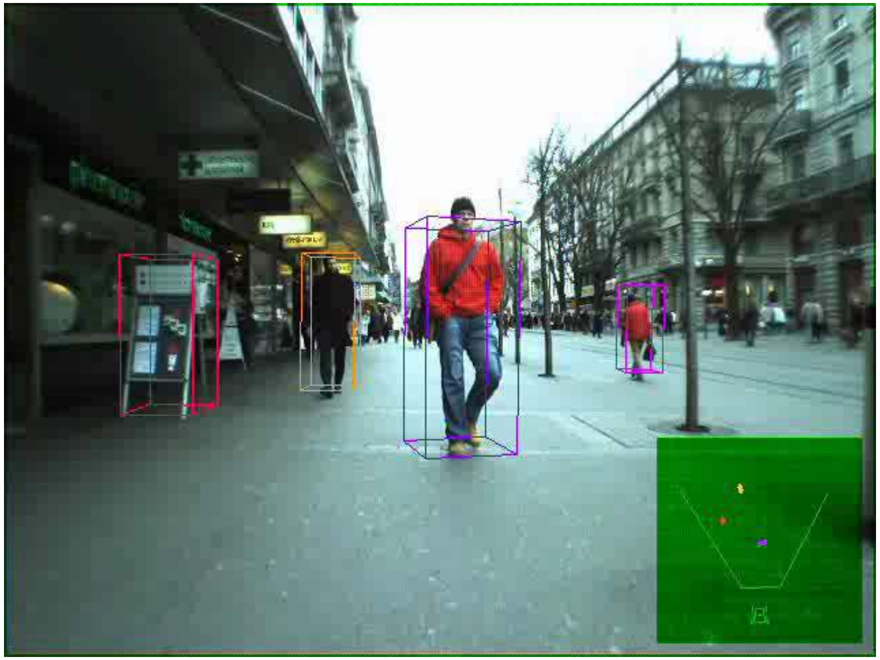
Driving

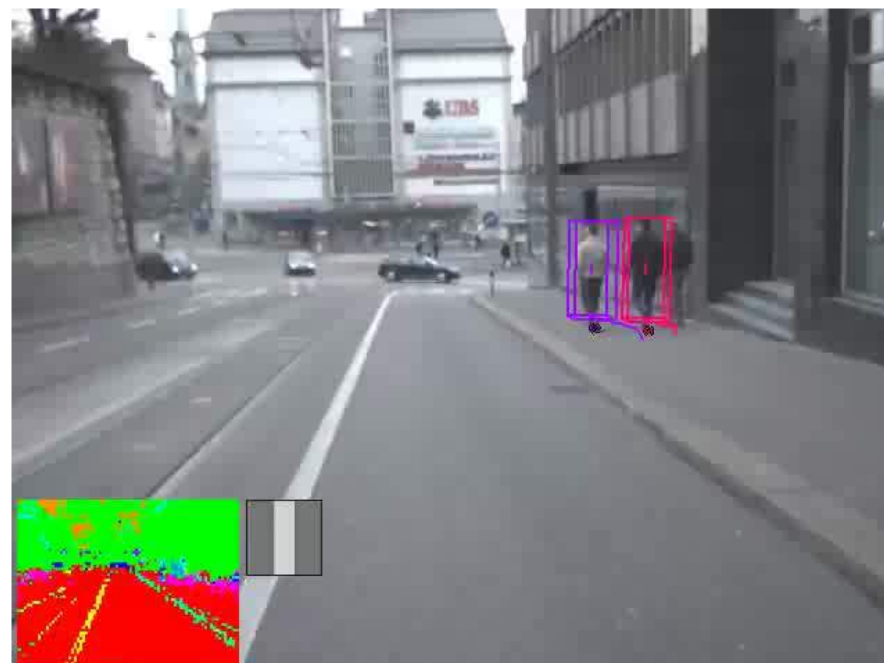
Input (Object Detections)

"Tracking" Result

Computer Vision

[Ess et al. CVPR'08]





Outline

Feature

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)

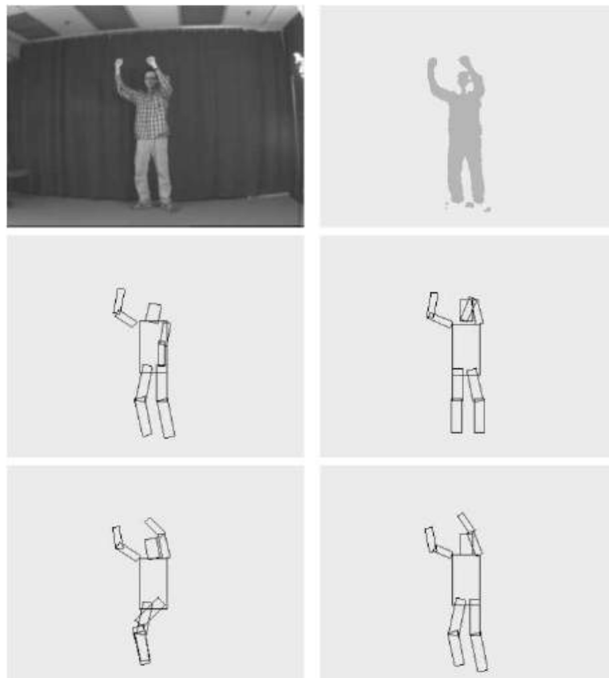
Model

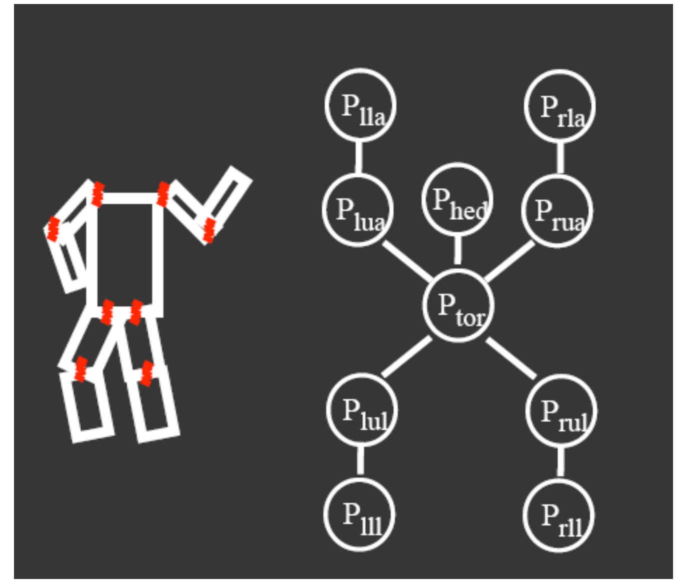
- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class (multiple object tracking)
- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)

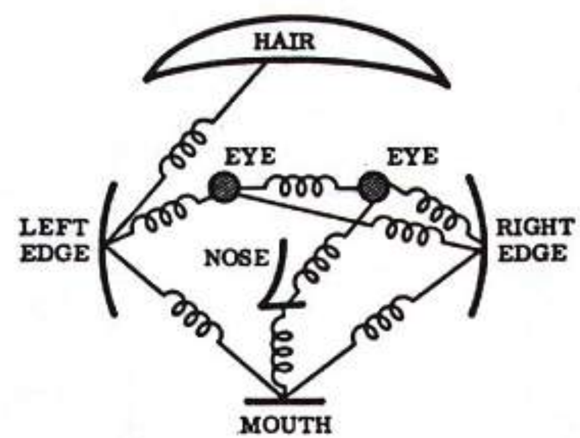
Model based Tracking

Articulated Tracking: Part-Based Models



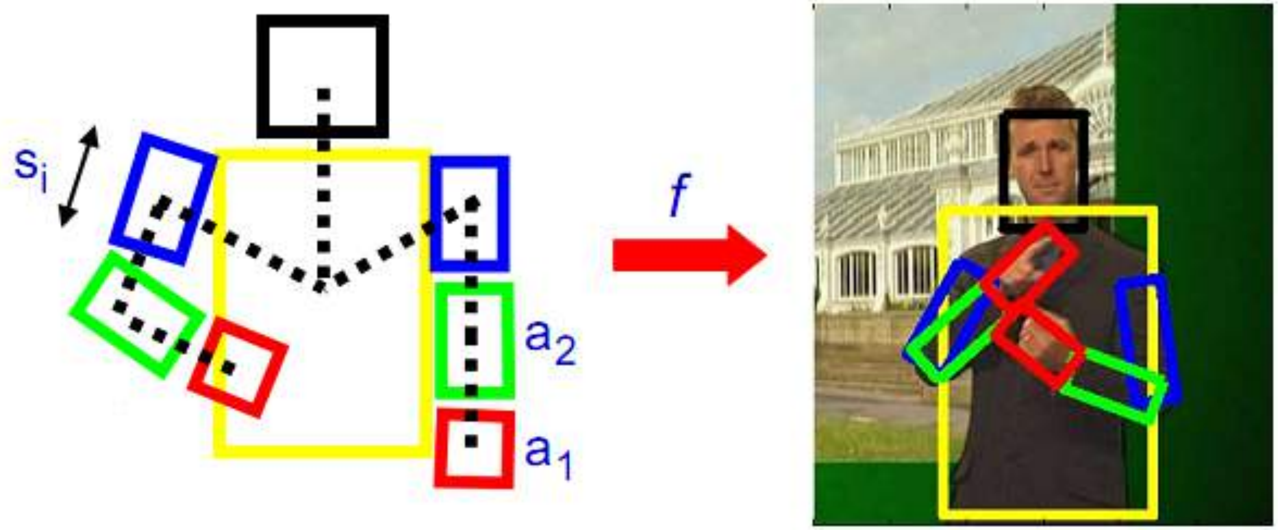


- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973



Parts-based analysis

Objective: detect human and determine upper body pose (layout)

