Object Tracking

Visual Understanding

What you will learn in this course

Low-Level Vision

Image Processing:
Photometric Image formation
Point Operations
Histogram Equalization
Linear & Non-Linear filters
Convolution & Cross correlation

Feature Detection:
Edge detection
Corner detection
Line detection

Feature Description:
SIFT
Feature Matching
Image Stitching
Panorama

Geometric Vision

Transformation and Camera Mode:
Geometric Image Formation
Pinhole Camera Model

Camera Calibration:
Estimating intrinsics
Estimating extrinsics

Visual Understanding

Object Detection & Tracking:
Single-stage detectors
Two-stage detectors
YOLO
Object tracking

Image Segmentation:
Traditional Image Segmentation
Learning-based Image Segmentation

Machine Learning for CV

Machine Learning for Classification:
k-NN
Linear Classifier

Deep Learning:
Optimization
Neural Network
CNN
CNN architectures:

Generative & Geometric Learning

Generative Models:
AE, VAE, GAN

3D Vision and 3D Deep Learning:
3D shape representations
Geometric deep learning overview
3D Deep Learning models : pointnet

What we will learn today

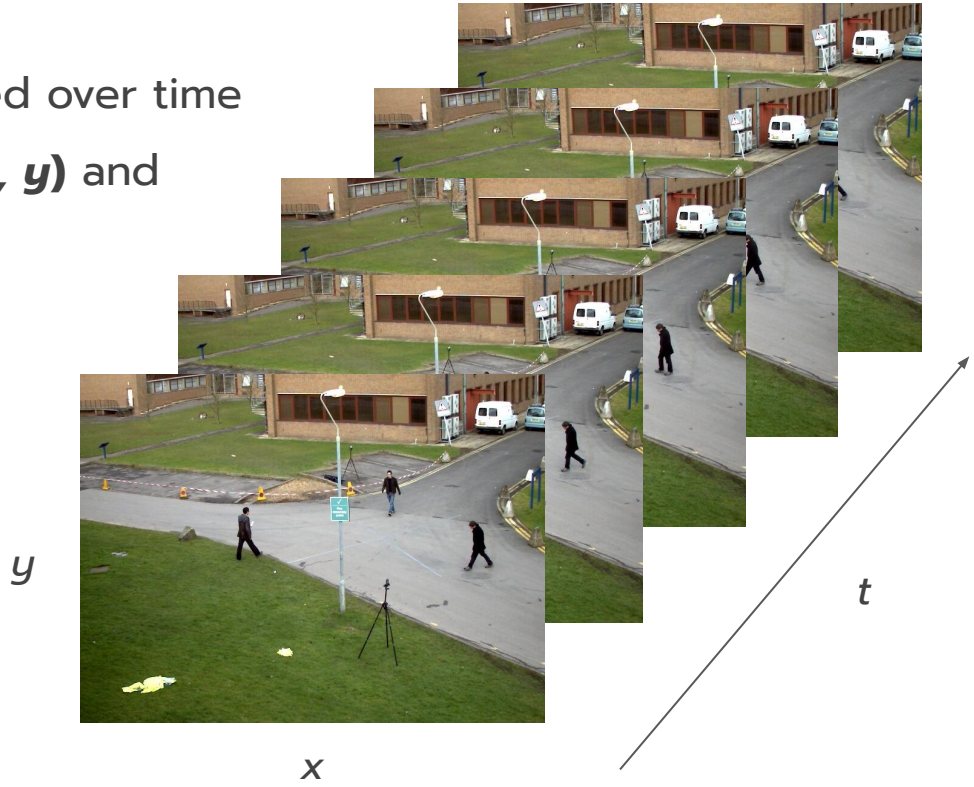
- Change Detection
- Tracking-by-Detection
- Tracking-by-Dynamics
- Multiple Object Detection

What is Object Tracking?

How can we track an object's motion over time?

A video is a sequence of frames captured over time

Our image data is a function of **space (*x*, *y*)** and **time (*t*)**



Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

What is Object Tracking?

Goal

Estimate the ***number*** and state of objects in a region of interest

Number

1: Single-target tracking

0 or 1: Detection and tracking

N: Multi-target detection and tracking

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

What is Object Tracking?

Goal

Estimate the number and ***state*** of objects in a region of interest

State

We are using the term state to describe a vector of quantities that characterize the object being tracked.

$[x, y]$	(location)
$[x, y, dx, dy]$	(location + velocity)
$[x, y, appearance\text{-}params]$	(location + appearance)

Because our observations will be noisy, estimating the state vector will be a statistical estimation problem.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

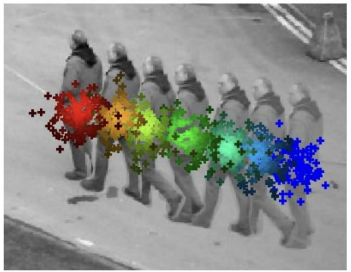
What is Object Tracking?

Goal

Estimate the number and state of ***objects*** in a region of interest

Objects

- We will look at a large variety of objects to track.
- They can be given by a user or detected automatically.
- Very interesting are people.
- Special case: Tracking the camera pose wrt. the environment/object



Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

What is Object Tracking?

Goal

Estimate the number and state of objects in a region of interest

What distinguishes tracking from "typical" statistical estimation (or machine learning) problems?

- Typically a strong temporal component is involved.
- Estimating quantities that are expected to change over time (thus, expectations of the dynamics play a role).
- Interested in current state S_t for a given time step t.
- Usually assume that we can only compute information seen at previous time steps 1, 2, …, *t*-1. (Can't look into the future!)
- Usually we want to be as efficient as possible, even "real-time".

These concerns lead naturally to recursive estimators.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

What is Object Tracking?

Why do we need Tracking?

- To **model** objects when detection fails:
  - Occlusions
  - Viewpoint/pose/blur/illumination variations (in a few frames of a sequence)
  - Background clutter
- To reason about the **dynamic** world, e.g., trajectory prediction (is the person going to cross the street?)

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

What is Object Tracking?

Tracking can be formulated as

- Similarity measurement
- Correlation
- Correspondence
- Matching/retrieval
- Data association

Tracking is also about learning **to model** our targets. We can learn to model our targets in terms of

- **Appearance** : to know how the target looks like
  - Single object tracking, Re-Identification (Multiple Object Tracking)
- **Motion** : to make predictions of where the target goes.
  - Trajectory prediction

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Elements of Tracking

- **Detection** : Find the object(s) of interest in the image.

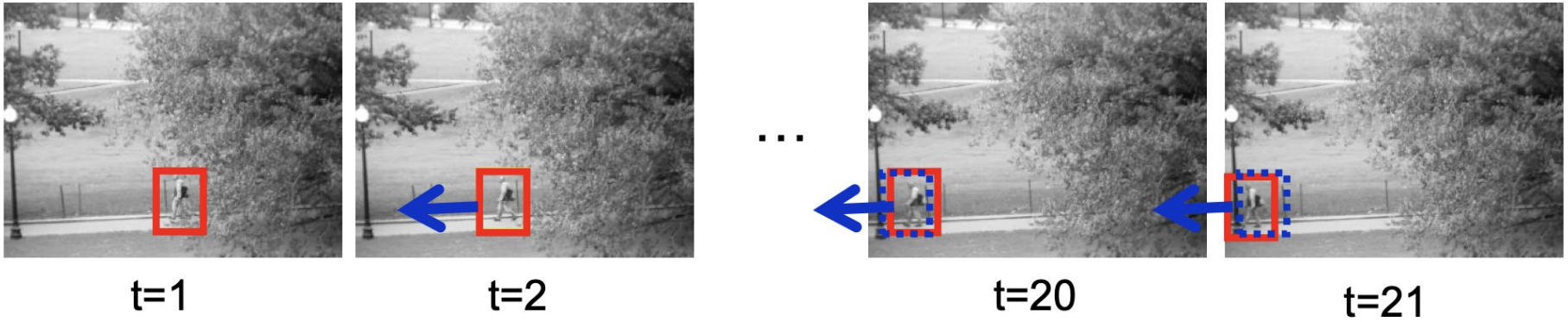
Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Elements of Tracking

- **Detection** : Find the object(s) of interest in the image.
- **Association** : Determine which observations come from the same object.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Elements of Tracking

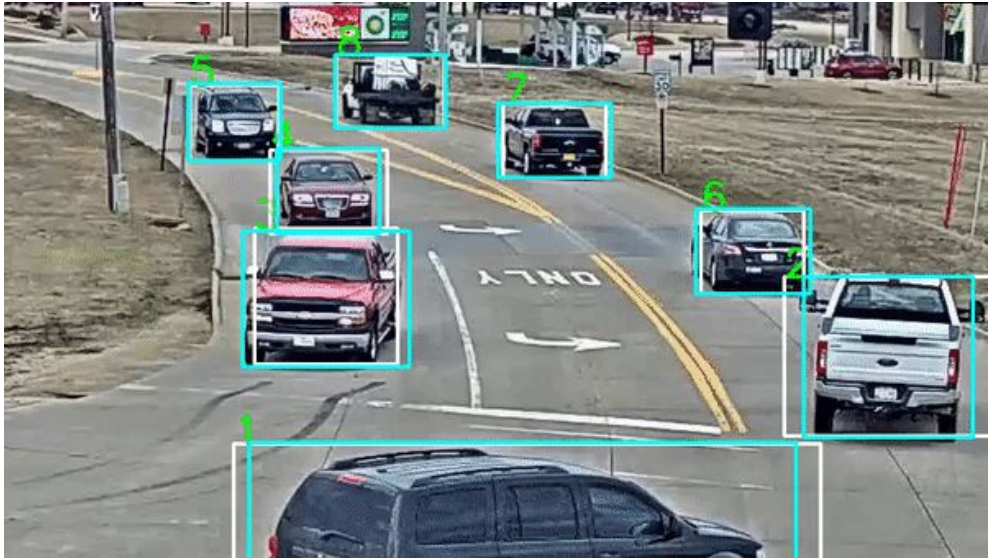


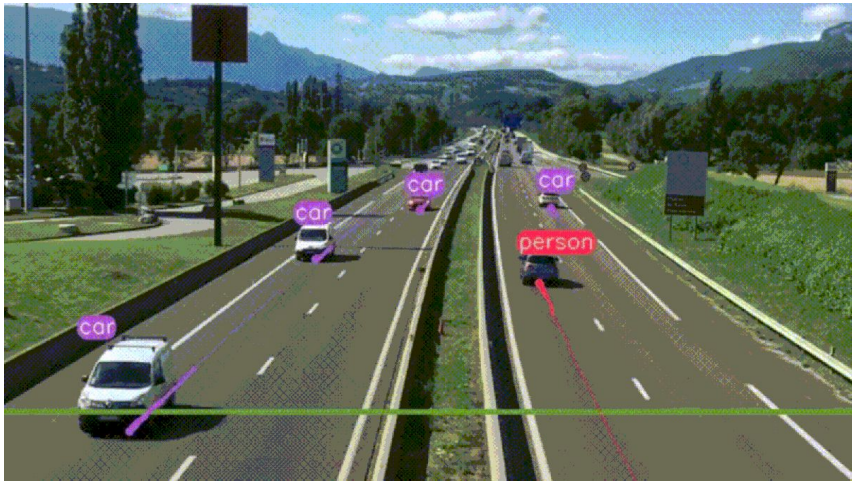
- **Detection** : Find the object(s) of interest in the image.
- **Association** : Determine which observations come from the same object.
- **Prediction** :
  - Predict future motion based on the observed motion pattern.
  - Use this prediction to improve detection and data association in later frames.

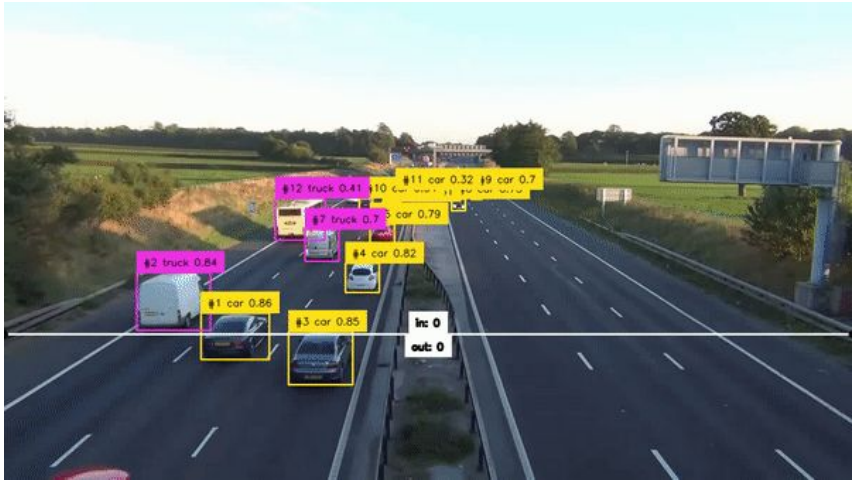
Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Applications

Traffic Monitoring

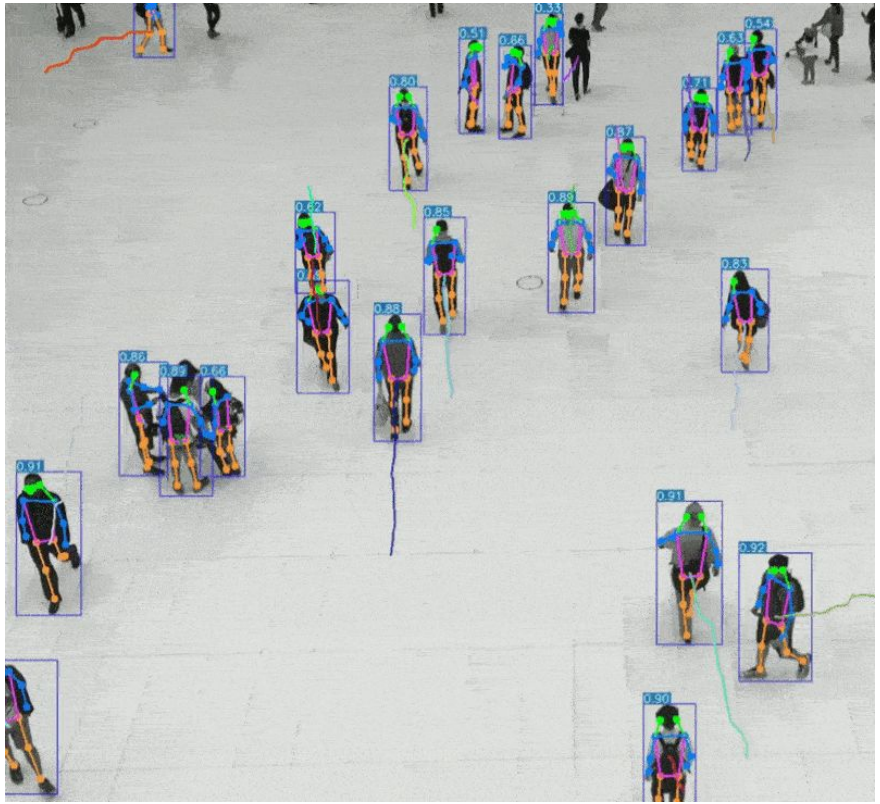


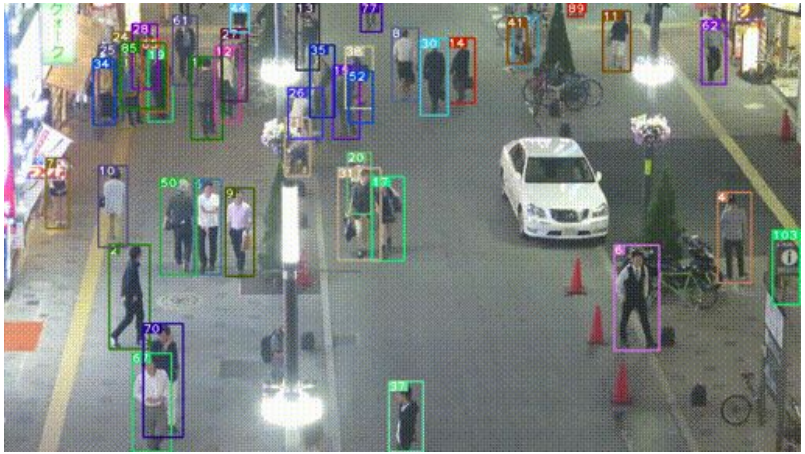


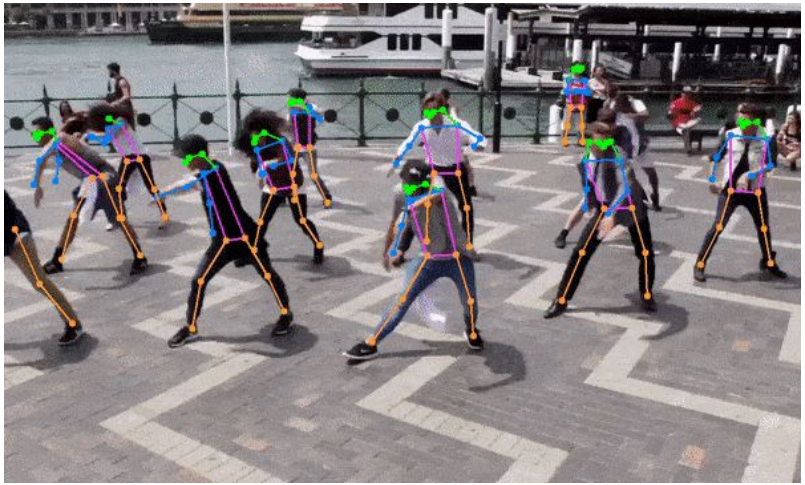


Applications

Surveillance







Applications

Self Driving Car

Applications

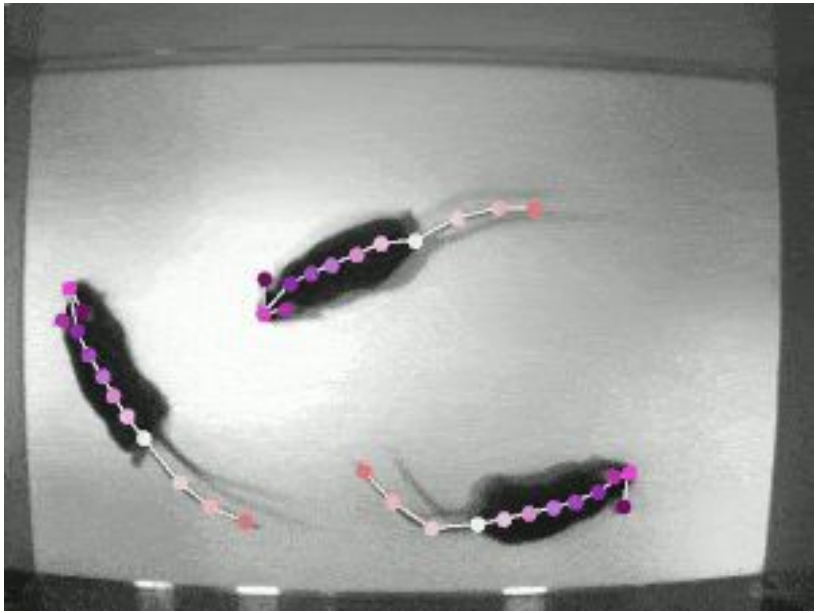
Sport Analysis

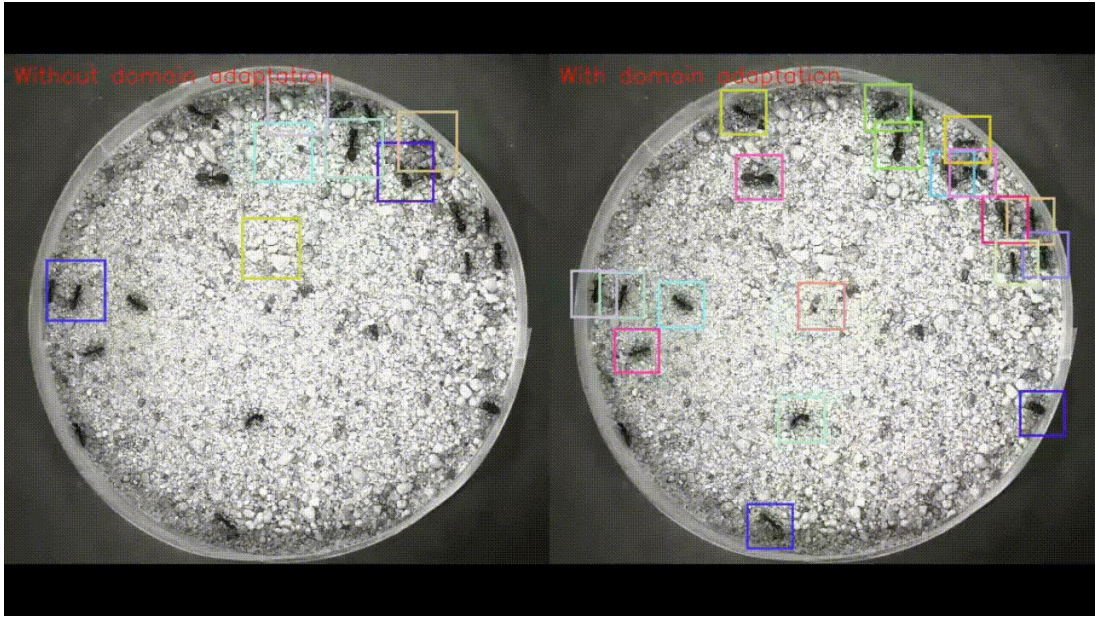
Applications

Video Analysis : Customer Behavior for In-Store Analytics

Applications

Video Analysis : Animal Behavior Analysis





Object Tracking

Change Detection

Background Modeling

Change Detection

Given: Static cameras observing scene (room, street, etc.)
Find: Meaningful changes (moving objects, people, etc.)

Robust and real-time classification of each pixel as "foreground" (motion/change) or "background" (static).

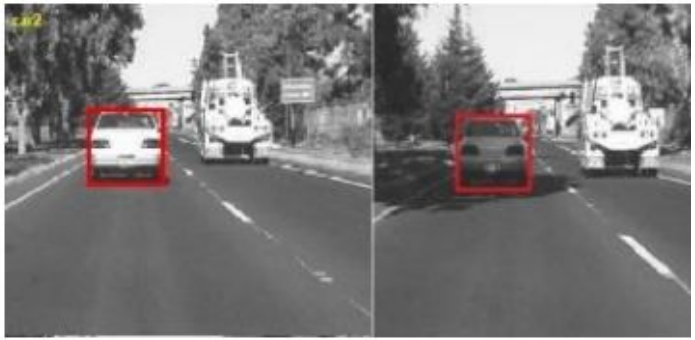
Background Modeling

Change Detection

Challenges

Ignore uninteresting changes:

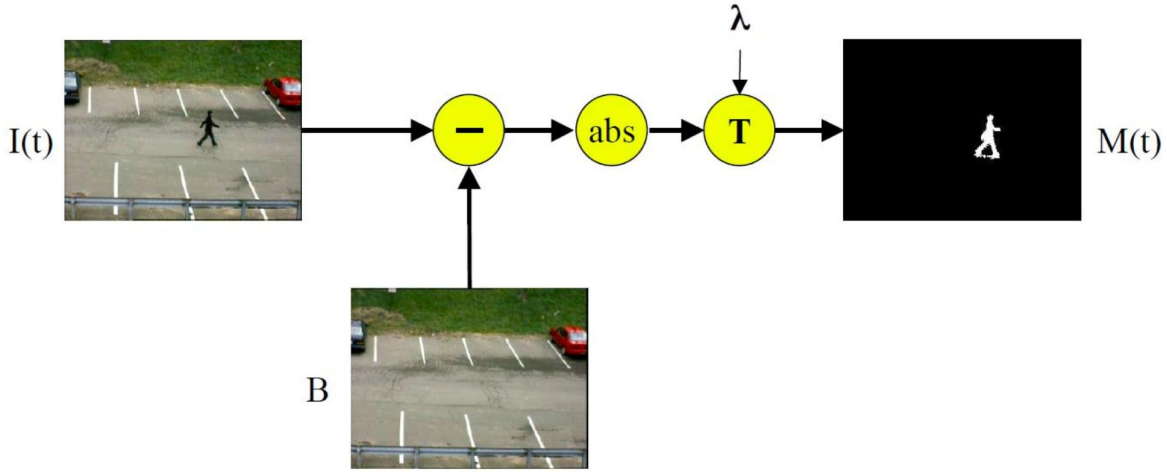
- Background fluctuations
- Image noise
- Rain, snow, turbulence
- Illumination changes & shadows
- Camera shake



Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Background Modeling

Simple Background Subtraction



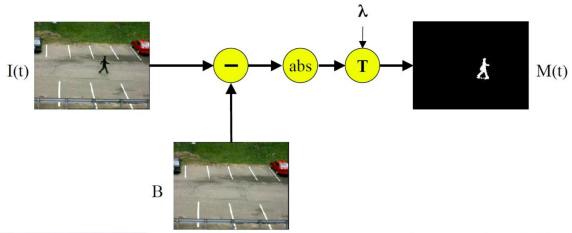
Procedure

- Background model is a static image (without any objects).
- Pixels are labeled based on thresholding the absolute intensity
- difference between current frame and background.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Simple Background Subtraction: Results

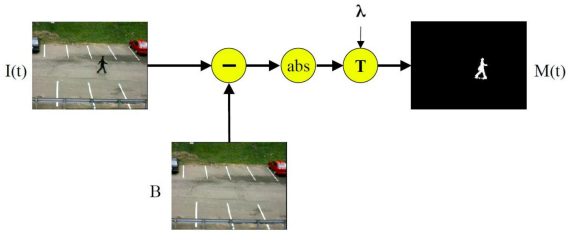


- **Observation**
  - Background subtraction **does a reasonable job** of extracting the object shape if the **object intensity/color is sufficiently different** from the background.
- **What are the limitations of this simple procedure?**

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Simple Background Subtraction : Limitations



Outdated reference frame

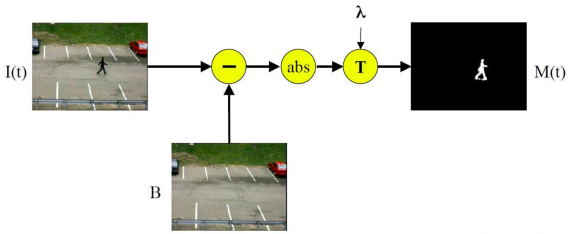
- Objects that enter the scene and stop continue to be detected, making it difficult to detect new objects that pass in front of them.

- If part of the assumed static background starts moving, both the object and its negative ghost (the revealed background) are detected.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

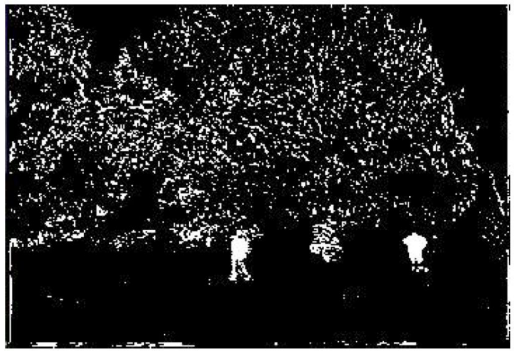
Background Modeling

Simple Background Subtraction : Limitations



Illumination changes

- Background subtraction is **sensitive to illumination changes** and unimportant scene motion (e.g., tree branches swaying in the wind).



Global threshold

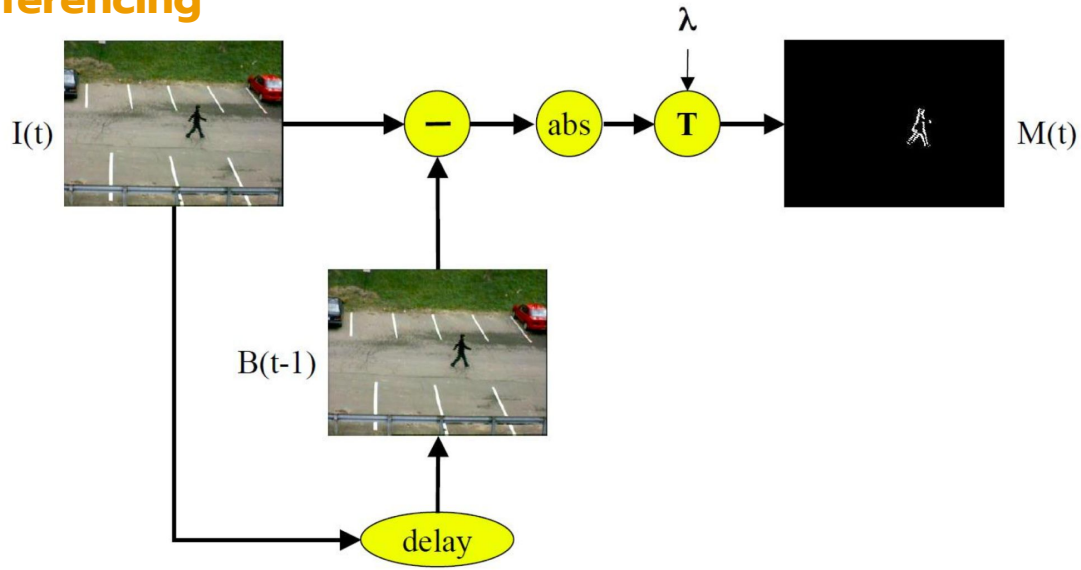
- A single, global threshold for the entire scene is often suboptimal.

Need adaptive model with local decisions

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Simple Frame Differencing



Other idea

- Background model is replaced with the previous image.

Background Modeling

Simple Frame Differencing

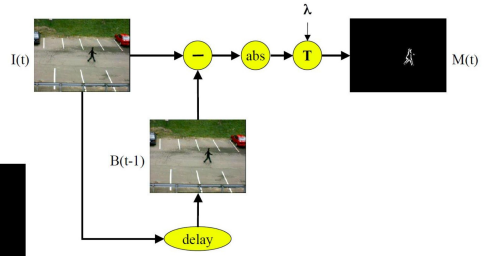
Input video

Frame Difference

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Background Modeling

Simple Frame Differencing: Observations



Advantages

- Frame differencing is very quick to adapt to changes in lighting or camera motion.
- Objects that stop are no longer detected.
- Objects that start up no longer leave behind ghosts.

Limitations

- Frame differencing only **detects the leading and trailing edge** of a uniformly colored object.
- Very few pixels on the object are labeled.
- Very hard to detect an object moving towards or away from the camera.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Differencing and Temporal Scale

Note what happens when we adjust the temporal scale (frame rate) at which we perform two-frame differencing

Define $D(N) = \| I(t) - I(t+N) \|$

Effect of increasing the temporal scale

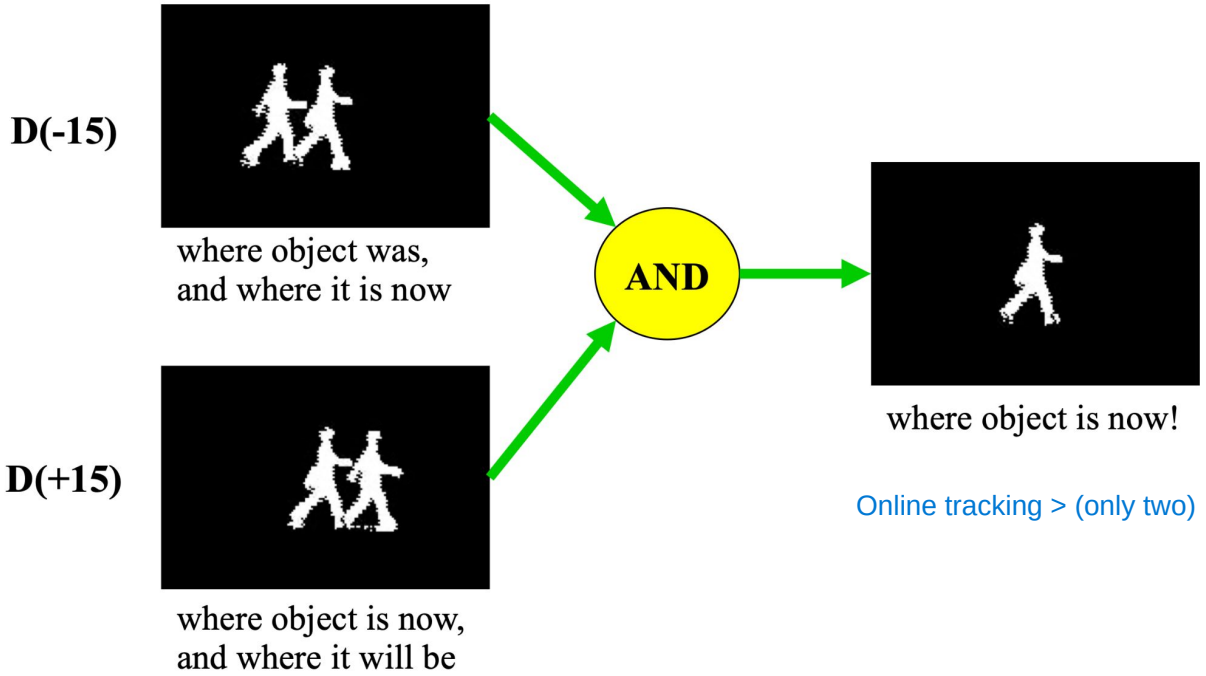
- More complete object silhouette, but two copies of the object (one where it used to be, one where it is now).