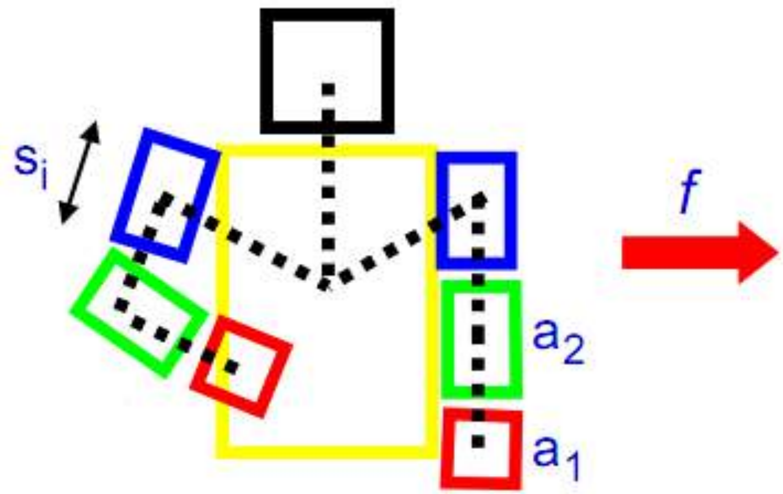


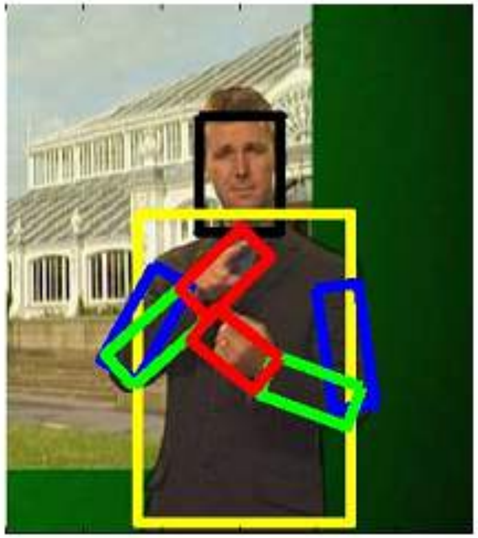
Model as a graph labelling problem

- Vertices $\mathcal{V}$ are parts, $a_i, i = 1, \cdots, n$
- Edges $\mathcal{E}$ are pairwise linkages between parts
- For each part there are h possible poses $\mathbf{p}_j = (x_j, y_j, \phi_j, s_j)$
- Label each part by its pose: $f : \mathcal{V} \longrightarrow \{1, \cdots, h\}$, i.e. part a takes pose $\mathbf{p}_{f(a)}$.

Parts-based analysis

Pictorial structure model – CRF





- Each labelling has an energy (cost):

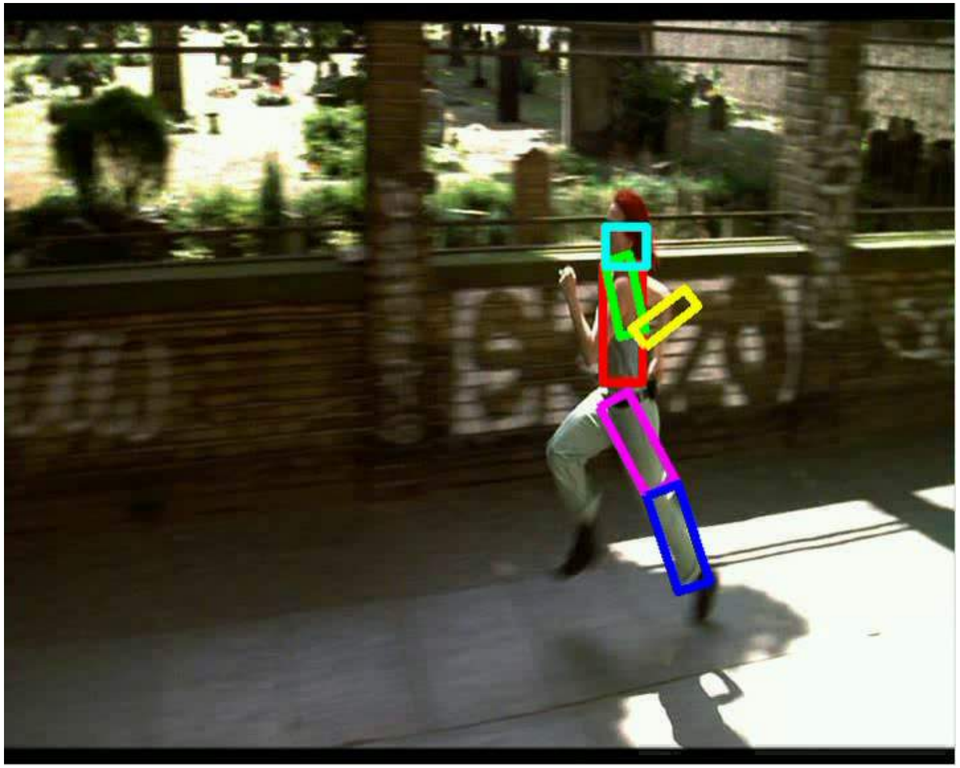
$$E(f) = \underbrace{\sum_{a \in \mathcal{V}} \theta_{a;f(a)}}_{\substack{\text{unary terms} \\ \text{(appearance)}}} + \underbrace{\sum_{(a,b) \in \mathcal{E}} \theta_{ab;f(a)f(b)}}_{\substack{\text{pairwise terms} \\ \text{(configuration)}}}$$

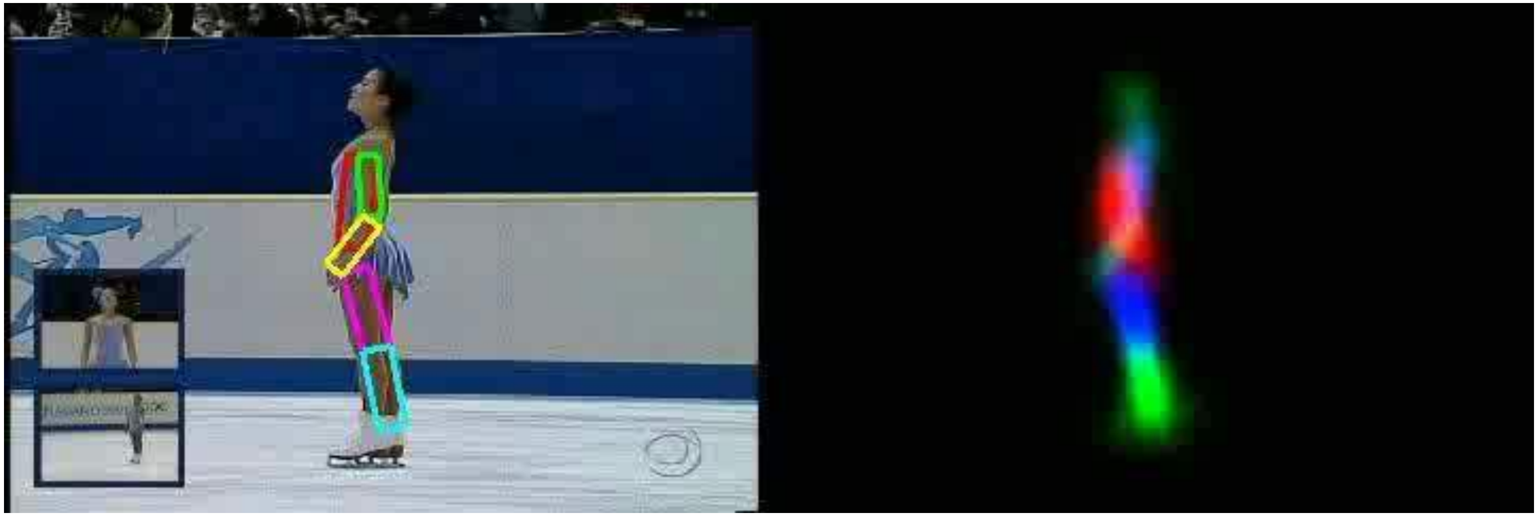
Features for unary:
- colour
- HOG

for limbs/torso

- Fit model (inference) as labelling with lowest energy

Computer Vision





[Ramanan et al. CVPR'05]

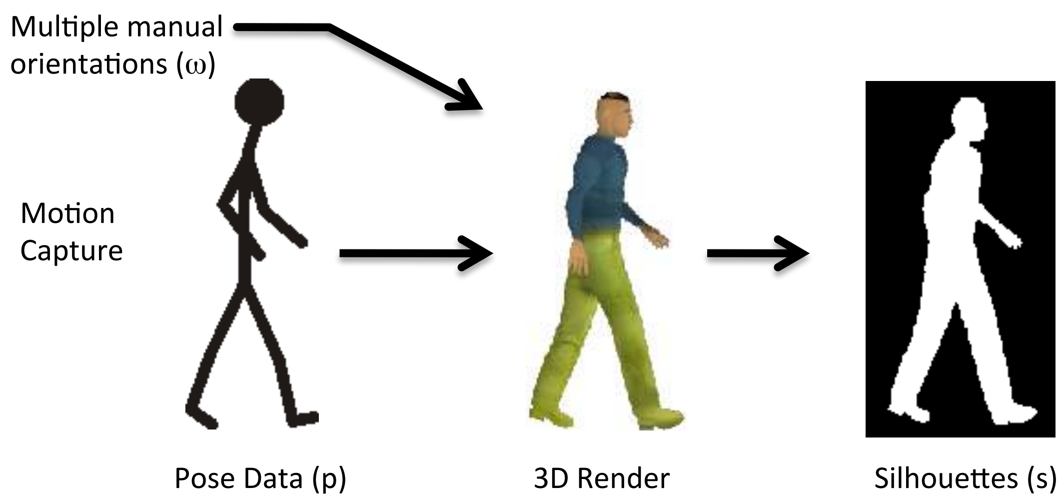
Walking

- What temporal info can we use for tracking?
- What variation would we expect in population?