Background Modeling

Three-Frame Differencing

Offline tRACKING (HAVE EVERYTHING)

Improved approach to handle this problem (two copies of the object).



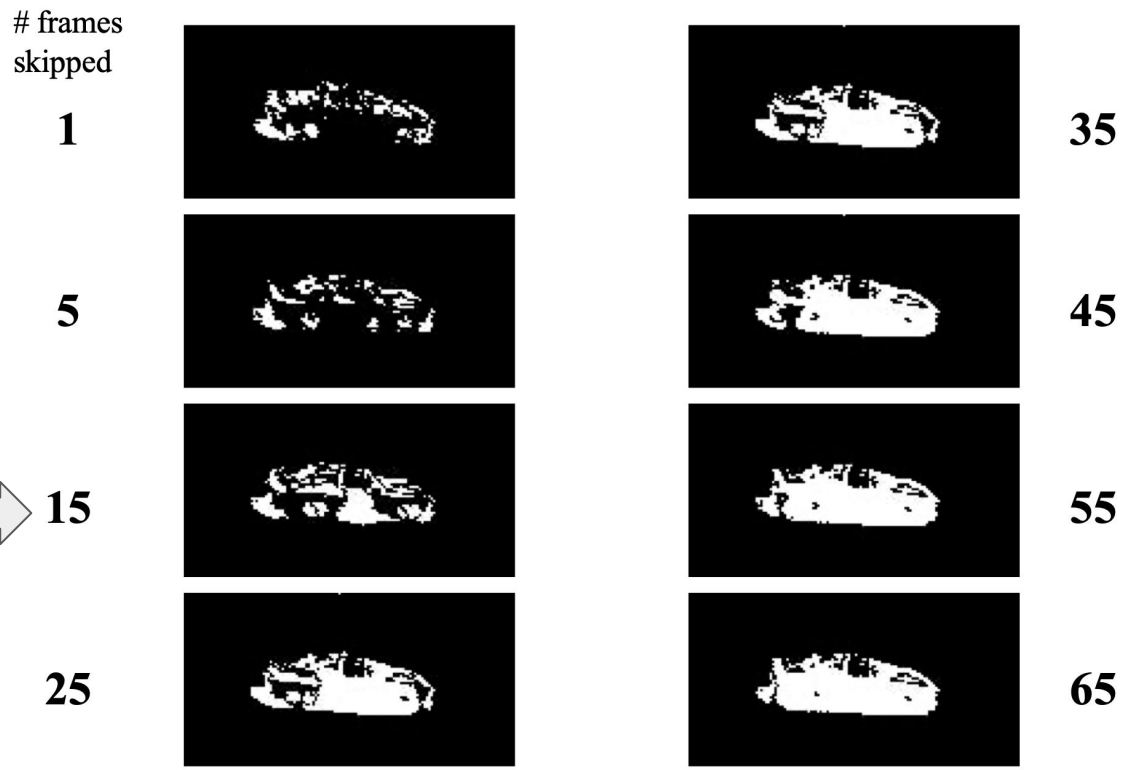
Online tracking > (only two)

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Three-Frame Differencing

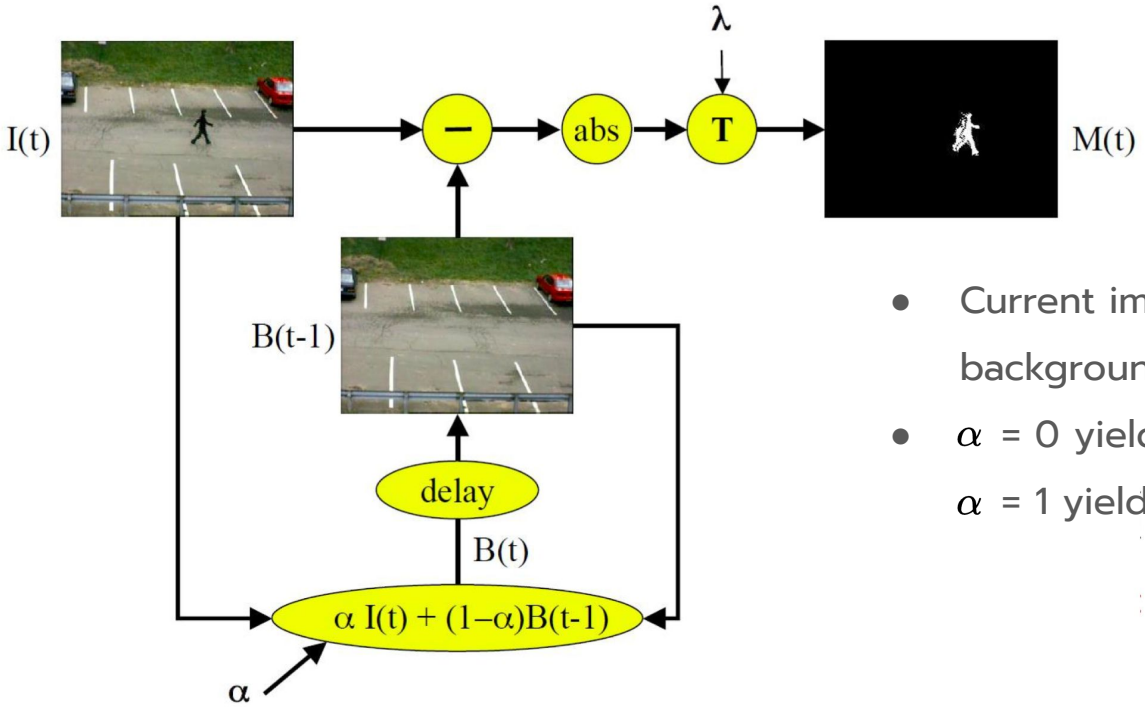
Choice of good frame-rate for three-frame differencing depends on the **size** and **speed** of the object



This worked well for the person

Background Modeling

Adaptive Background Subtraction



- Current image is "blended" into the background model with parameter α
- α = 0 yields simple background subtraction,
  α = 1 yields frame differencing

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Adaptive Background Subtraction

Properties

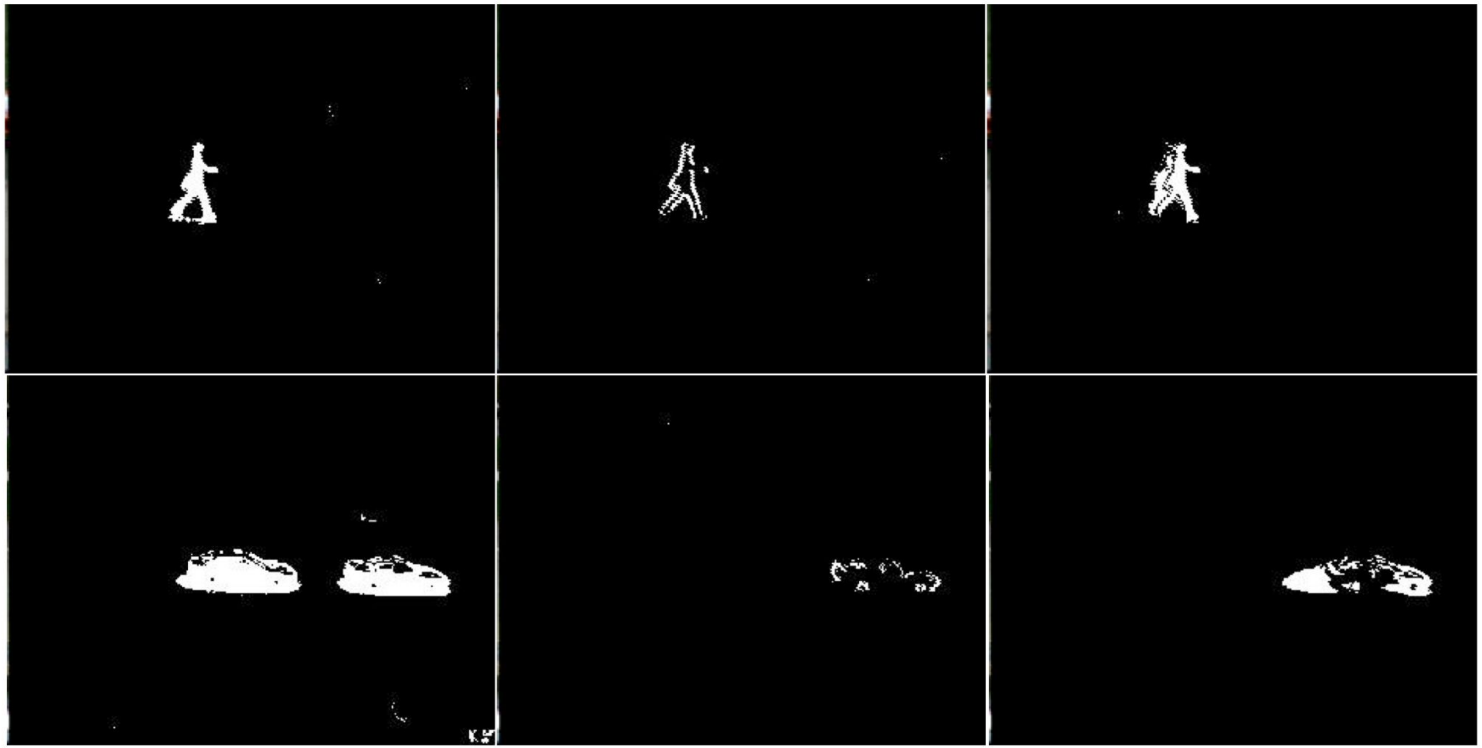
- More responsive to changes in **illumination** and **camera motion**.
- Small, fast-moving objects are well-segmented, but they leave behind short "trails" of pixels.

- Objects that stop and ghosts left behind by objects that start both gradually fade into the background.
- The centers of large, slow-moving objects start to fade into the background, too!
- This can be fixed by decreasing the blend parameter α, but then it takes longer for ghost objects to disappear.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Comparison



Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Background Modeling

Average

Background B

$average\{I_1, I_2, \ldots, I_k\}$

First *K* frames

Input Frame I_t

Foreground F_t

$F_t = |I_t - B| > T$

Cannot handle change in lighting, background, etc.

Background Modeling

Median

Background B

$median\{I_1, I_2, \ldots, I_k\}$

First *K* frames

Input Frame I_t

Foreground F_t

$F_t = |I_t - B| > T$

Cannot handle change in lighting, background, etc.

Background Modeling

Moving Median: Building a simple Adaptive model of background over time.

Background B

$$median\{I_{t-1}, I_{t-2}, \ldots, I_{t-k}\}$$

Last *K* frames

Input Frame I_t

Foreground F_t

$$F_t = |I_t - B| > T$$

Requires keeping the last *K* frames in memory.
Finding median for each pixel is expensive.

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Background Modeling

Moving Median: Building a simple Adaptive model of background over time.

Background B

$$median\{I_{t-1}, I_{t-2}, \ldots, I_{t-k}\}$$

Last *K* frames

Input Frame I_t

Foreground F_t

$$F_t = |I_t - B| > T$$

Cannot handle significant pixel fluctuations
(weather, shadows, shakes, etc.)

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Background Modeling

Background subtraction / Frame differencing

- Very simple techniques, historically among the first.
- Straightforward to implement, fast to test out.
- We've seen some fixes for the most pressing problems.

Remaining limitations

- Rather heuristic approach.
- Leads to relatively poor foreground/background decisions.
- Optimal temporal scale still depends on object size and speed.
- Global threshold is often suboptimal for parts of the image.

  ⇒ Very fiddly in practice, requires extensive parameter tuning.

Let's try to come up with a better founded approach

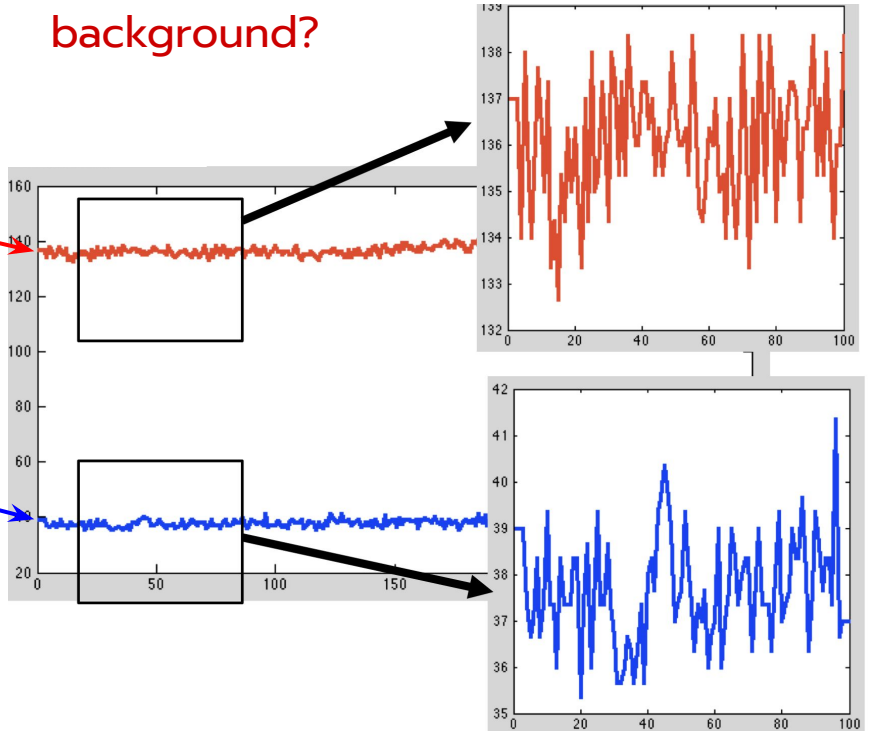
⇒ Using a statistical model of background probability…

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Gaussian Mixture Model

Intensity distribution at each pixel over time:

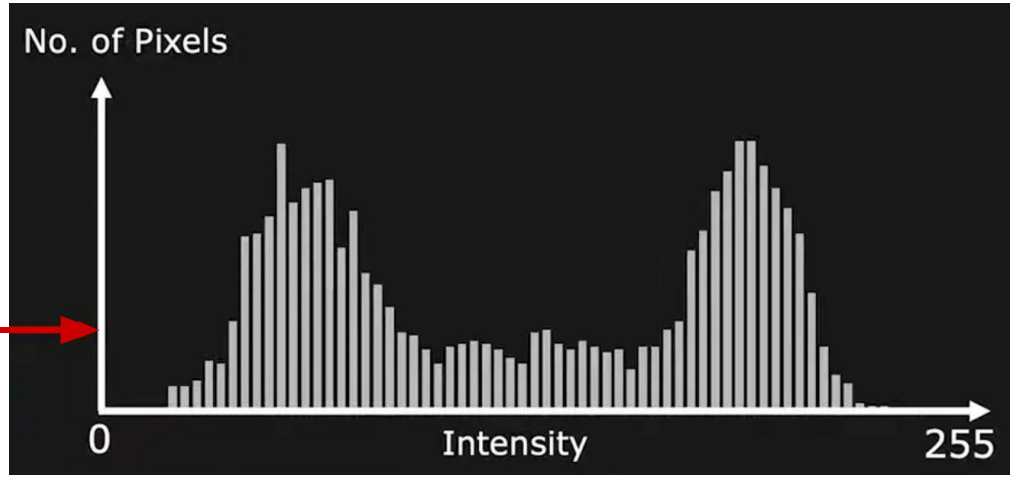
Q: What is a good statistical model of the value of a pixel in the unchanging background?



Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Intensity distribution at each pixel over time:



Intensity histogram for a pixel over time

Intensity variations due to static scene (road), noise (snow), and occasional moving objects (vehicles).

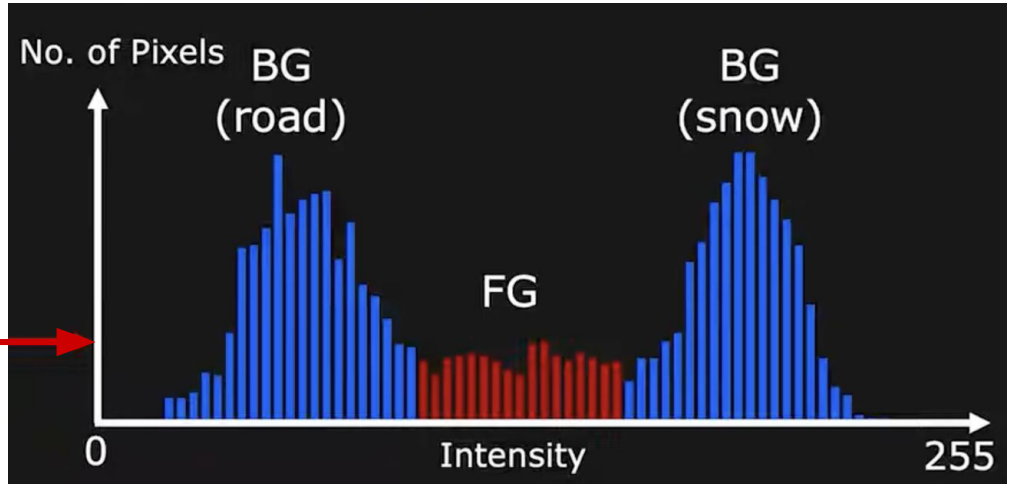
Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Pixel are more on the BG than FG

Intensity distribution at each pixel over time:

Intuition : Pixels are background most of the time.