Articulation Space

Tracking Articulated Motion as High-Dimensional Inference

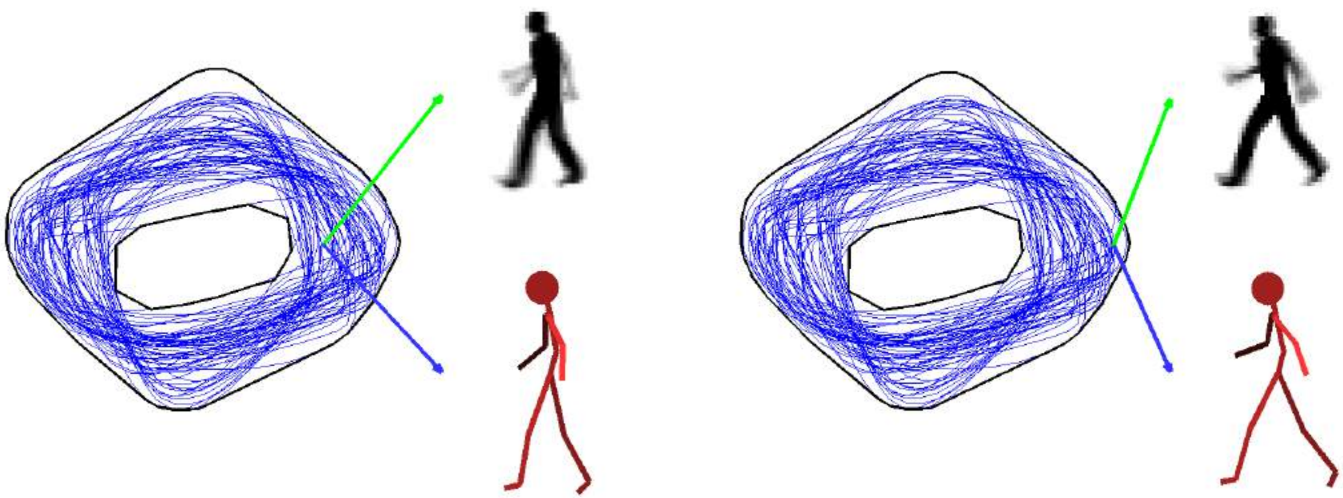
- Walking cycles have one main (periodic) DOF
- Regressors to learn this (latent) space, and its variation (Gaussian Process regression, **PCA**, etc)
- (Pose,Silhouette) training data can be obtained by 3D rendering



Articulation Space

Tracking Articulated Motion as High-Dimensional Inference

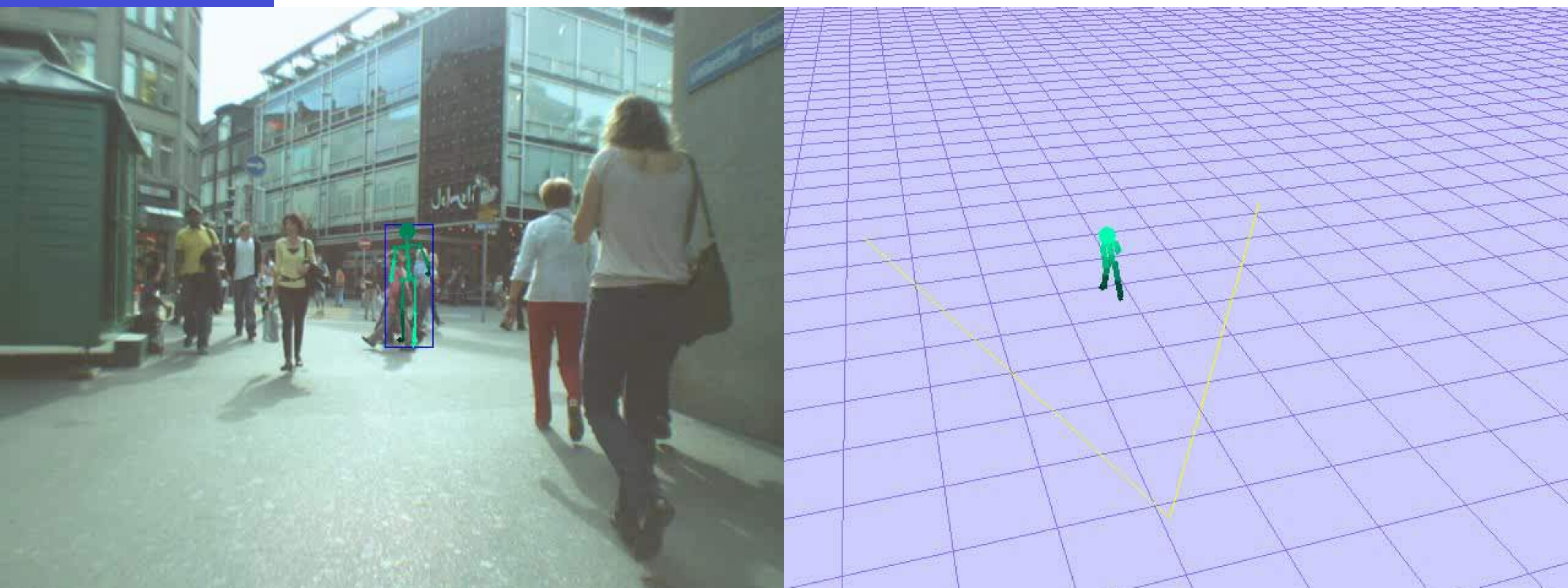
- Walking cycles have one main (periodic) DOF
- Regressors to learn this (latent) space, and its variation
  (Gaussian Process regression, PCA, etc)
- (Pose,Silhouette) training data can be obtained by 3D rendering



P(Silhouette|k)
perform inference
on silhouettes

P(Pose|k)
recover pose
from latent space

Articulation Space Tracking



Outline

Feature

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)

Model

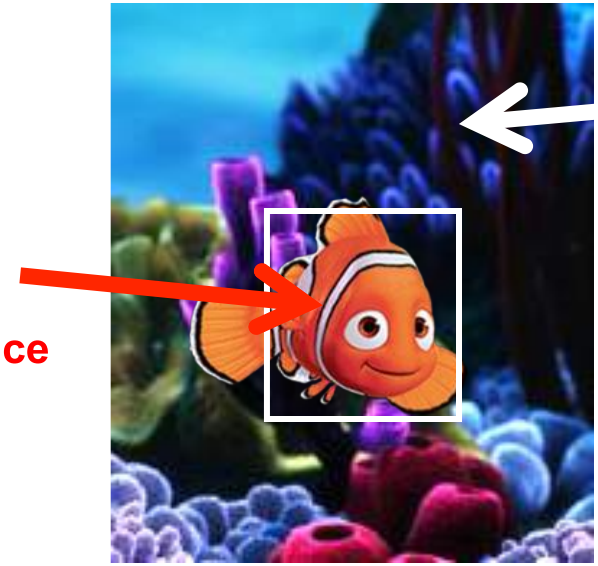
- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class (multiple object tracking)
- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)

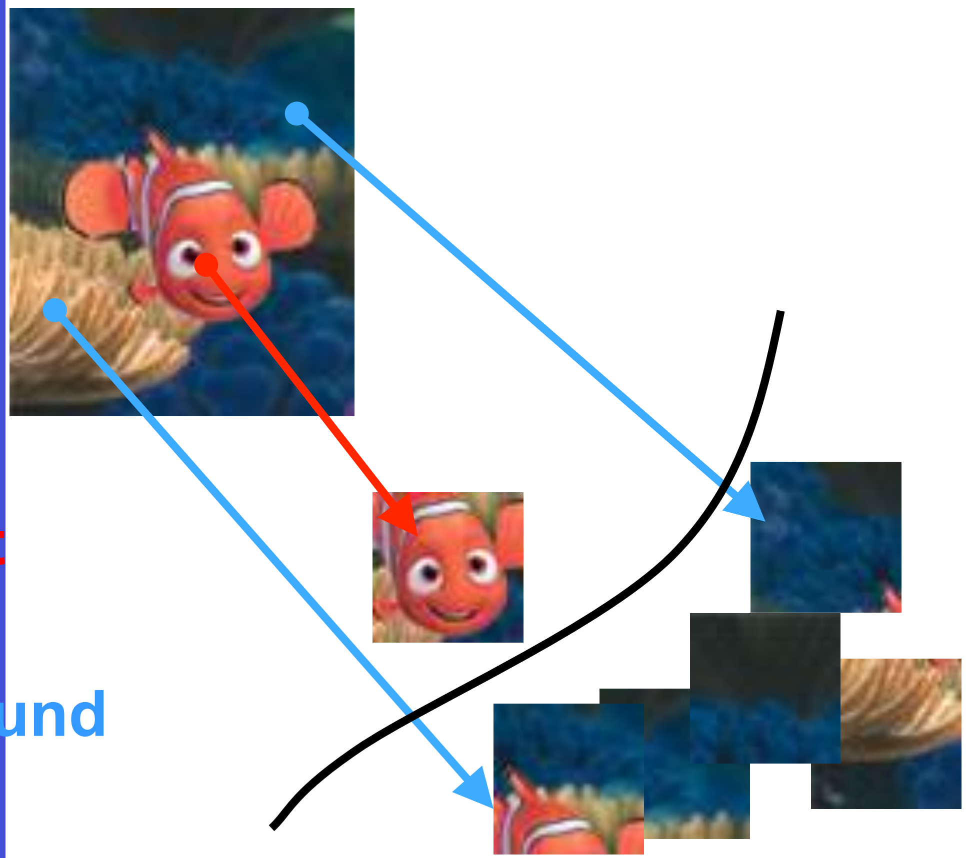
Tracking as On-line learning (updating tracking models)

Tracking as Classification

- Learning current object appearance vs. local background.