Intensity histogram for a pixel over time

Intensity variations due to static scene (road), noise (snow), and occasional moving objects (vehicles).

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Gaussian Model

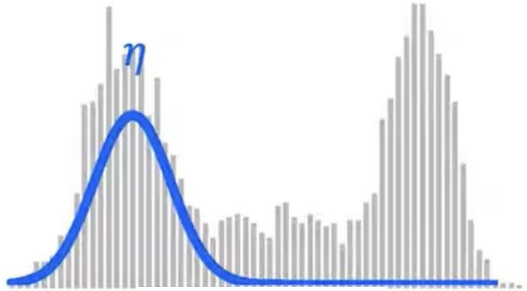
Probability Distribution
P(x) (*x*: pixel intensity)

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Gaussian Model

Probability Distribution
P(x) (*x*: pixel intensity)



Gaussian
$\omega, \eta(x, \mu, \sigma)$

1-Dimensional Gaussian:

$$\omega\,\eta(x,\mu,\sigma) = \omega\frac{1}{\sqrt{2\pi}\sigma}\,e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

μ: Mean

σ: Std. Deviation

ω: Scale (Evidence)

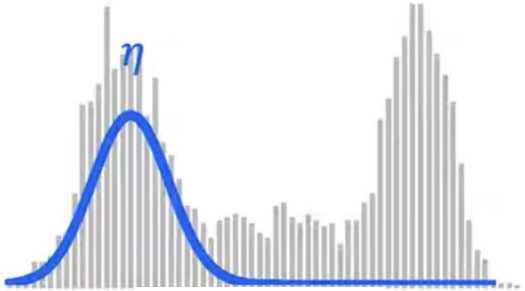
Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

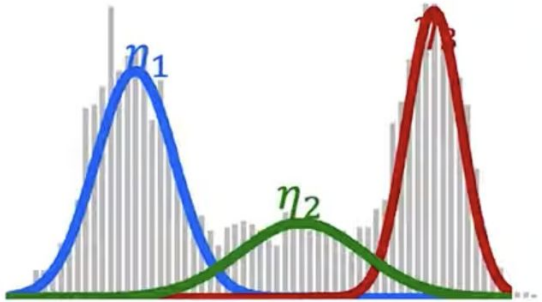
Gaussian Model

Assume *P(x)* is made of *K* different Gaussians.

Probability Distribution
P(x) (*x*: pixel intensity)



Gaussian
$\omega, \eta(x, \mu, \sigma)$



Mixture of Gaussians
$\omega_k \eta_k(x, \mu_k, \sigma_k)$

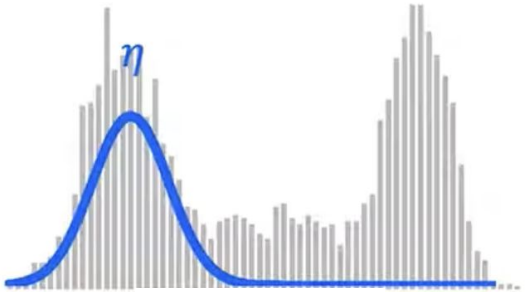
Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Gaussian Model

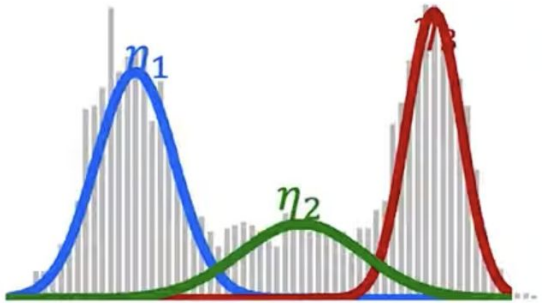
Assume *P(x)* is made of *K* different Gaussians.

Probability Distribution
P(x) (*x*: pixel intensity)



Gaussian
$\omega, \eta(x, \mu, \sigma)$

K=3

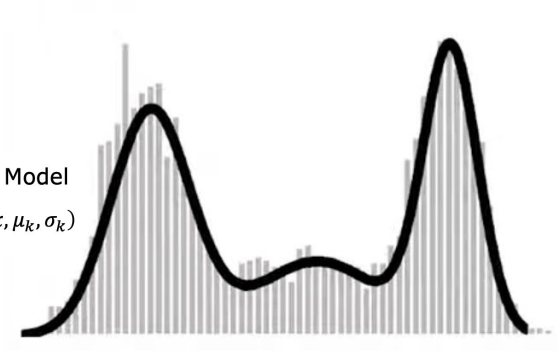


Mixture of Gaussians
$\omega_k \eta_k(x, \mu_k, \sigma_k)$

GMM Distribution: Weighted sum of *K* Gaussians

$$P(x) \cong \sum_{k=1}^{K} \omega_k \eta_k(x, \mu_k, \sigma_k) \quad \text{such that} \quad \sum_{k=1}^{K} \omega_k = 1$$

Gaussian Mixture Model
$P(x) = \sum_{k=1}^{K} \omega_k \eta_k(x, \mu_k, \sigma_k)$



Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

High Dimensional GMM

Let $P(\mathbf{X})$ be a probability distribution of a D-dimensional random variable $\mathbf{X} \in \mathcal{R}^D$. For example: $\mathbf{X} = [r, g, b]^T$

GMM of $P(\mathbf{X})$: Sum of K D-dimensional Gaussians

$$P(\mathbf{X}) \cong \sum_{k=1}^{K} \omega_k \eta_k(\mathbf{X}, \boldsymbol{\mu}_k, \Sigma_k) \quad \text{such that} \quad \sum_{k=1}^{K} \omega_k = 1$$

where: $$\eta(\mathbf{X}, \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu})^T(\Sigma)^{-1}(\mathbf{X}-\boldsymbol{\mu})}$$

Mean $\boldsymbol{\mu} = \begin{bmatrix} \mu_r \\ \mu_g \\ \mu_b \end{bmatrix}$ Covariance matrix $\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$ (can be a full matrix)

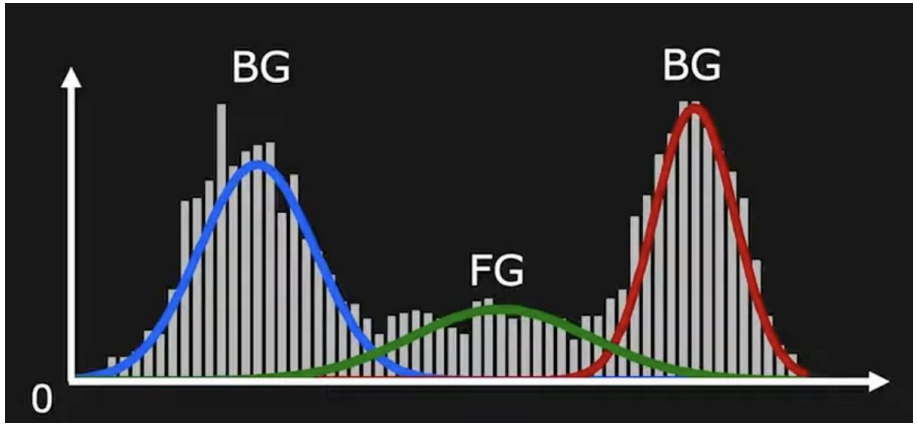
GMM can be estimated from $P(\mathbf{X})$.

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Background Modeling with GMM

Given: A GMM for intensity/color variation at a pixel over time.
Classify: Individual Gaussians as foreground/background



Intensity histogram for a pixel over time

Intuition: Pixels are background most of the time. That is, Gaussians with large supporting evidence ω and small σ .

height

Large: $\frac{\omega}{\sigma}$ Background

Small: $\frac{\omega}{\sigma}$ foreground

standard deviation (width of disctibution)

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Change Detection using GMM

For each pixel:

1. Compute pixel color histogram H using first N frames.
2. Normalize histogram: $\hat{H} \leftarrow H/\|H\|$.
3. Model $\hat{H}$ as mixture of K (3 to 5) Gaussians.
4. For each subsequent frame:
   a. The pixel value $\mathbf{X}$ belongs to Gaussian k in GMM for which $\|\mathbf{X} - \boldsymbol{\mu}_k\|$ is minimum and $\|\mathbf{X} - \boldsymbol{\mu}_k\| < 2.5\sigma_k$.
   b. If ${}^{\omega_k}/_{\sigma_k}$ is large then classify pixel as background. Else classify as foreground.
   c. Update histogram H using new pixel intensity.
   d. If $\hat{H}$ and $H/\|H\|$ differ a lot ($\left\|\hat{H} - {}^{H}/_{\|H\|}\right\|$ is large), $\hat{H} \leftarrow H/\|H\|$ and refit GMM.

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Adaptive GMM based Change Detection

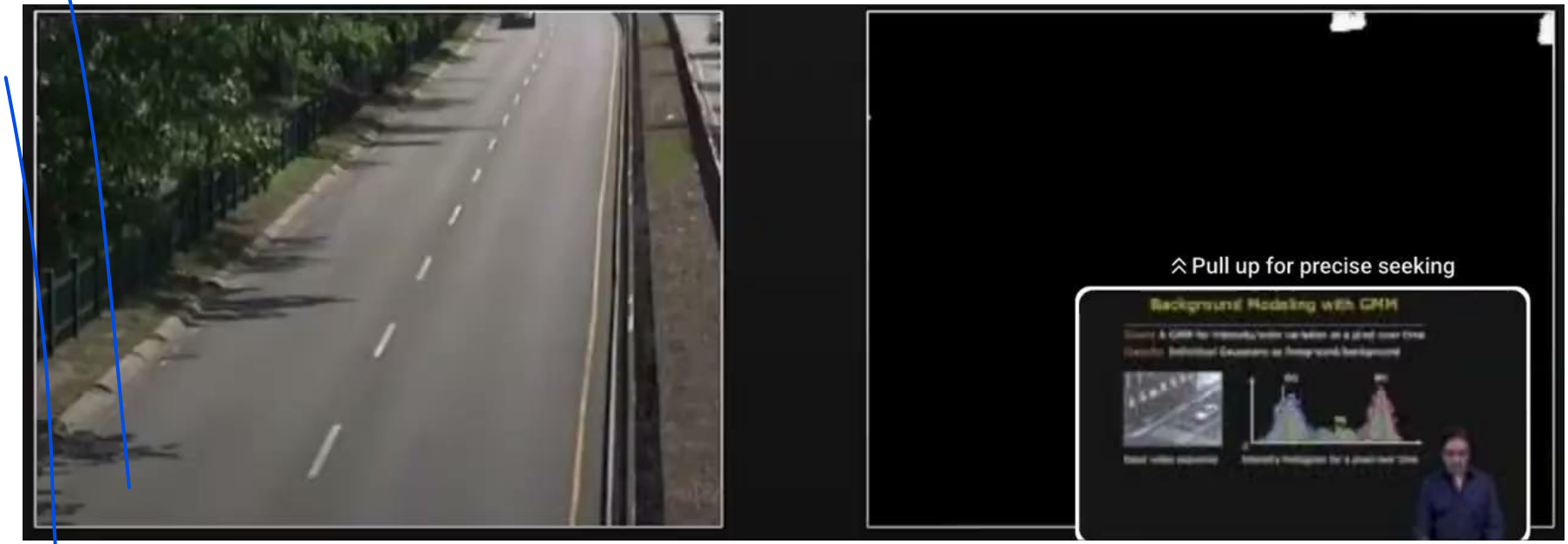
Input Video

Foreground

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Adaptive GMM based Change Detection



Input Video

Foreground

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Gaussian Mixture Model

Adaptive GMM based Change Detection

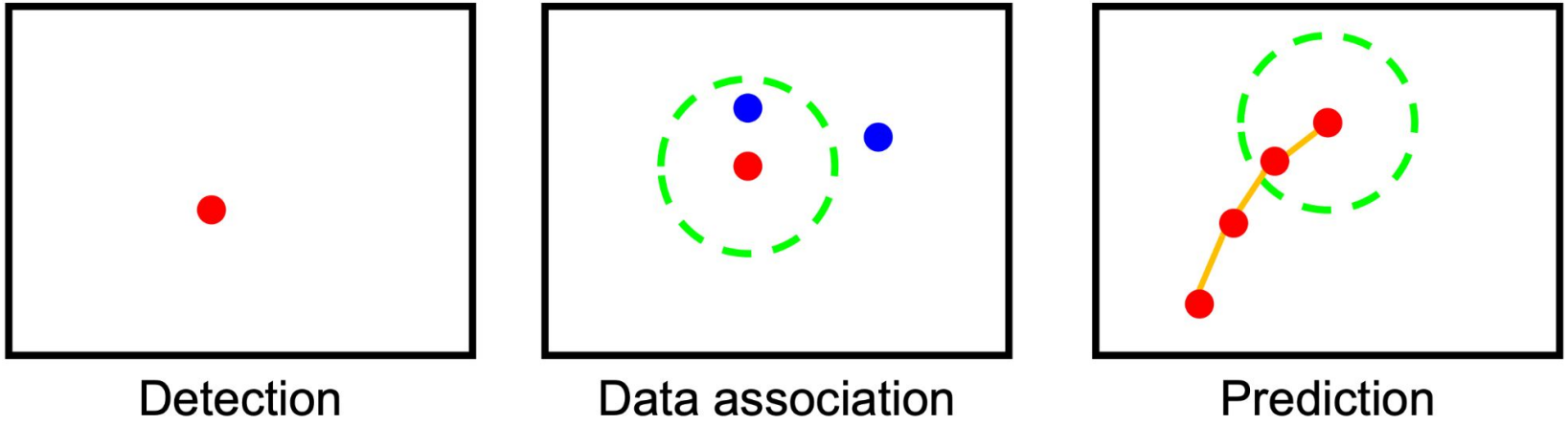
Input Video

Foreground
(Moving Median)

Foreground
(Adaptive GMM)

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

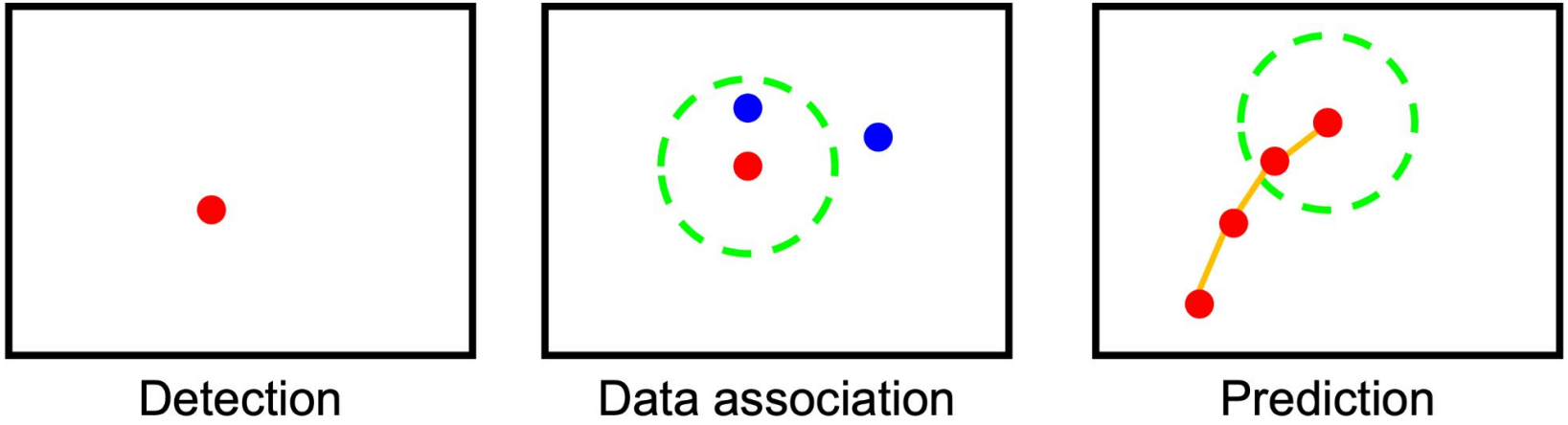
Elements of Tracking



- **Detection** : Where are candidate objects?
  - Find the object(s) of interest in the image.
- **Data Association :** Which detection corresponds to which object?
  - Determine which observations come from the same object.
- **Prediction** : Where will the tracked object be in the next time step?
  - Predict future motion based on the observed motion pattern.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Elements of Tracking