Tracking as Classification

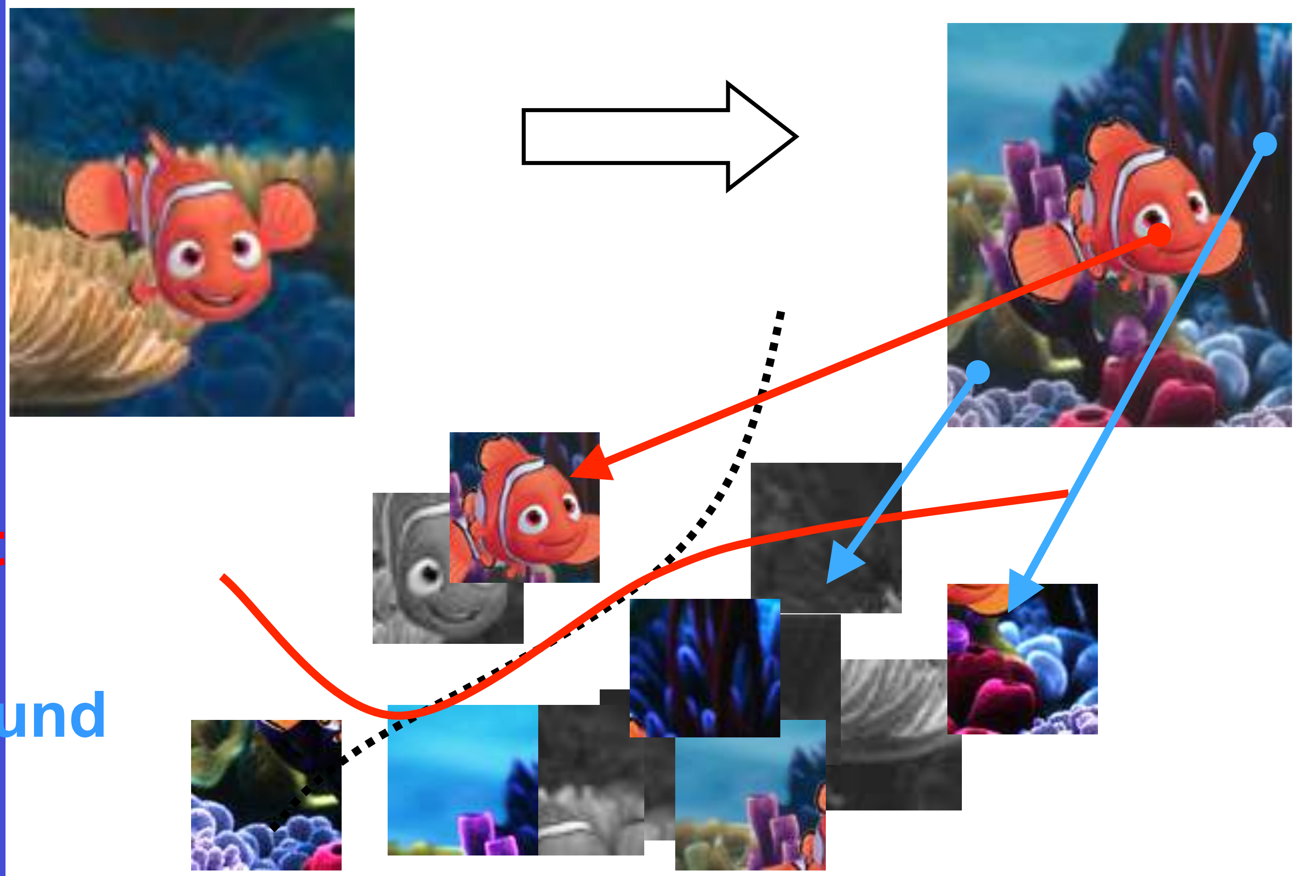


object

vs.