- **Detection** : Where are candidate objects?
  - Find the object(s) of interest in the image.
- **Data Association :** Which detection corresponds to which object?
  - Determine which observations come from the same object.
- **Prediction** : Where will the tracked object be in the next time step?
  - Predict future motion based on the observed motion pattern.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Object Tracking

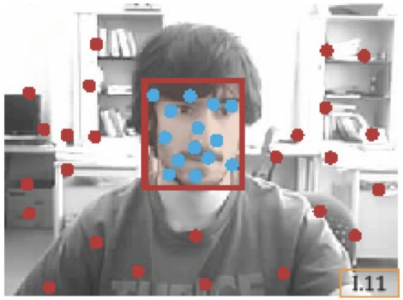
Tracking-by-Detection

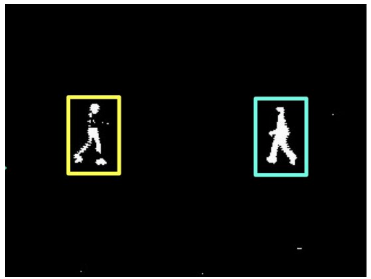
Tracking-by-Detection

Feature-based Detectors

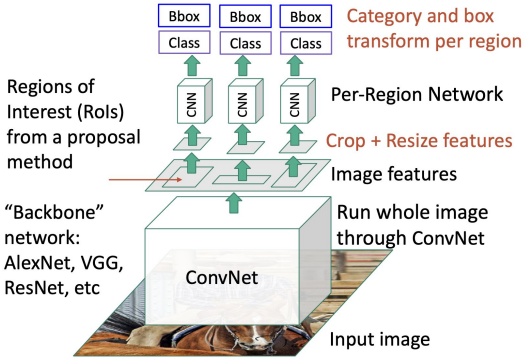
Template Matching

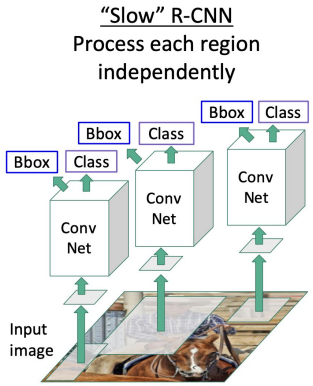
Feature Matching (SIFT, Blobs)

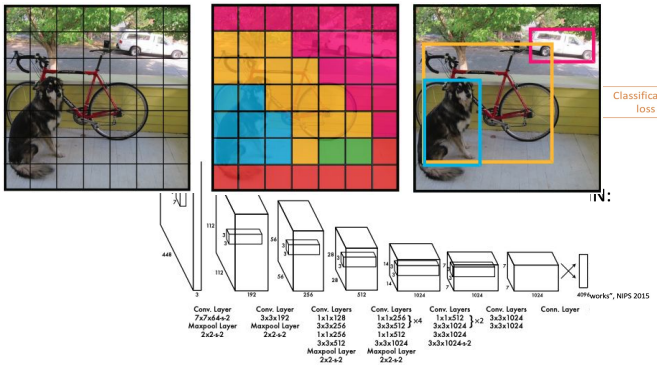


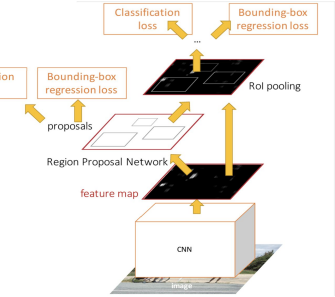


Learning-based Detectors



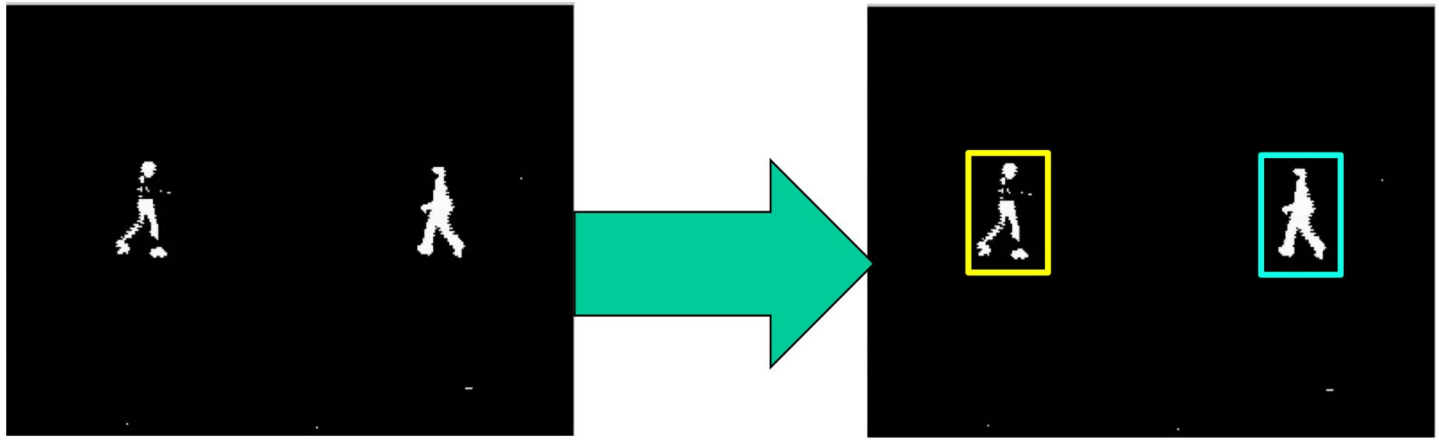






Feature based Detectors

Grouping Pixels into Blobs



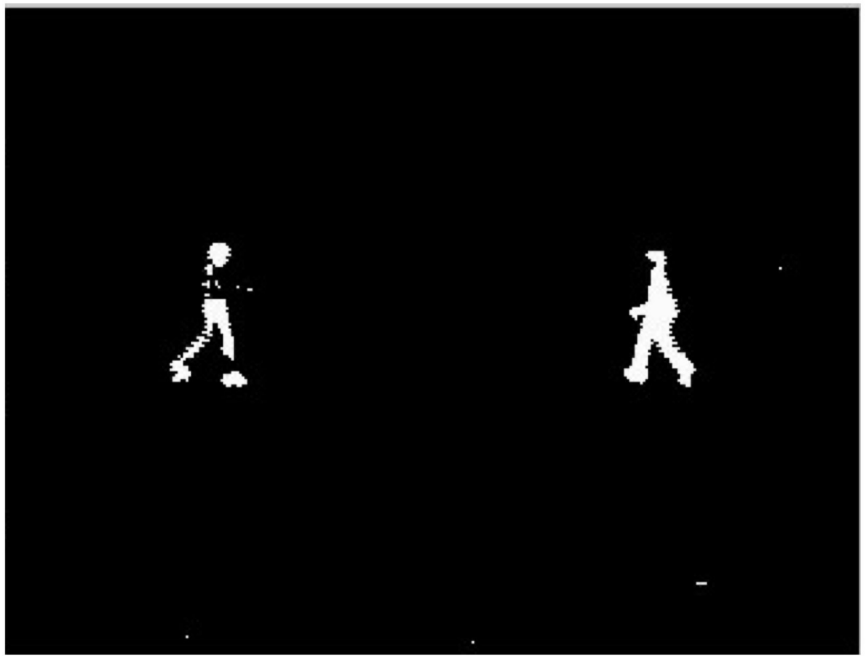
Motivation: Change detection is a pixel-level process.

We want to raise our description to a higher level of abstraction, so we can reason about objects and their motions. It also allows us to count the objects.

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

Grouping Pixels into Blobs



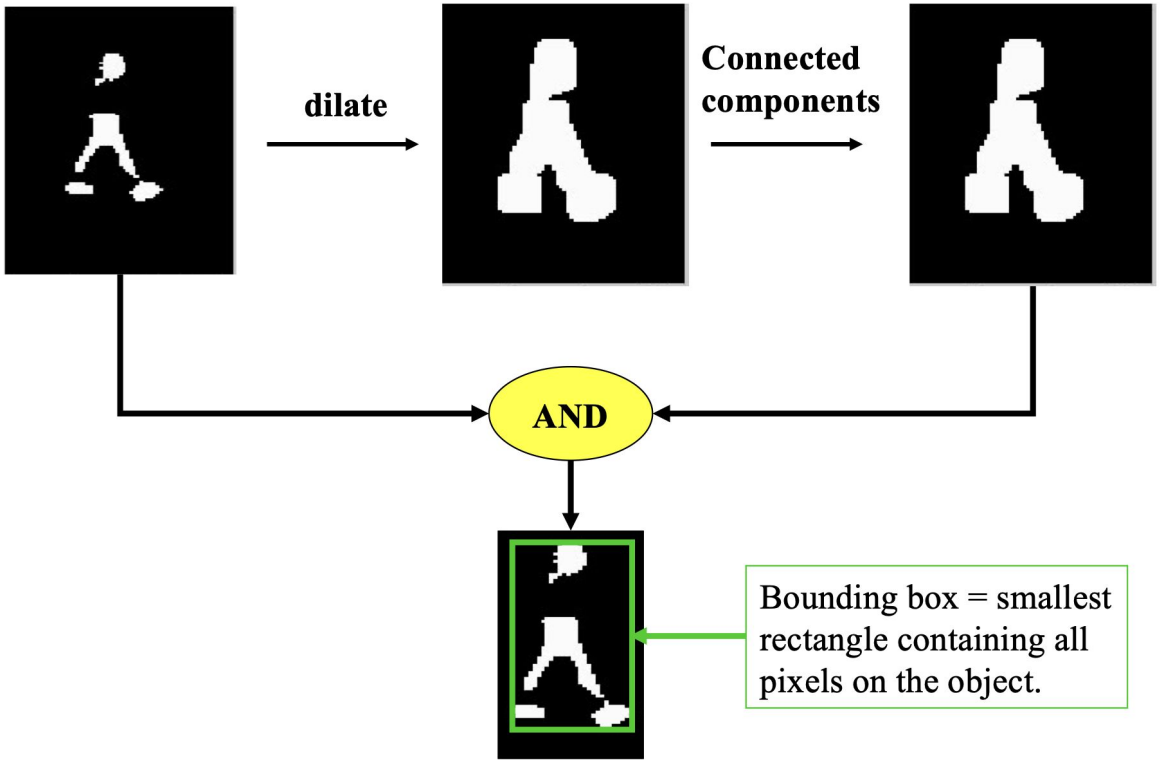
Simple Approach :

- median filter to remove noisy pixels
- connected components (with gap spanning)
- Size filter to remove small regions

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

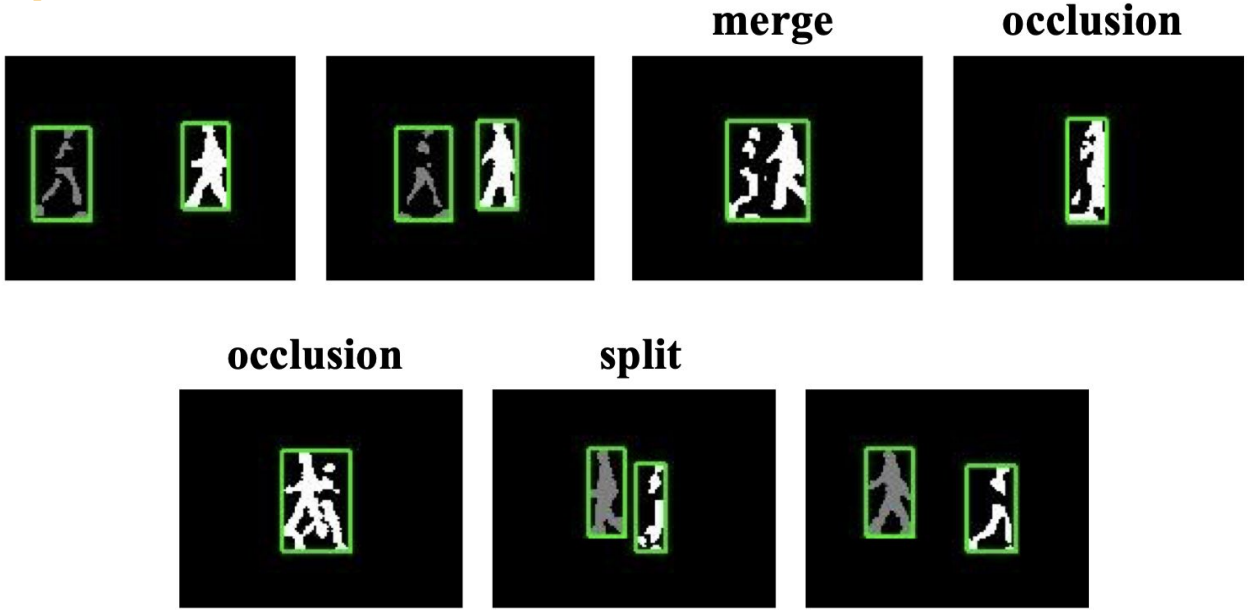
Gap Spanning Connected Components



Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

Blob Merge/Split

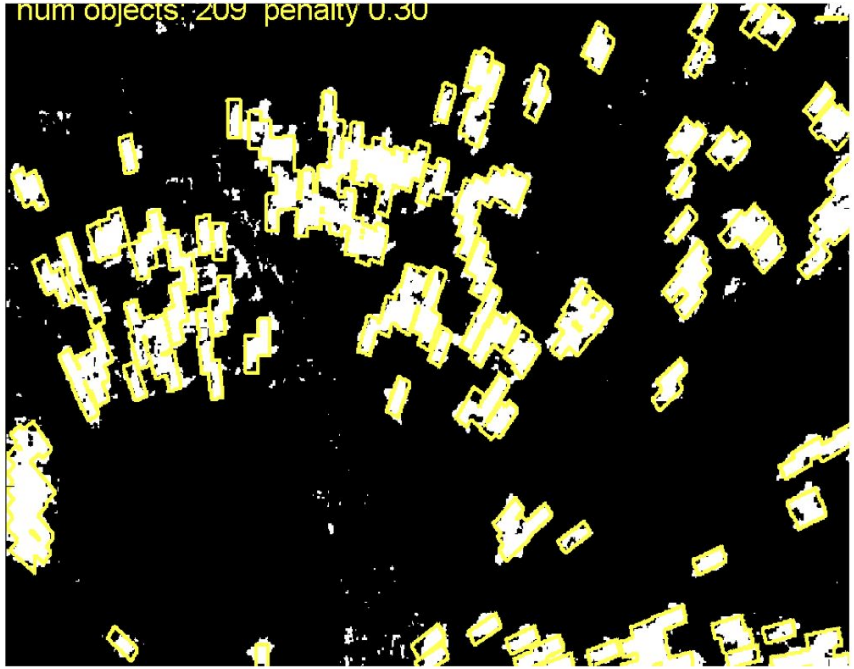


When two objects **pass close** to each other, they are **detected as a single blob**. Often, one object will become occluded by the other one. One of the **challenging problems** is to **maintain correct labeling** of each object after they split again.

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

Counting Crowds



W.Ge, R.Collins and B.Ruback, "Vision-based Analysis of Small Groups in Pedestrian Crowds," IEEE Trans Pattern Analysis and Machine Intelligence (PAMI), Vol 34(5), May 2012, pp.1003-1016.
W.Ge and R. Collins, "Crowd Density Analysis with Marked Point Processes [Applications Corner],"Signal Processing Magazine, Vol 27, Issue 5, Sept 2010, pp. 107-111.

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

Template Matching (Appearance Matching)

Frame I_{t-1}

Object Template

Frame I_t

Given template window S in frame I_{t-1}, search neighborhood to find match in image I_t .

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

Template Matching (Appearance Matching)

Frame I_{t-1}

Object Template

Frame I_t

Given template window S in frame I_{t-1}, search neighborhood to find match in image I_t .

Simple implementation. Not robust to change in scale, viewpoint, occlusion etc.

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

Template Matching: Similarity Metrics

Find pixel $(k,l) \in S$ with Minimum Sum of Absolute Differences:

$$SAD(k,l) = \sum_{(i,j)\in T} |I_1(i,j) - I_2(i+k, j+l)|$$

Find pixel $(k,l) \in S$ with Minimum Sum of Squared Differences:

$$SSD(k,l) = \sum_{(i,j)\in T} |I_1(i,j) - I_2(i+k, j+l)|^2$$

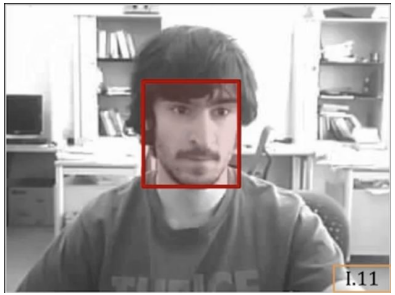
Find pixel $(k,l) \in S$ with Maximum Normalized Cross-Correlation:

$$NCC(k,l) = \frac{\sum_{(i,j)\in T} I_1(i,j)\, I_2(i+k, j+l)}{\sqrt{\sum_{(i,j)\in T} I_1(i,j)^2 \sum_{(i,j)\in T} I_2(i+k, j+l)^2}}$$

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

SIFT

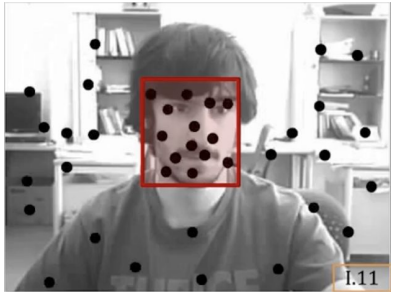


Frame 1 with
bounding box

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

SIFT

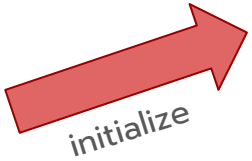


Frame 1 with bounding box and SIFT features

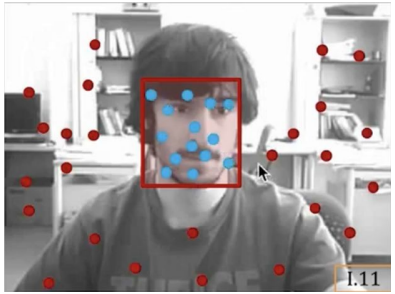
Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

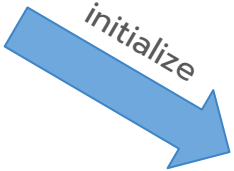
SIFT



Background Model



Frame 1 with bounding box and SIFT features

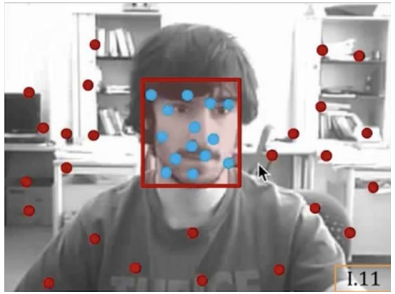


Object Model

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

SIFT



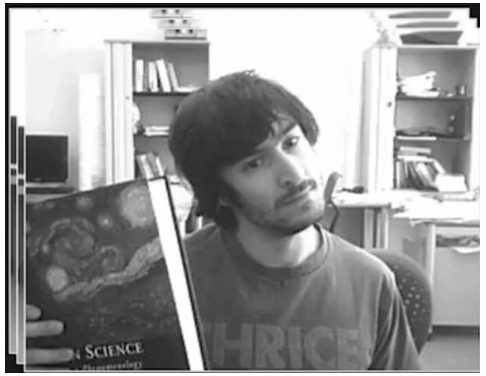
Frame 1 with bounding box and SIFT features

initialize

Background Model

initialize

Object Model

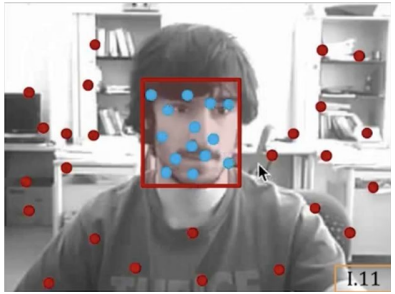


Subsequent frame:

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

SIFT



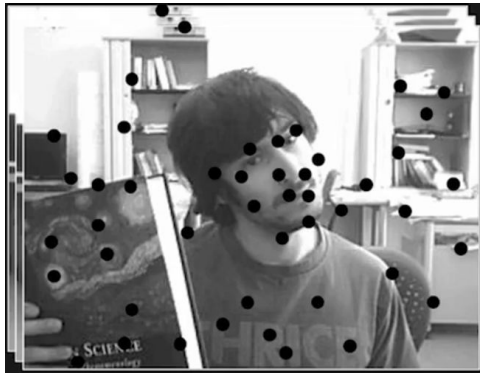
Frame 1 with bounding box and SIFT features

initialize

Background Model

initialize

Object Model

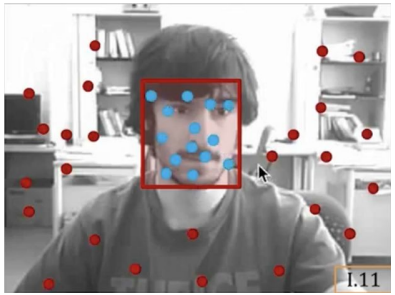


Subsequent frame:

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

SIFT

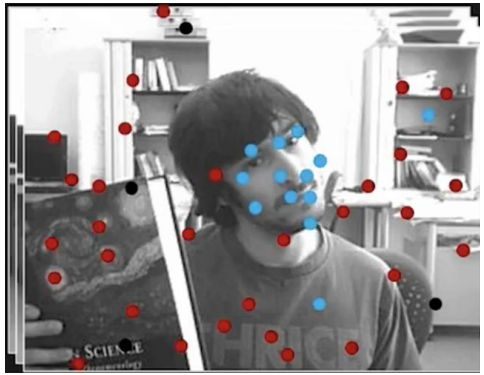


Frame 1 with bounding box and SIFT features

initialize

Background Model

match



Subsequent frame: find window with most matches

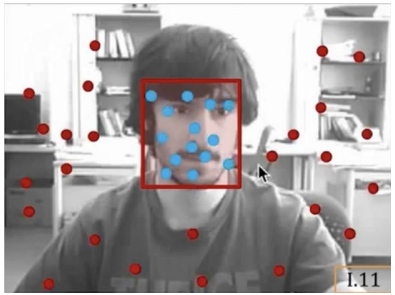
initialize

Object Model

match

Feature based Detectors

SIFT

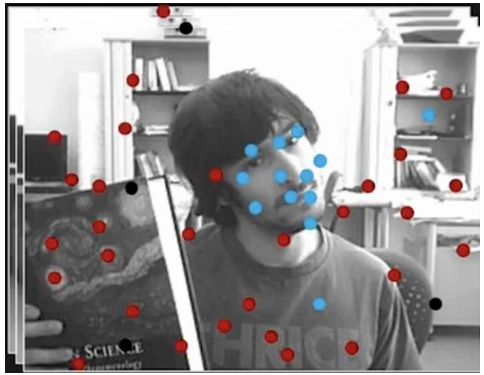


Frame 1 with bounding box and SIFT features

initialize

Background Model

match



Subsequent frame: find window with most matches

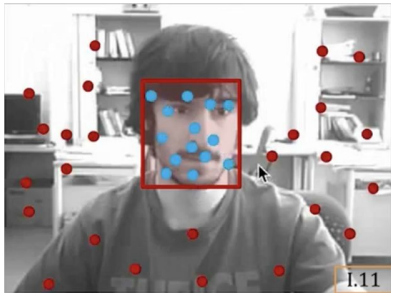
match

Object Model

initialize

Feature based Detectors

SIFT

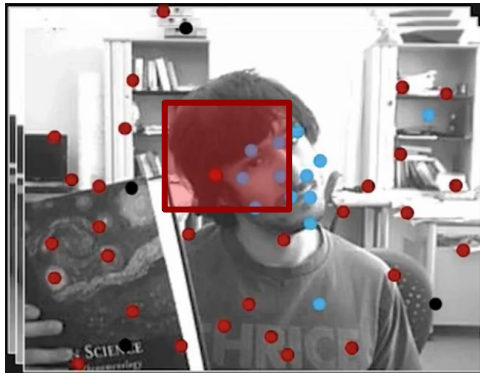


Frame 1 with bounding box and SIFT features

initialize

Background Model

match



Subsequent frame: find window with most matches

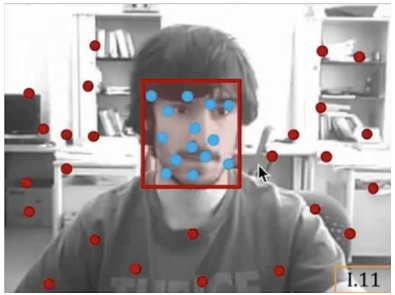
initialize

Object Model

match

Feature based Detectors

SIFT

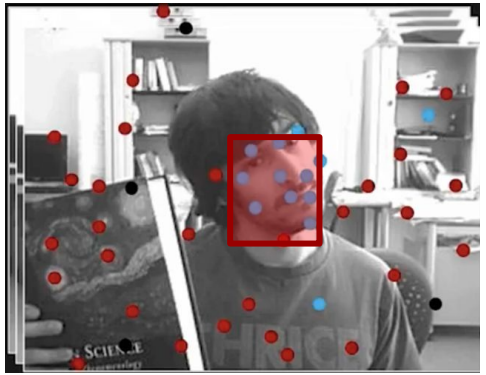


Frame 1 with bounding box and SIFT features

initialize

Background Model

match



Subsequent frame: find window with most matches

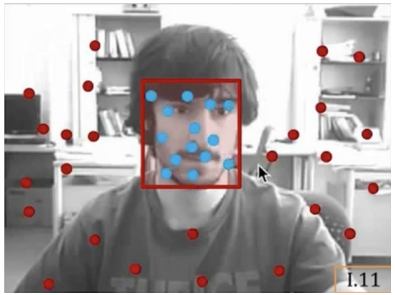
initialize

Object Model

match

Feature based Detectors

SIFT

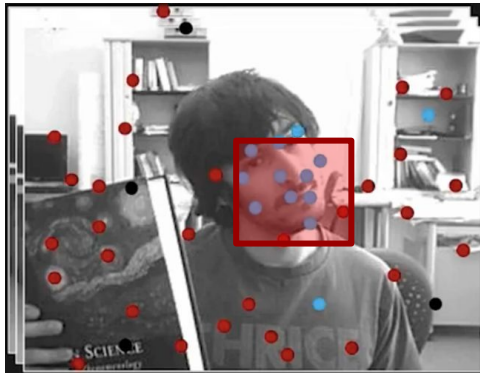


Frame 1 with bounding box and SIFT features

initialize

Background Model

match



Subsequent frame: find window with most matches

initialize

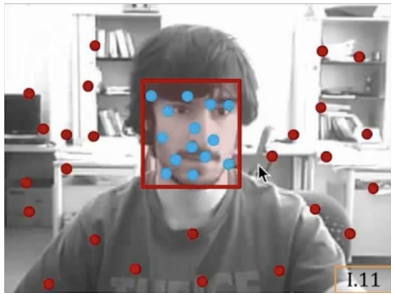
Object Model

match

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

SIFT



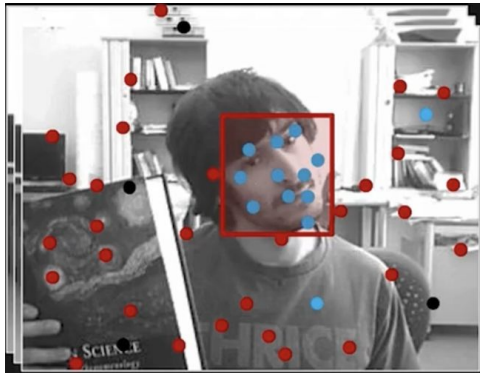
Frame 1 with bounding box and SIFT features

initialize

Background Model

Update

match



Subsequent frame: find window with most matches

match

initialize

Object Model

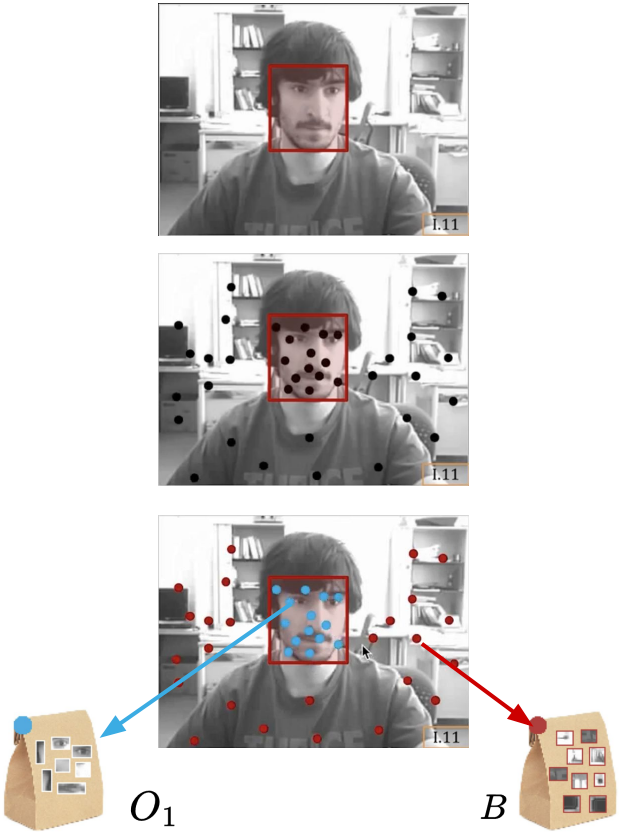
Update

Feature based Detectors

SIFT : Tracking Initialization

At frame 1:

1. User selects a bounding box as object/target.
2. Compute SIFT (or similar) features for the frame.
3. Classify features within the box as object and assign them to set O_1.
4. Classify remaining features as background and assign them to set B.



Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

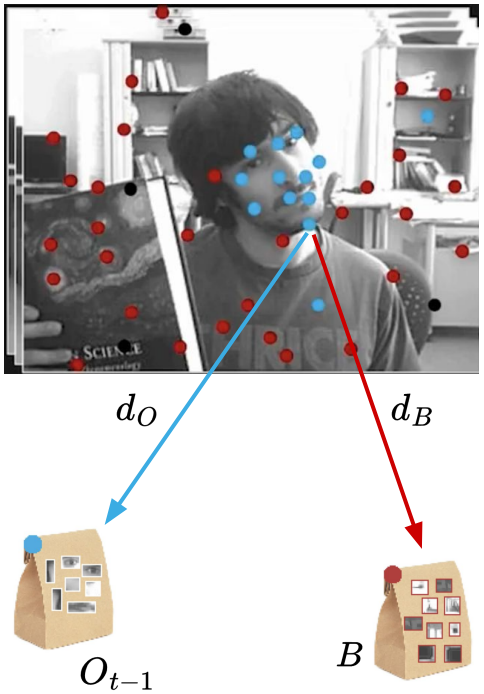
SIFT : Tracking

At frame t:

1. Compute SIFT features and SIFT descriptors $\{v_1, \ldots, v_k\}$ for frame v_i .
2. For each feature and corresponding descriptors :
   a. Compute distance d_O between v_i and the best match in object set . O_{t-1}
   b. Compute distance d_B between v_i and the best match in background set . B
   c.

$$C(v_i) = \begin{cases} +1 \text{ if } \dfrac{d_O}{d_B} < 0.5 \ (v_i \text{ may belong to object}) \\ -1 \text{ otherwise } (v_i \text{ does not belong to object}) \end{cases}$$