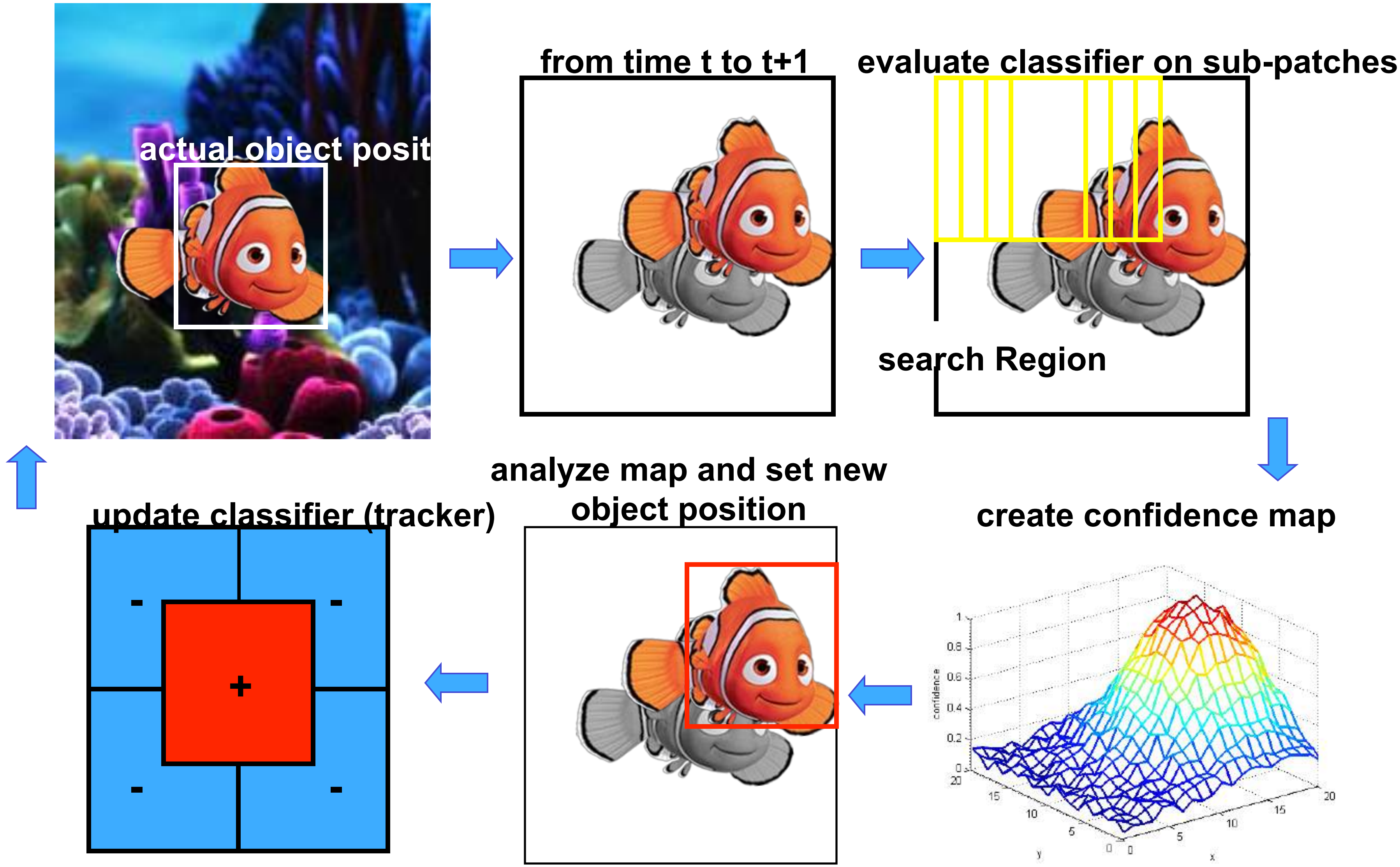
background

Tracking as Classification

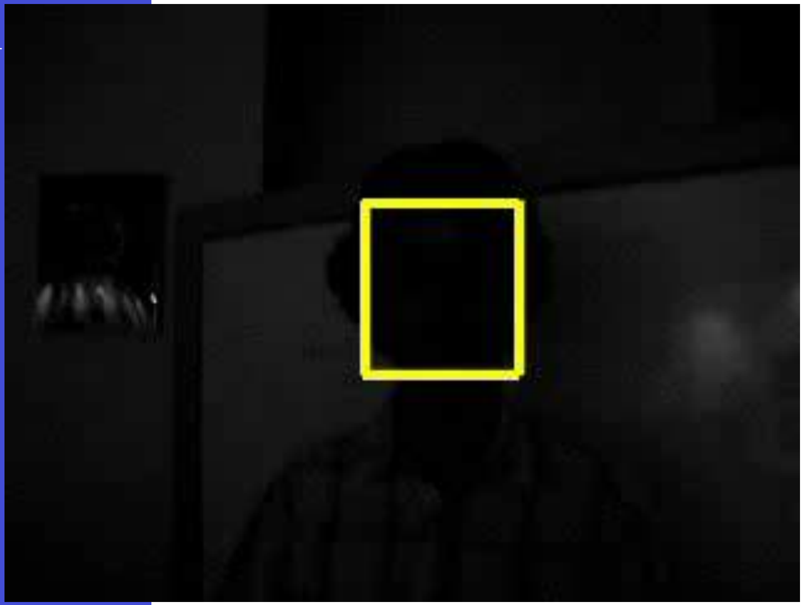


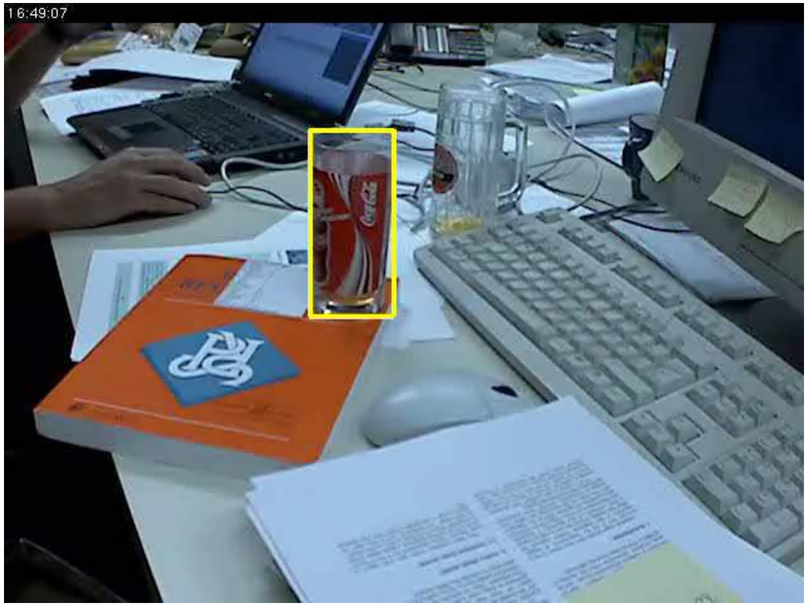
object
vs.
background

Tracking Loop



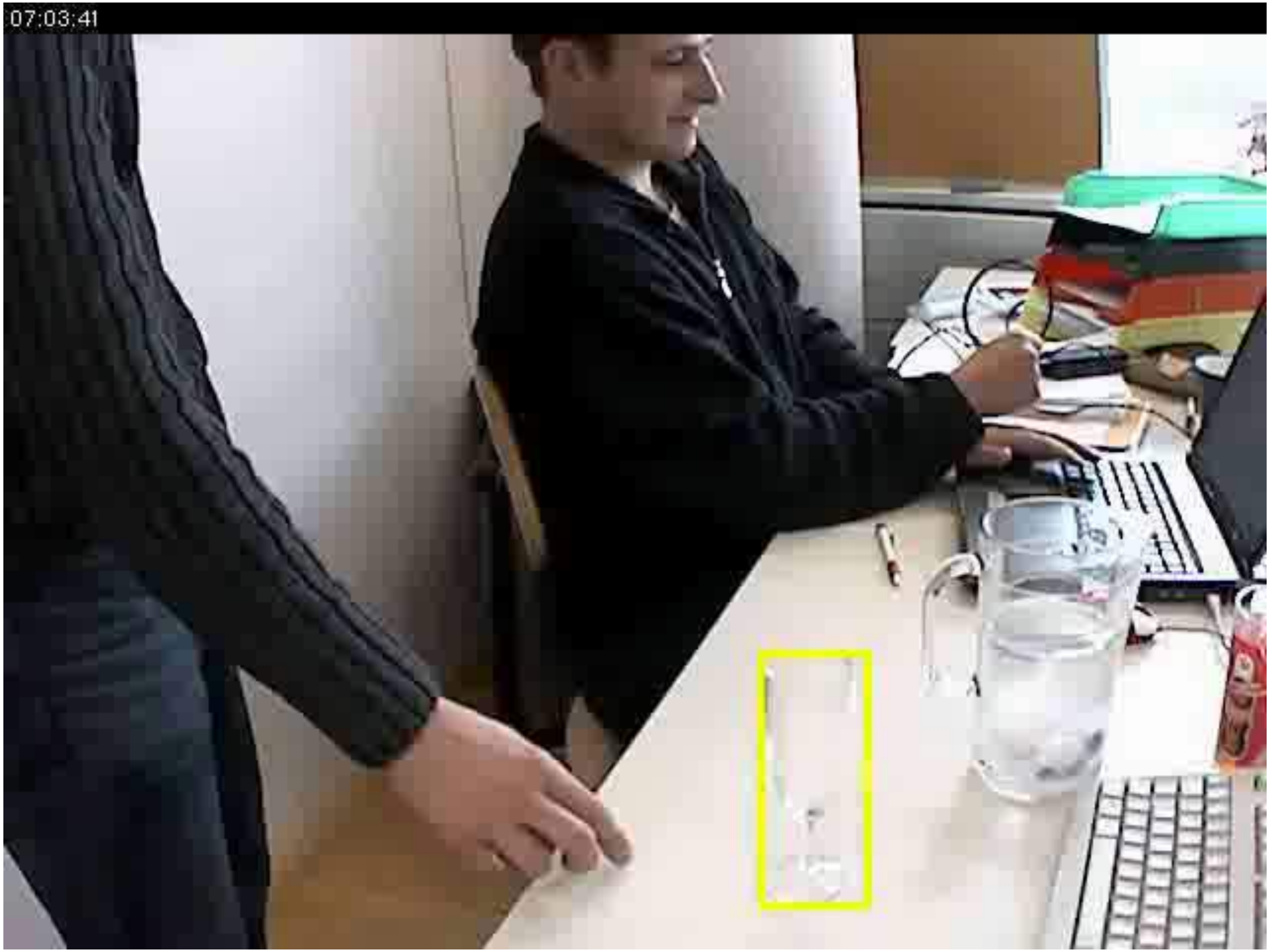
Computer
Visi





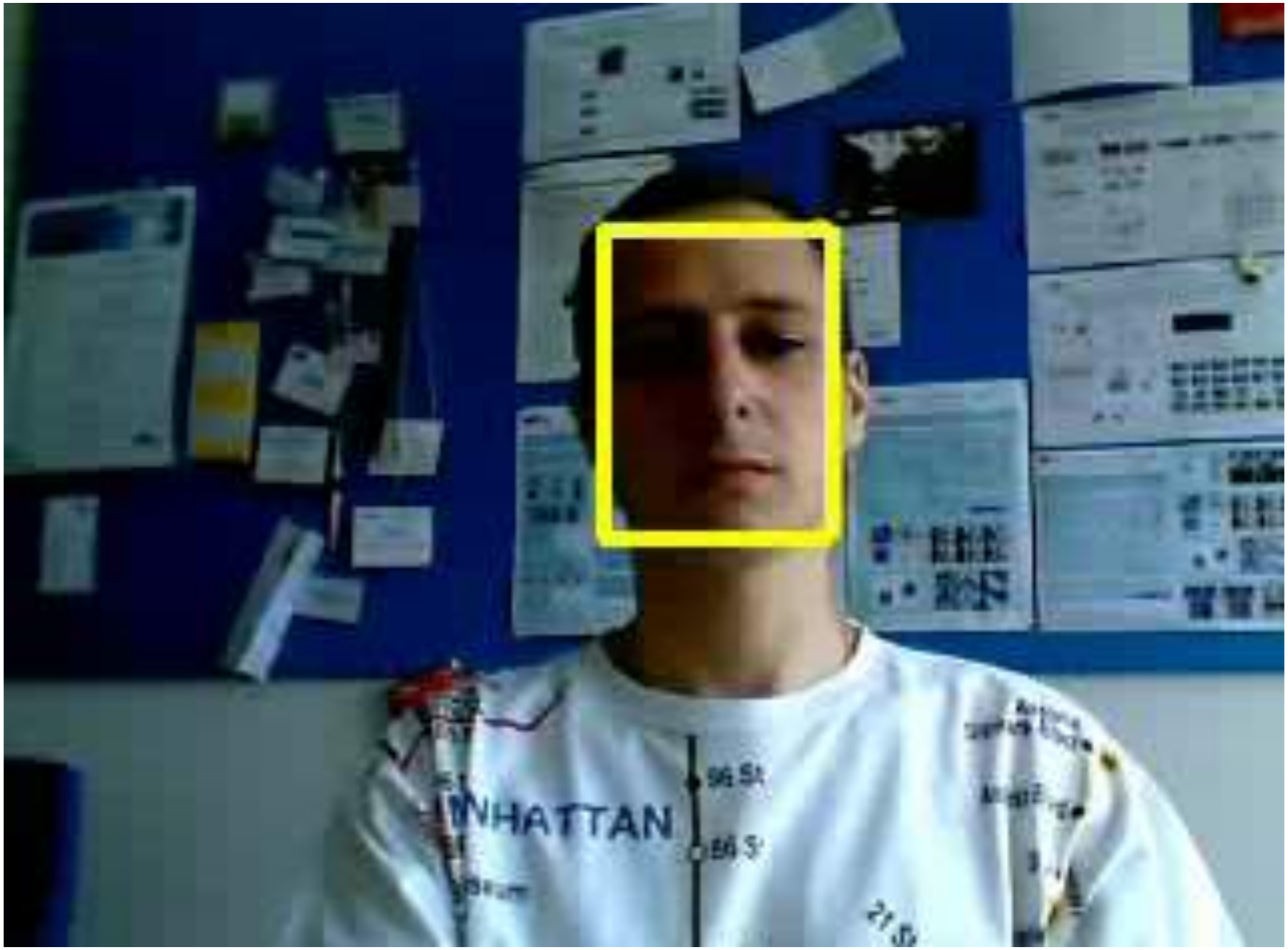
Computer Vision

“Tracking the Invisible”