Frame I_t



Feature based Detectors

SIFT : Tracking

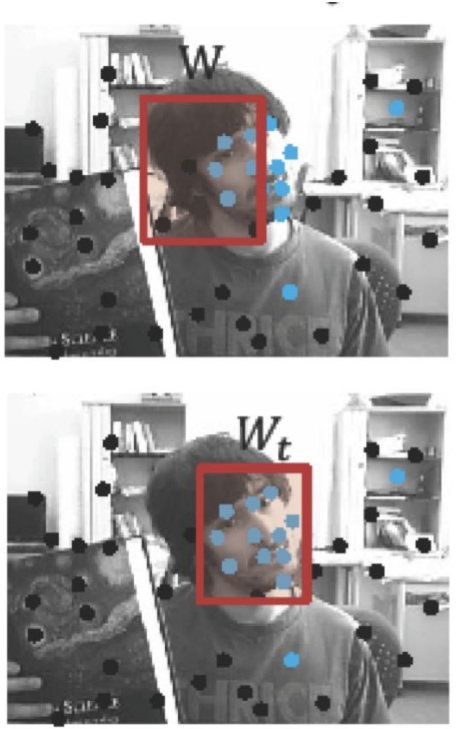
3. For each **search window** W :
   A. Compute $\varphi(W) = \sum C(v_i)$ for all features v_i in side W
   B. Compute a heuristic $\tau(W, W_{t-1})$ that penalizes large deviations from previous location, size and shape W_{t-1}
   C. Compute **match score**

$$\mu(W) = \varphi(W) - \tau(W, W_{t-1})$$

4. Select window W_t with the best match score as new object location.
5. Update object appearance model:

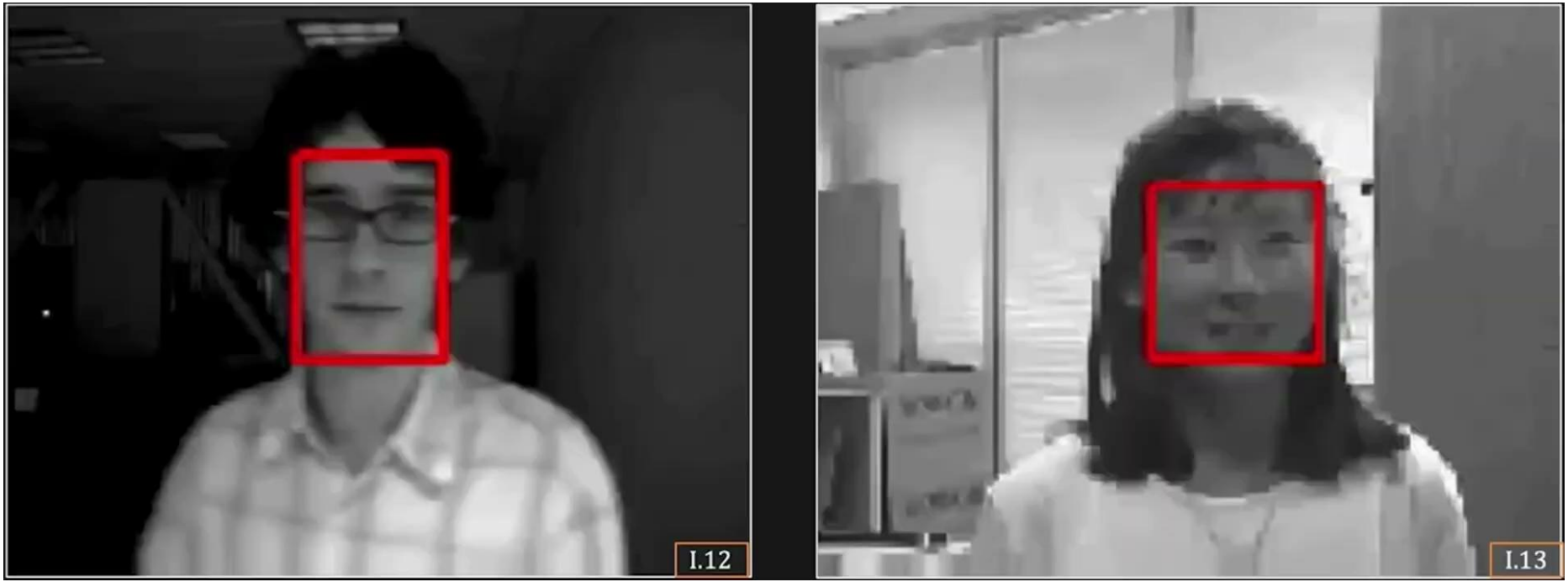
$$O_t = O_{t-1} \cup \{v_i\} \forall v_i \text{ inside } W_t \text{ such that } C(v_i) = +1.$$

Frame I_t



Feature based Detectors

Results : Scale and Orientation



Resilient to changes in scale and orientation

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Feature based Detectors

Results : Occlusion

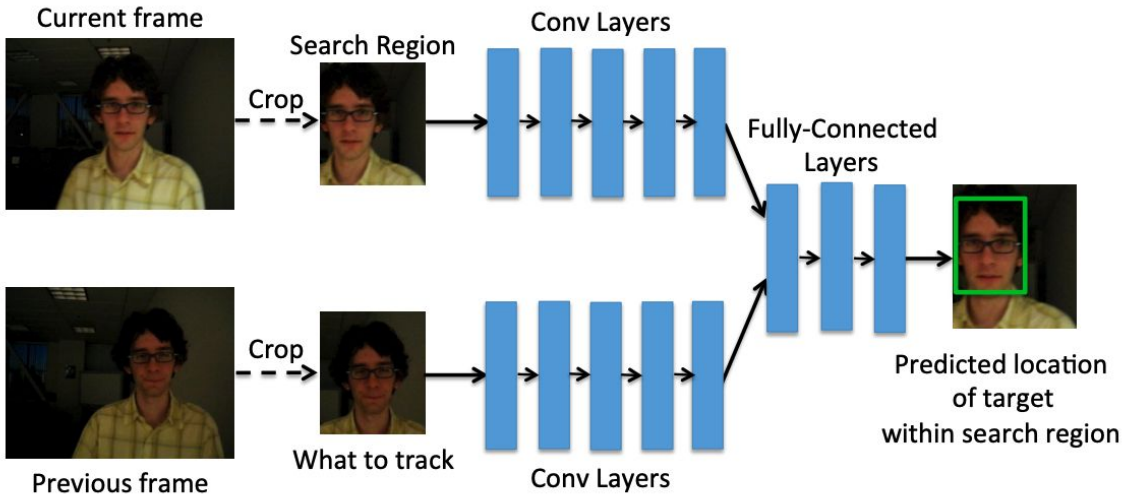
Resilient to occlusion

Adopted from: Shree K. Nayar, Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, Mar. 2022

Learning based Detectors

GOTURN: Generic Object Tracking Using Regression Networks

D. Held, S. Thrun, S. Savarese. “Learning to Track at 100 FPS with Deep Regression Networks”. ECCV 2016.

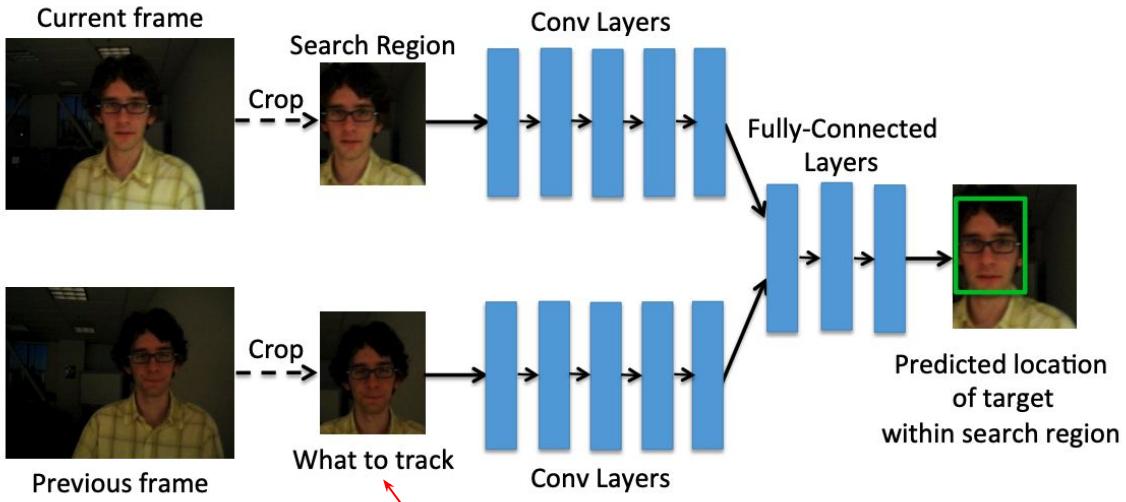


Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., “Computer Vision III: Detection, Segmentation and Tracking (CV3DST)”,the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Learning based Detectors

GOTURN: Generic Object Tracking Using Regression Networks

Input: What to track?



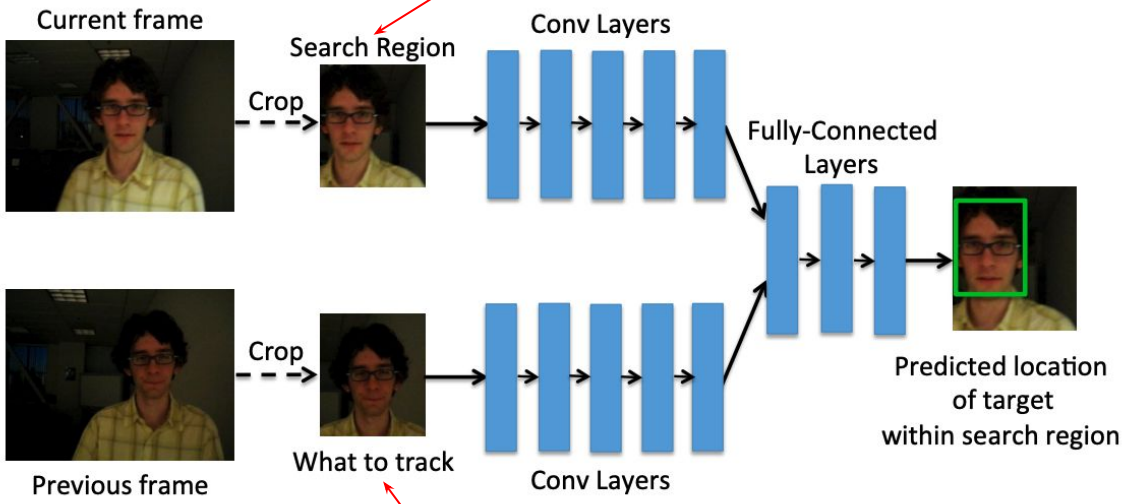
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Learning based Detectors

GOTURN: Generic Object Tracking Using Regression Networks

Where to look

Assume smooth "slow" motion. The object cannot be far away from where it was in the previous frame.



Crop the object to be tracked
Initialization of our tracker

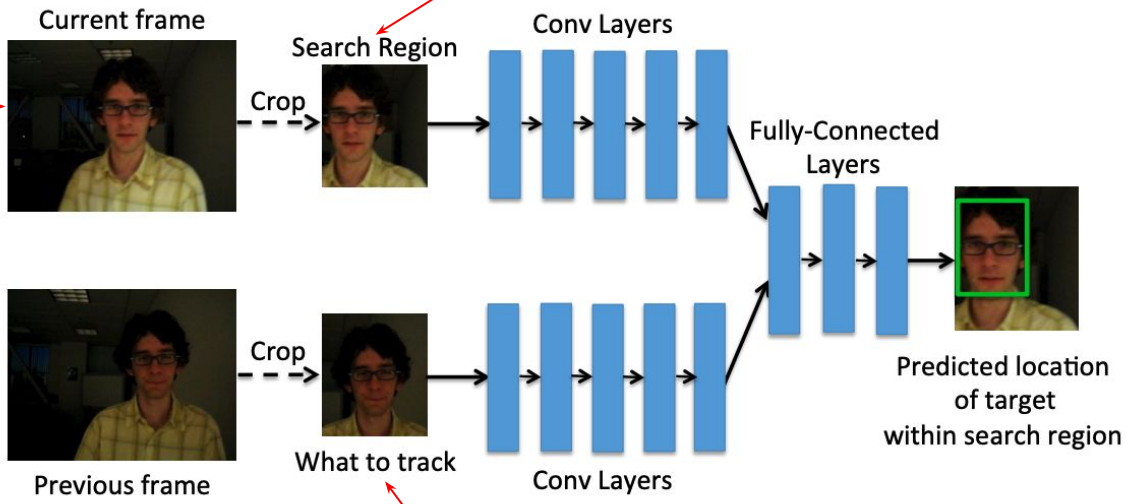
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Learning based Detectors

GOTURN: Generic Object Tracking Using Regression Networks

Where to look

Assume smooth "slow" motion. The object cannot be far away from where it was in the previous frame.



Crop the object to be tracked
Initialization of our tracker

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Learning based Detectors

GOTURN: Generic Object Tracking Using Regression Networks

PROS of GOTURN:

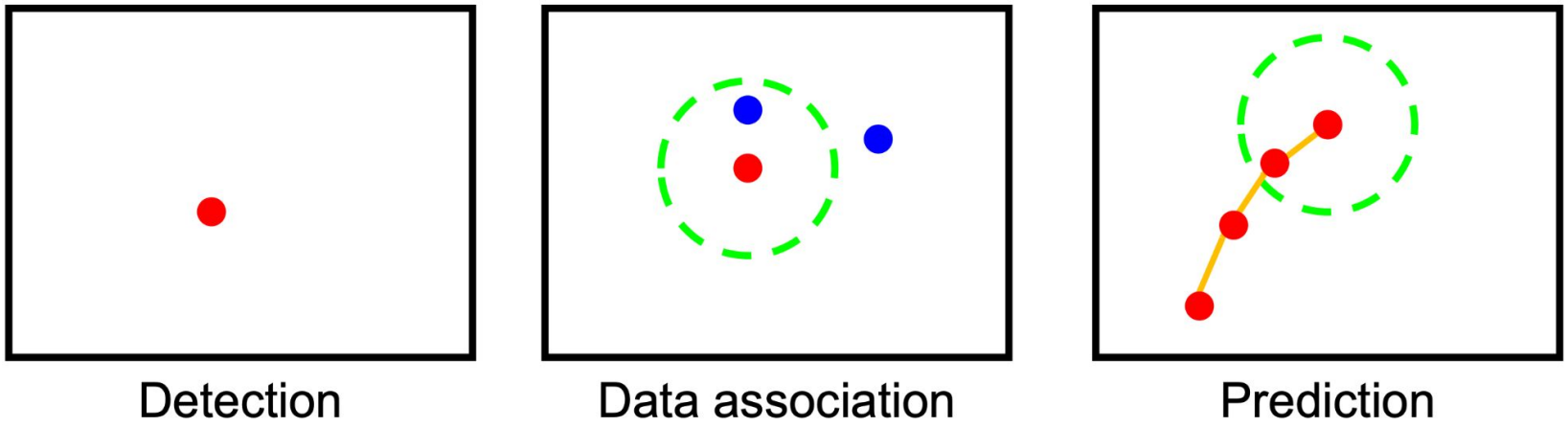
- No online training required.
- Tracking is done by comparison, so we do not need to retrain or finetune our model for every new object.
- Close to the template matching approach
- This makes it very fast!

CONS:

- We have a motion assumption. If the object moves fast and goes out of our search window, we cannot recover.

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Elements of Tracking



- **Detection** : Where are candidate objects?
  - Find the object(s) of interest in the image.
- **Data Association :** Which detection corresponds to which object?
  - Determine which observations come from the same object.
- **Prediction** : Where will the tracked object be in the next time step?
  - Predict future motion based on the observed motion pattern.

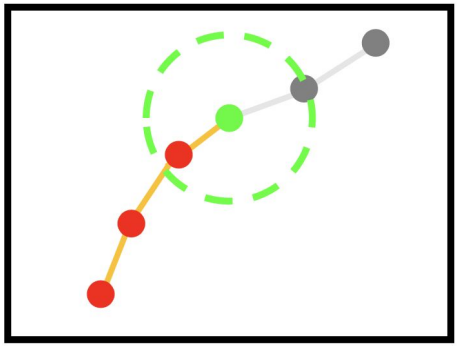
Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Object Tracking

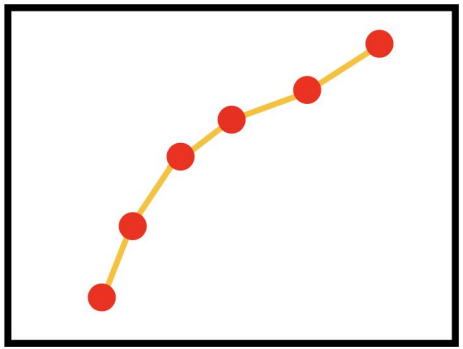
Tracking with Dynamics

Tracking with Dynamics

Online vs. Offline Tracking



Online Tracking



Offline Tracking

Estimate current state given current and past observations

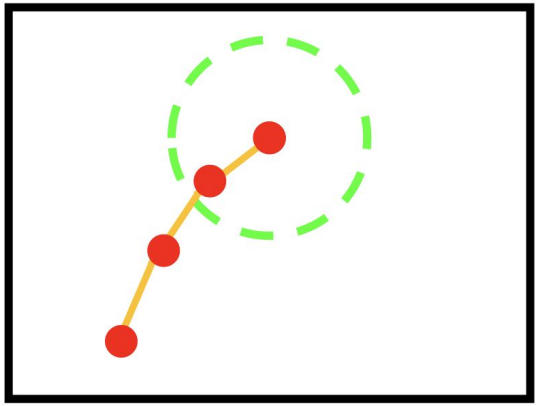
- Processes two frames at a time
- For real-time applications
- Prone to drifting à hard to recover from errors or occlusions

Estimate all states given all observations (batch mode)

- Processes a batch of frames
- Good to recover from occlusions
- Not suitable for real-time applications
- Suitable for video analysis

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Tracking with Dynamics



Key idea

- **Given** a model of expected motion, **predict** where objects will occur in next frame, even before seeing the image.

Goals

- **Restrict search** for the object
- Improved estimates since measurement noise is reduced by trajectory smoothness.
- Robustness to missing or weak observations

Assumption: continuous motion patterns

- Camera is not moving instantly to new viewpoint.
- Objects do not disappear and reappear in different places.
- Gradual change in pose between camera and scene.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Tracking with Dynamics

General Model for Tracking

Key idea

- **Given** a model of expected motion, **predict** where objects will occur in next frame, even before seeing the image.

Goals

- **Restrict search** for the object
- Improved estimates since measurement noise is reduced by trajectory smoothness.

Assumption: continuous motion patterns

- Camera is not moving instantly to new viewpoint.
- Objects do not disappear and reappear in different places.
- Gradual change in pose between camera and scene.