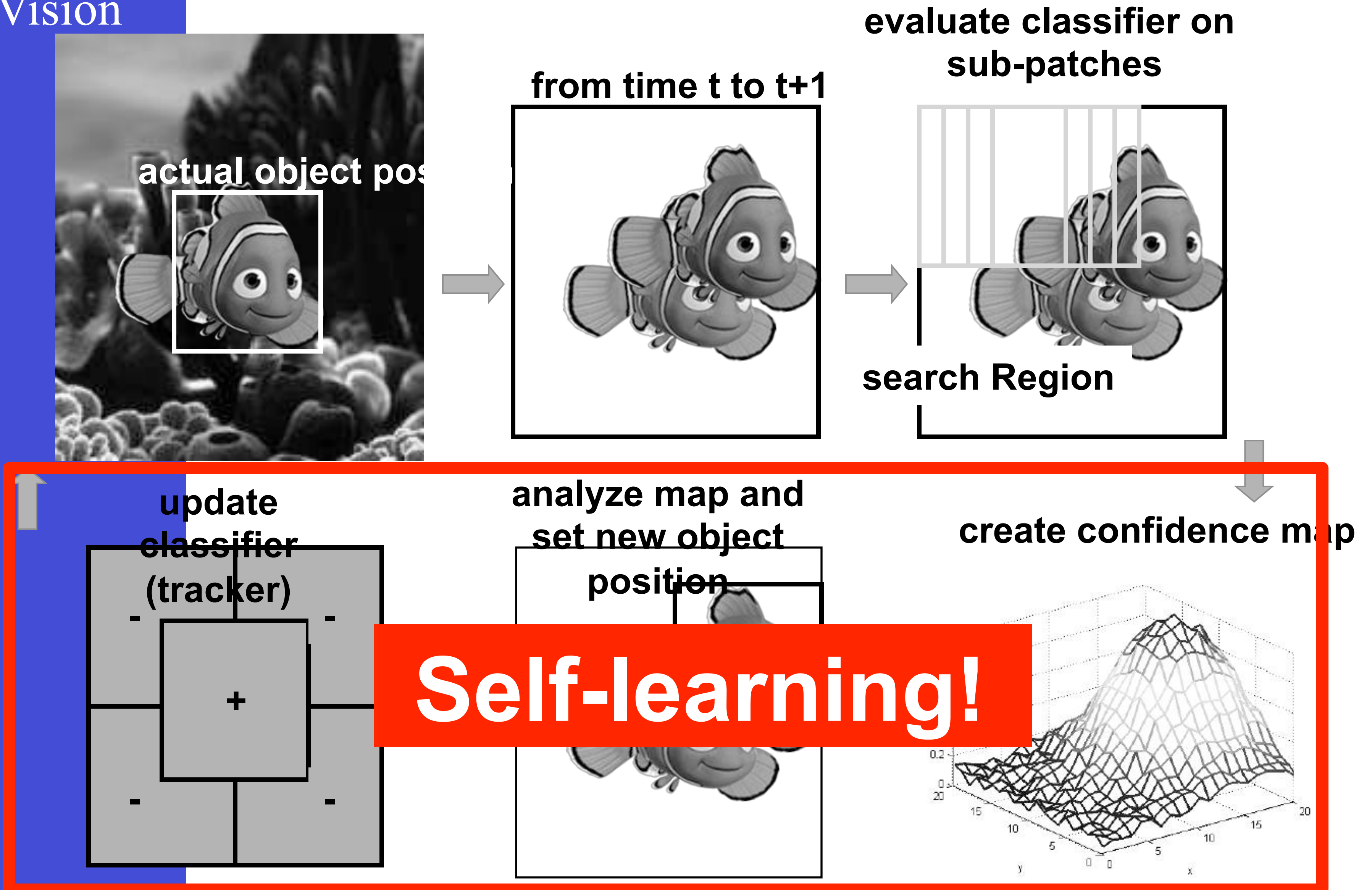


[Grabner et al. CVPR’06]

When does it fail…

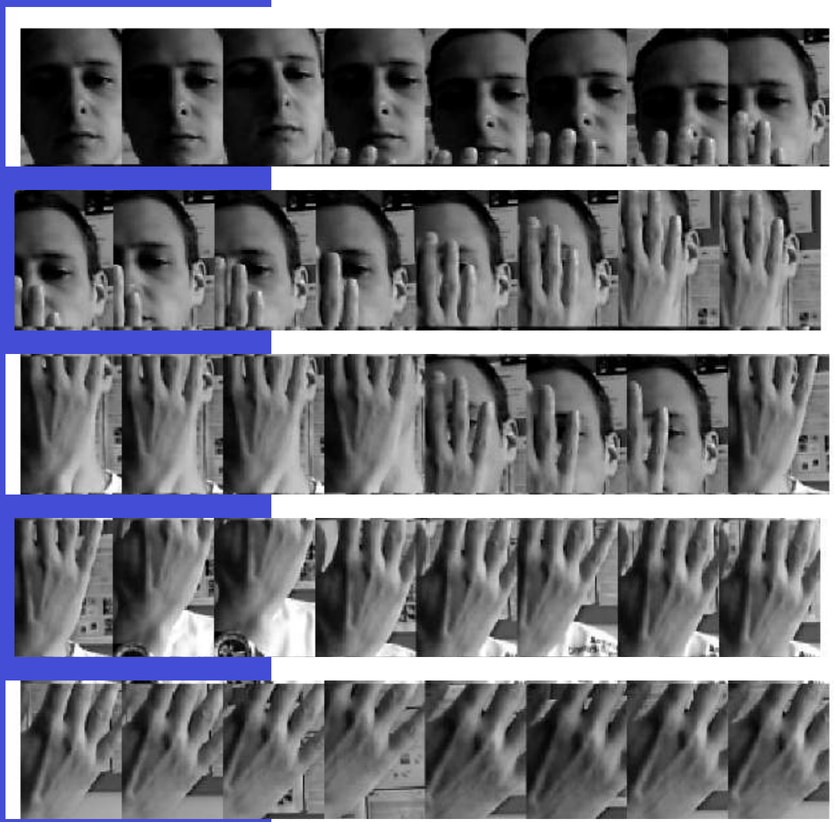


When does it fail…

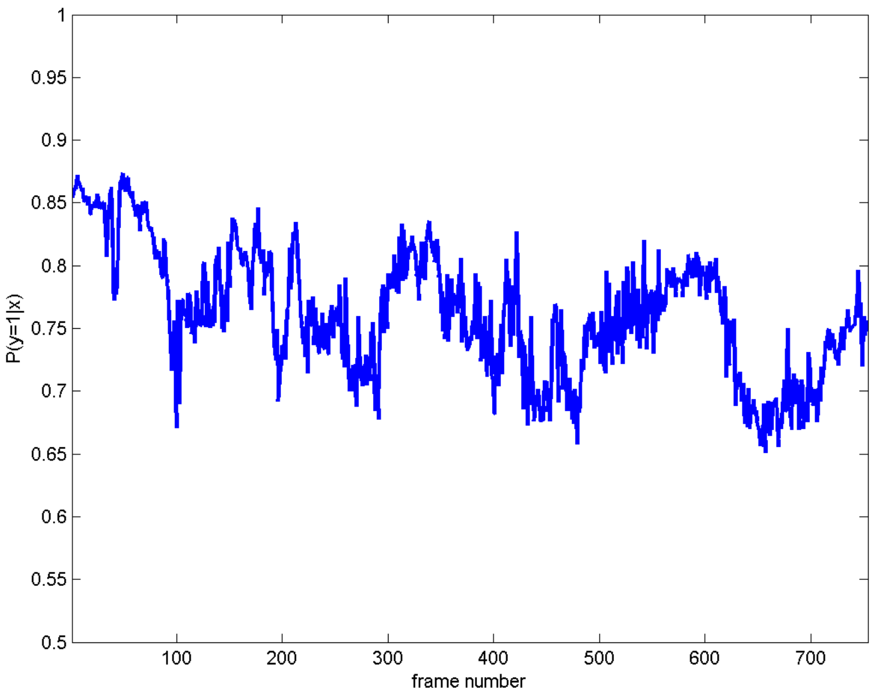

actual object pos
from time t to t+1
evaluate classifier on sub-patches
search Region
create confidence map
analyze map and set new object position
update classifier (tracker)
-
-
+
-
-
Self-learning!

Drift

Tracked Patches

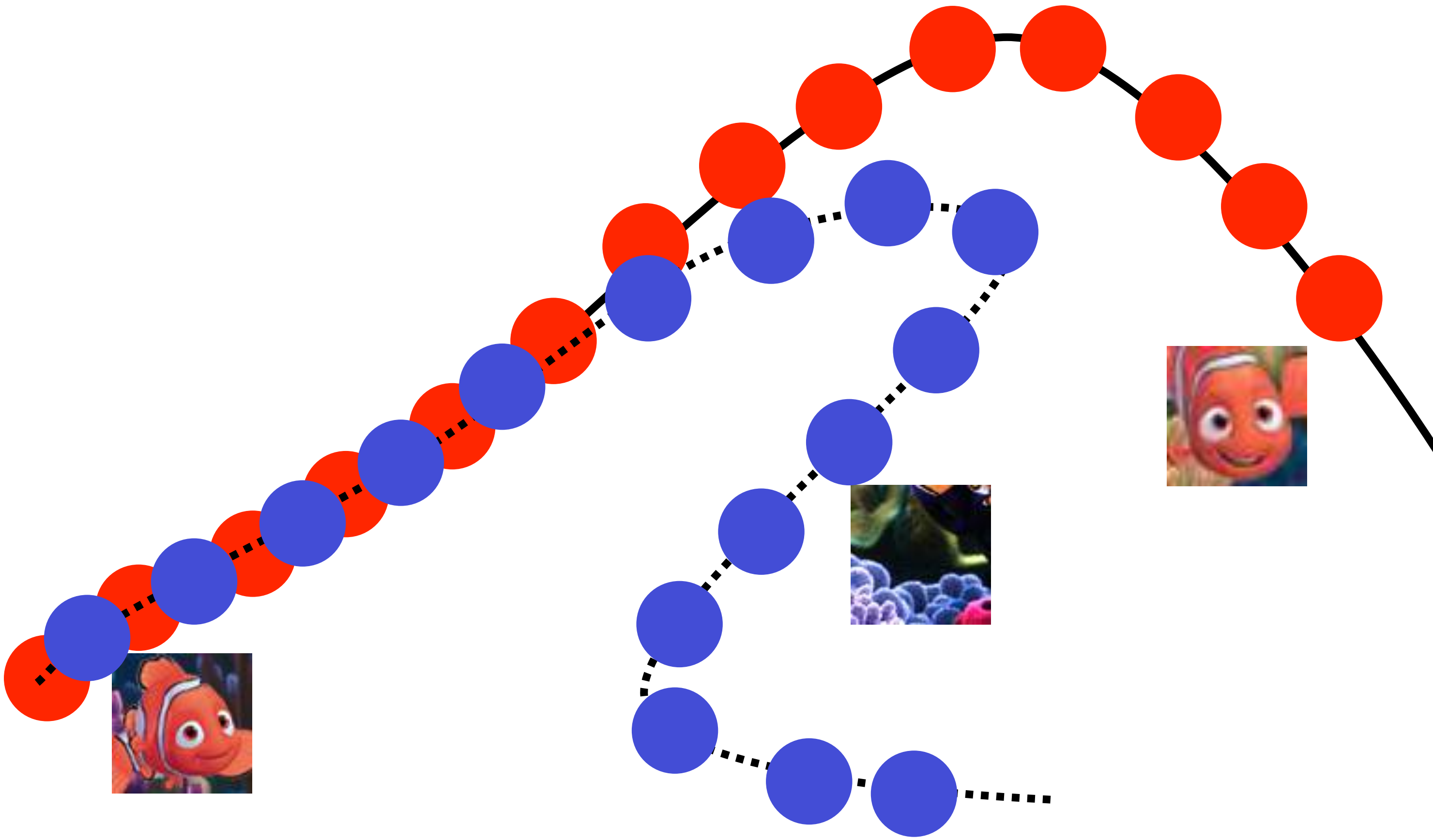


Confidence



Computer Vision

Drift



Computer Vision

Outline

Feature

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)

Model

- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class (multiple object tracking)
- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)

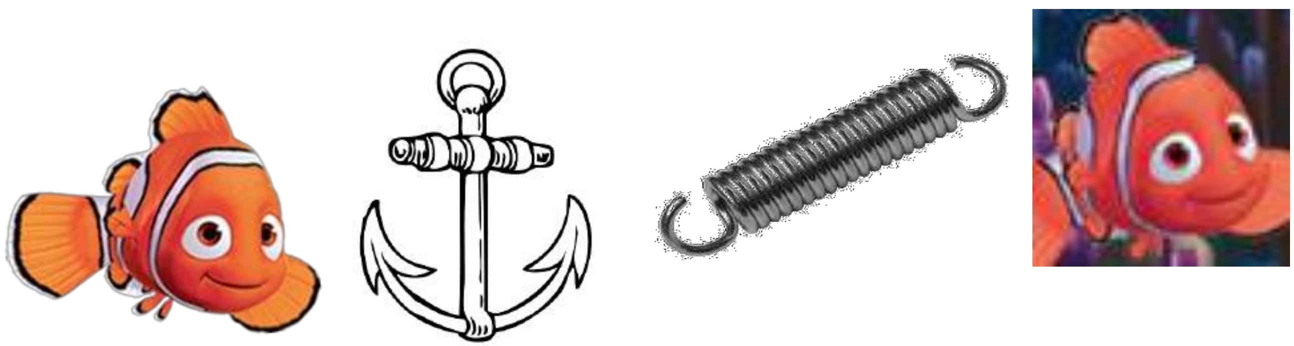
Combining Tracking and Detection

(to avoid drift)

Refining an object model

- Only thing we are sure about the object is its initial model (e.g. appearance in first frame)
- We can “anchor” / correct our model with that
- This can limit drift

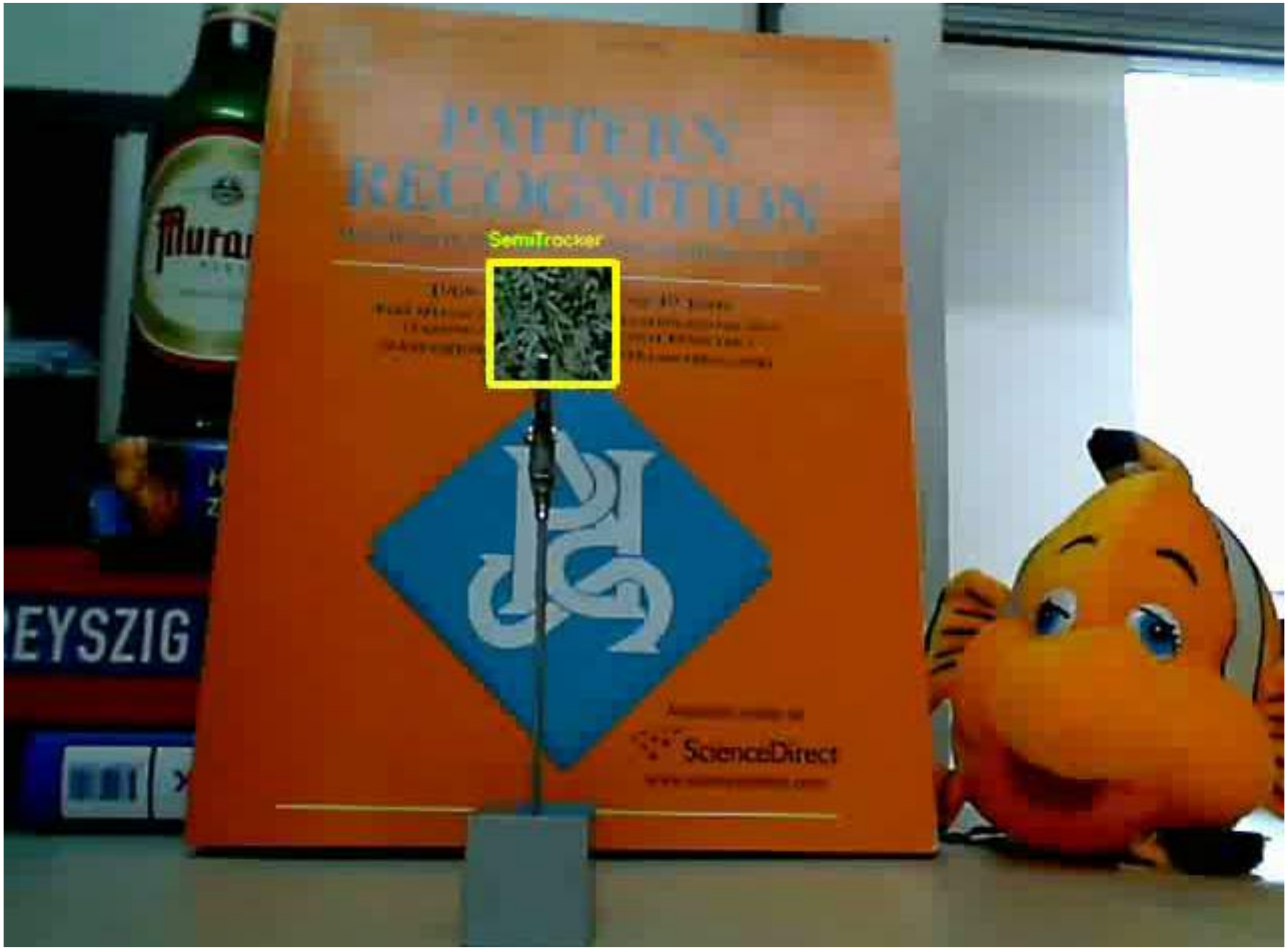
Current Model



Fix (initial) Model

Recover from Drift

using a fixed/anchor model (e.g. first frame)



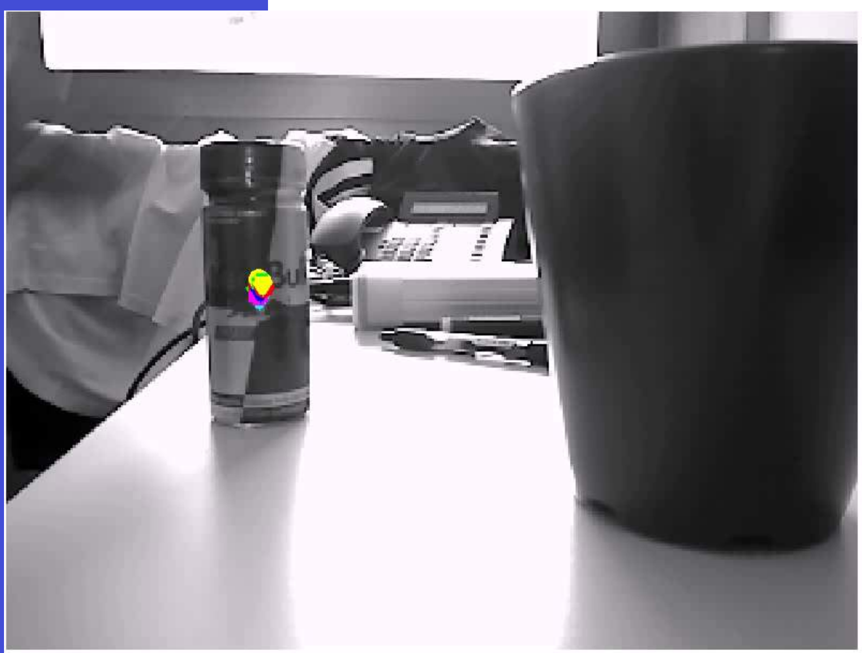
[Grabner et al. ECCV'08]

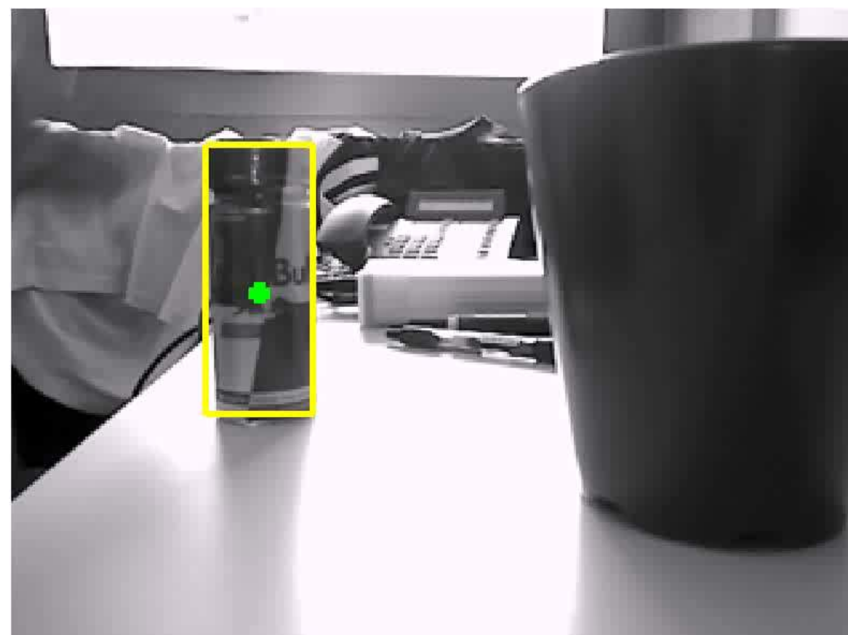
Context in Tracking

Humans use context to track

- … objects which change there appearance very quickly.
- … occluded objects or object outside the image.
- … small and/or low textured objects or even “virtual points”.

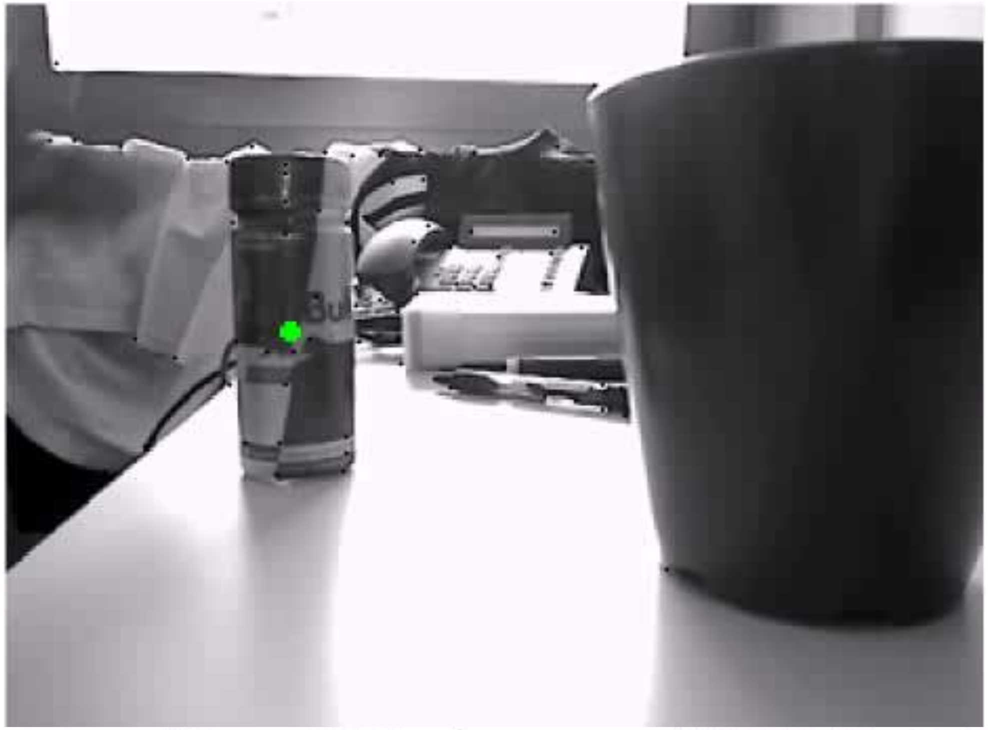
Computer Vision





Computer Vision

Using Supporters



Assumptions should hold

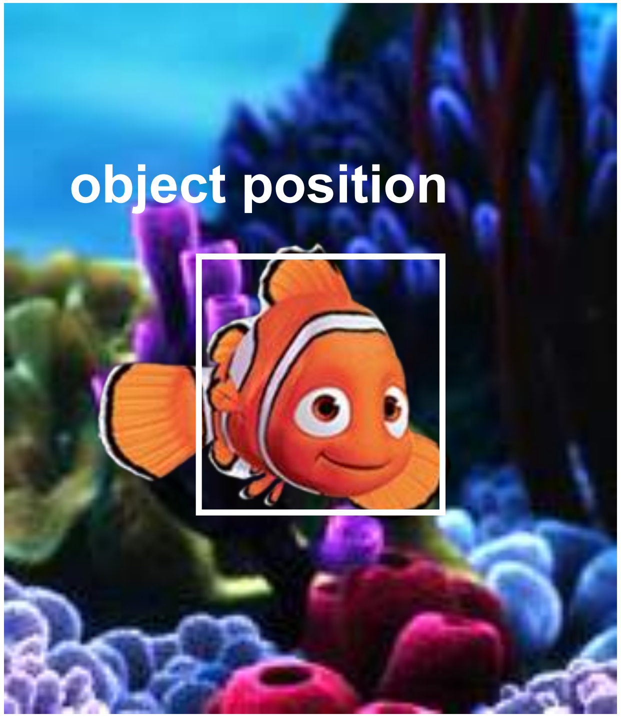
With Supporters

Problems in Tracking

Tracking Issues

- Initialization

Time t = 0



Tracking Issues

- Obtaining observation…
  - Generative: "render" the state on top of the image and compare
  - Discriminative: classifier or detector score
- …and dynamics model
  - specify using domain knowledge
  - learn (very difficult)

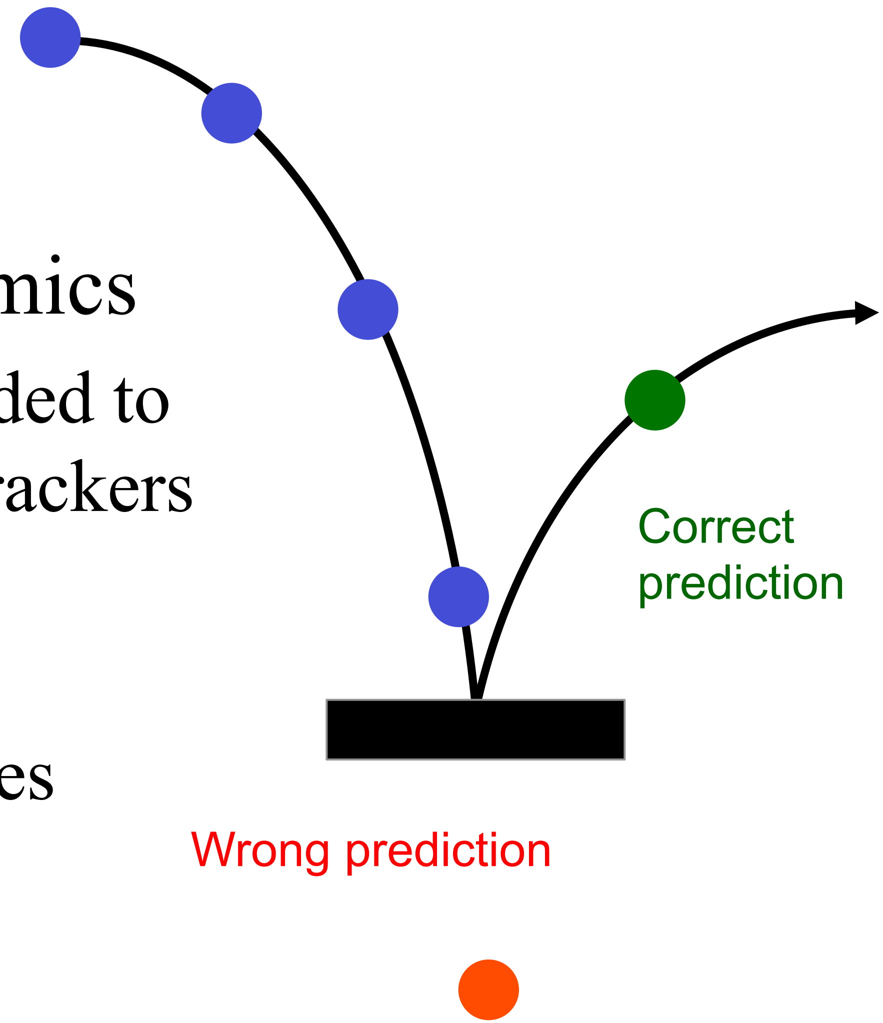
Computer Vision

Tracking Issues

- Model- vs. Model-free-Tracking

Tracking Issues

- Nonlinear dynamics
  - Sometimes needed to keep multiple trackers in parallel
  - E.g., for abrupt direction changes („Persons“)



Tracking Issues

- Prediction vs. Correction (cf. Kalman Filtering)
  - If the dynamics model is too strong, tracking will end up ignoring the data.
  - If the observation model is too strong, tracking is reduced to repeated detection.

08.10.2009
<< Rudolf Kalman, ETH-Zurich emeritus professor of mathematics, is awarded the National Medal of Science by Barack Obama – one of the highest accolades for researchers in the USA.

In January 2008, Hungarian-born Kalman received the Charles Draper Prize, which is regarded as the “Nobel Prize” of the engineering world. >>

http://www.ethlife.ethz.ch/archive_articles/091008_kalman_per/index

Tracking Issues

- Data Association –
  Multiple Object Tracking
  - What if we don't know which measurements to associate with which tracks?

Tracking Issues

- Data Association –
  Occlusions / Self Occlusions

Tracking Issues

- Data Association – Fast Motion

Tracking Issues

- Data Association – Background / Appearance Change
  - Cluttered Background
  - Changes in shape, orientation, color,…

Tracking Issues

- Drift
  - Errors caused by dynamical model, observation model, and data association tend to accumulate over time

Summary

Feature

- Region Tracking (and Mean Shift Algorithm)
- Point Tracking (and Aperture Problem)
- Template Tracking (Lucas-Kanade)

Model

- Tracking-by-Detection
  - a specific target (e.g., keypoints + Ransac)
  - object class (multiple object tracking)
- Model-based Body Articulation
- On-line Learning

- Misc (preventing drift, context, issues)

Let’s apply

Q. What tracking method would you use in each following application scenario?

What limitations you may expect?

Task: “Discuss one (or more) in groups”

App1. Safety: In a lumbar mill, you wish to use CV to stop the blade if a hand reaches nearby.

App2. Medical: You wish to track the motion of an ultrasound probe, to relate images in space.

App3. Autonomous driving: Tracking other nearby vehicles to adjust speed and course

AppX. Your favorite tracking app