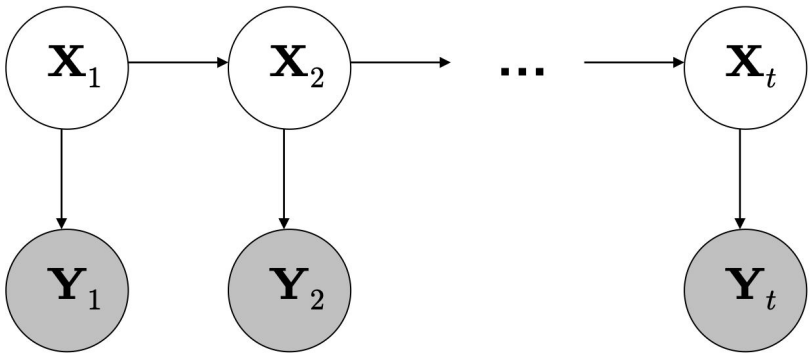
Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Tracking with Dynamics

General Model for Tracking

Representation

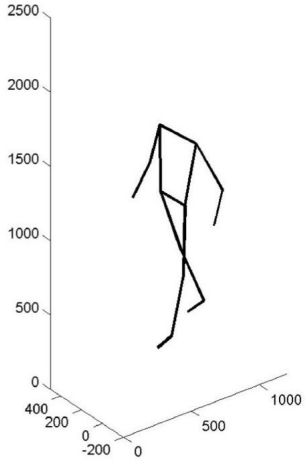
- The moving object of interest is characterized by an underlying state X
- State X gives rise to measurements or observations Y
- At each time t, the state changes to X_t and we get a new observation Y_t
- Given a sequence of observations $Y = \{y_t\}$, we aim to recover $X = \{x_t\}$

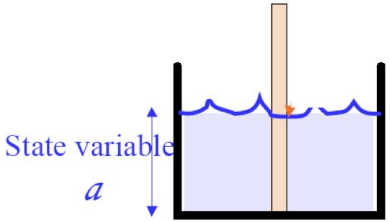


Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Tracking with Dynamics

State vs. Observation





Hidden state : parameters of interest (e.g. position, pose, viewpoint, velocity, acceleration)

Measurement: what we get to directly observe (=Observation)

Due to sensor noise/observability: hidden state != observation

In tracking, we often aim for a **probabilistic estimate**, not a point estimate

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

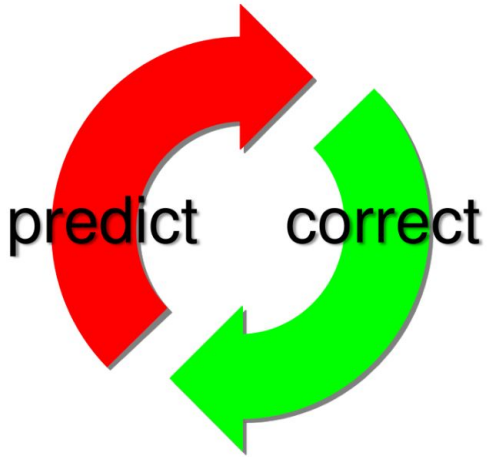
Tracking with Dynamics

Bayes Filter

We obtain a **recursive state estimation algorithm:**

1. **Prediction**

$$\underbrace{p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}_{\text{Prediction}} = \int \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1})}_{\text{Motion Model}} \underbrace{p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}}_{\text{Prev. Posterior}}$$



2. **Correction**

$$\underbrace{p(\mathbf{x}_t|\mathbf{y}_{1:t})}_{\text{Posterior}} \propto \underbrace{p(\mathbf{y}_t|\mathbf{x}_t)}_{\text{Observation}} \underbrace{p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}_{\text{Prediction}}$$

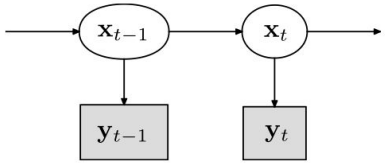
The Bayes filter runs **online** (i.e., update per frame).

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Tracking with Dynamics

Kalman Filter

$$p(\mathbf{X}, \mathbf{Y}) = p(\mathbf{x}_1) \prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t)$$

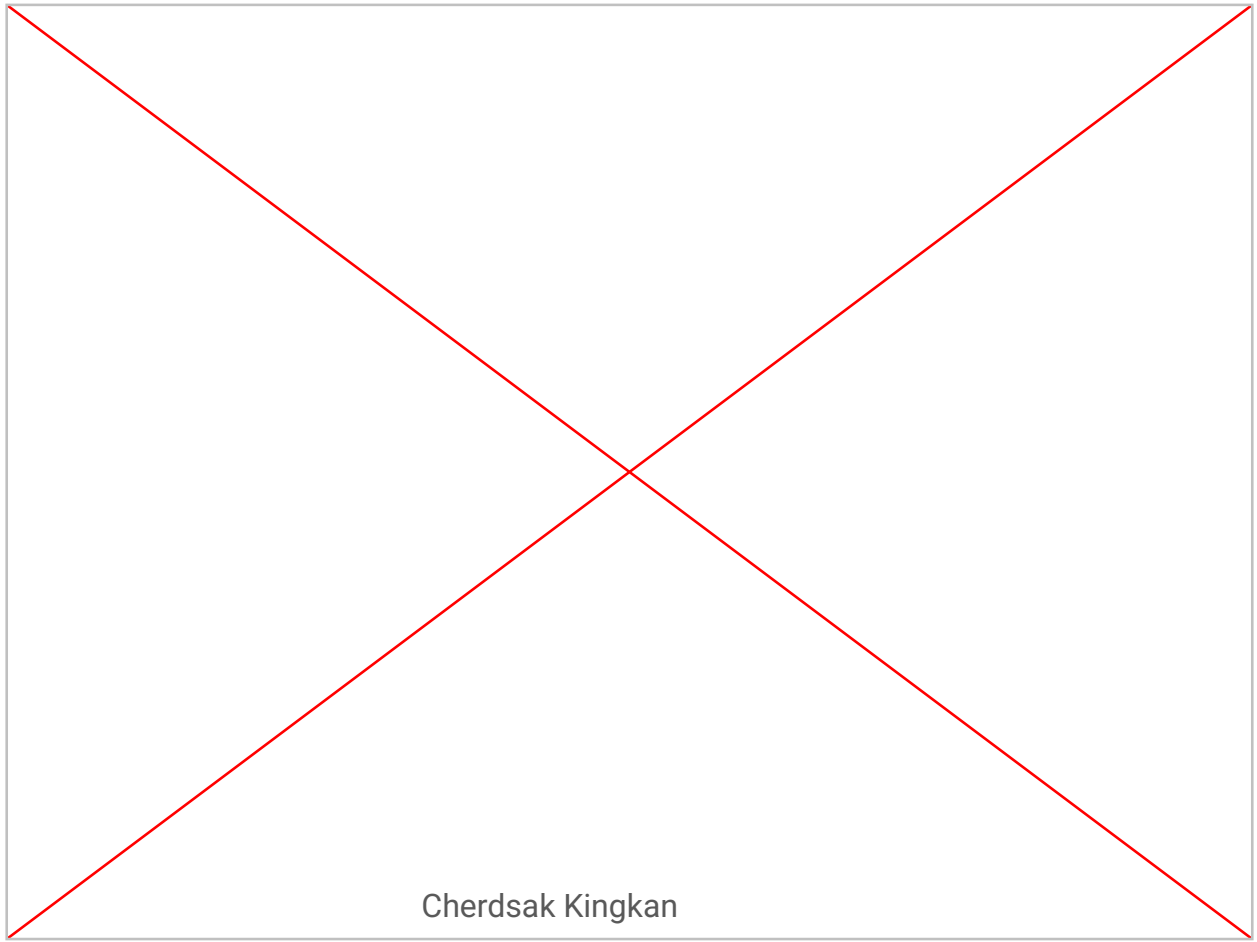


- If $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{y}_t|\mathbf{x}_t)$ are linear and normally distributed, we obtain the **Kalman filter** (KF) $\Rightarrow$ Parameters: mean $\boldsymbol{\mu}$ / covariance $\boldsymbol{\Sigma}$
- The solution for the Kalman filter is given in closed form as products/marginals are **Gaussian** $\Rightarrow$ max. likelihood = least squares (omitting formulas here)
- The Kalman filter is the optimal linear filter in least squares sense
- Non-linear case: **Extended Kalman Filter** (EKF), **Unscented Kalman Filter** (UKF)
- Multiple modes can be handled using a **Particle Filter** which computes a sampling-based approximation to the state posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$
- More recently, learning-based motion models (e.g., RNN) have been explored

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2, RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

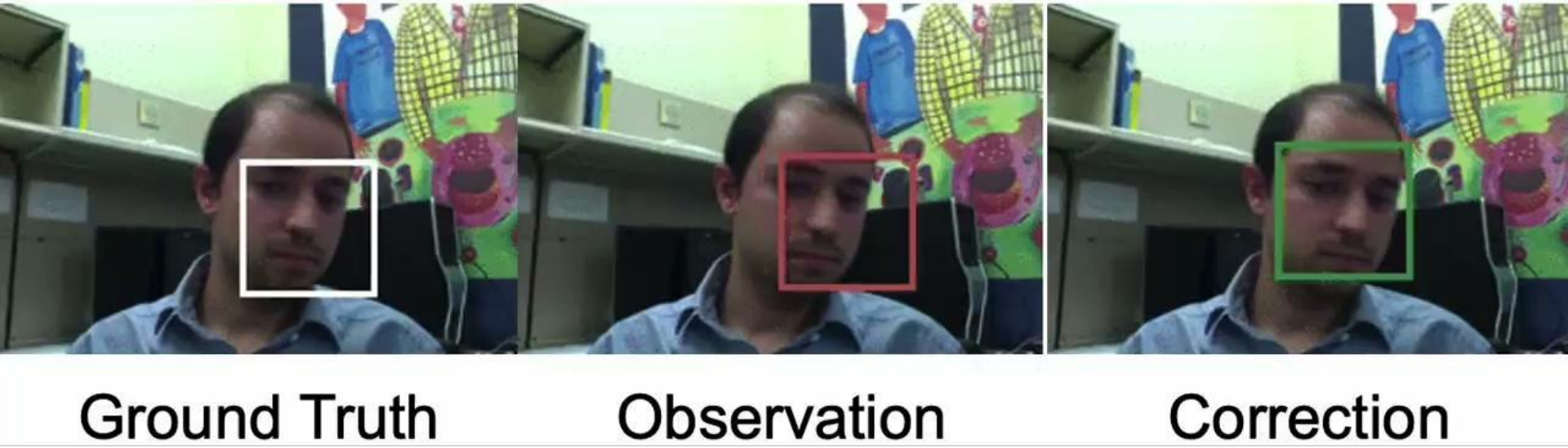
Tracking with Dynamics

Kalman Filter

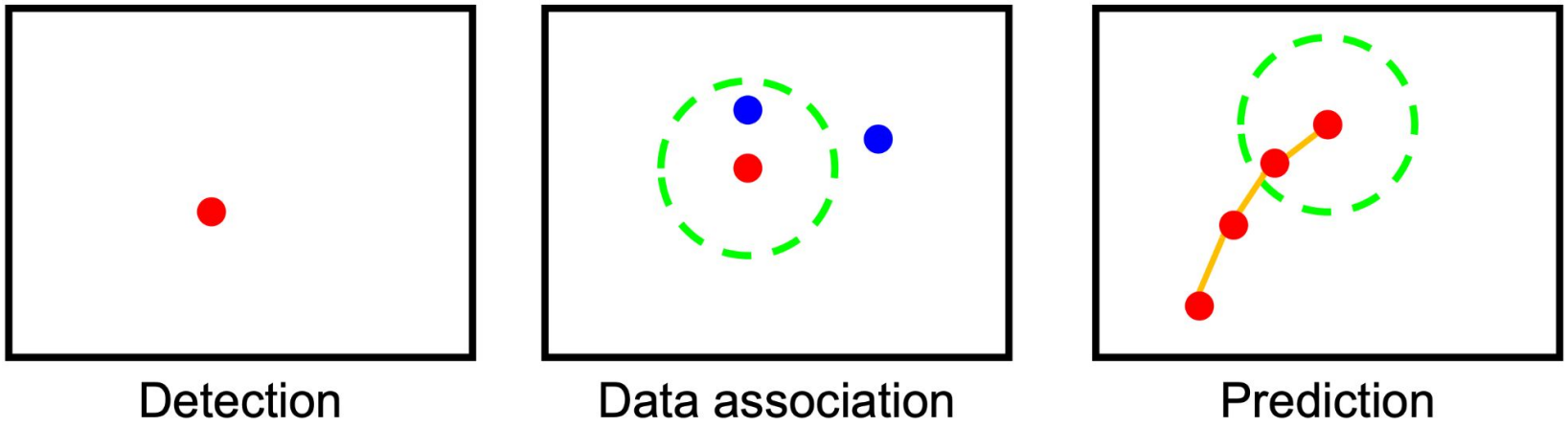


Tracking with Dynamics

Kalman Filter: Comparison



Elements of Tracking



- **Detection** : Where are candidate objects?
  - Find the object(s) of interest in the image.
- **Data Association :** Which detection corresponds to which object?
  - Determine which observations come from the same object.
- **Prediction** : Where will the tracked object be in the next time step?
  - Predict future motion based on the observed motion pattern.

Slide adopted from: Prof. Dr. Bastian Leibe, Computer Vision 2,
RWTH Aachen University, https://www.vision.rwth-aachen.de/course/25/

Object Tracking

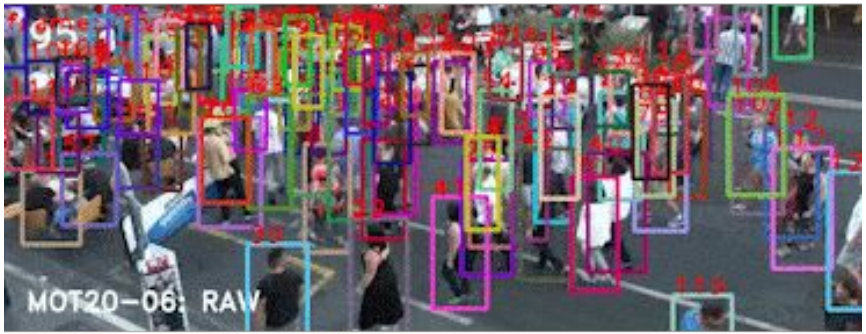
Multiple Object Tracking

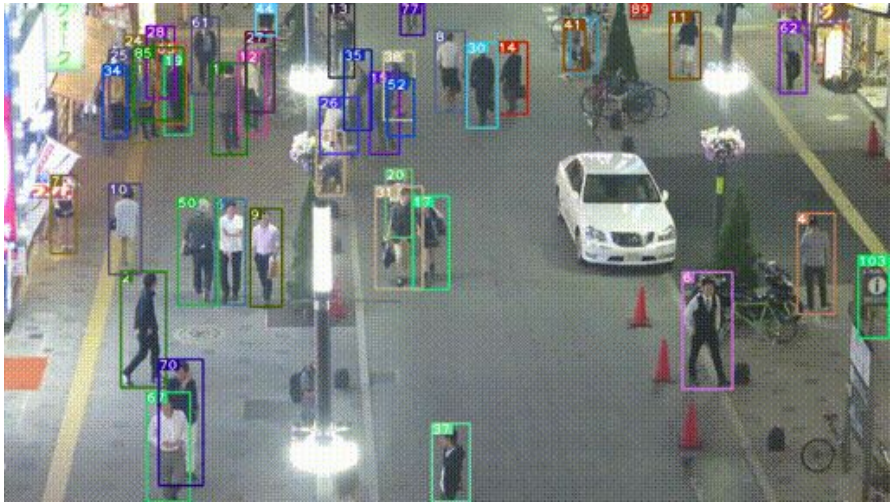
Multiple Object Tracking

Challenges

- Multiple objects of the same type
- Heavy occlusions
- Appearance is often very similar

Q: How can we associate detections in a new frame to existing object tracks?





Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

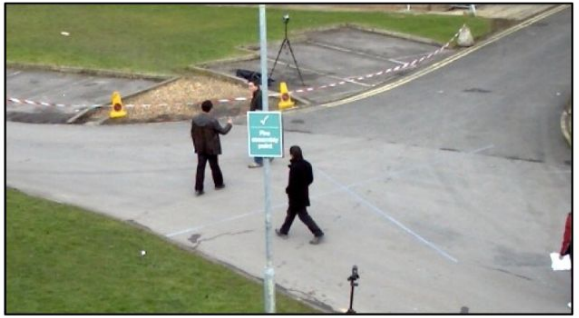
Multiple Object Tracking

Online Tracking

t

t + 1

t + 2

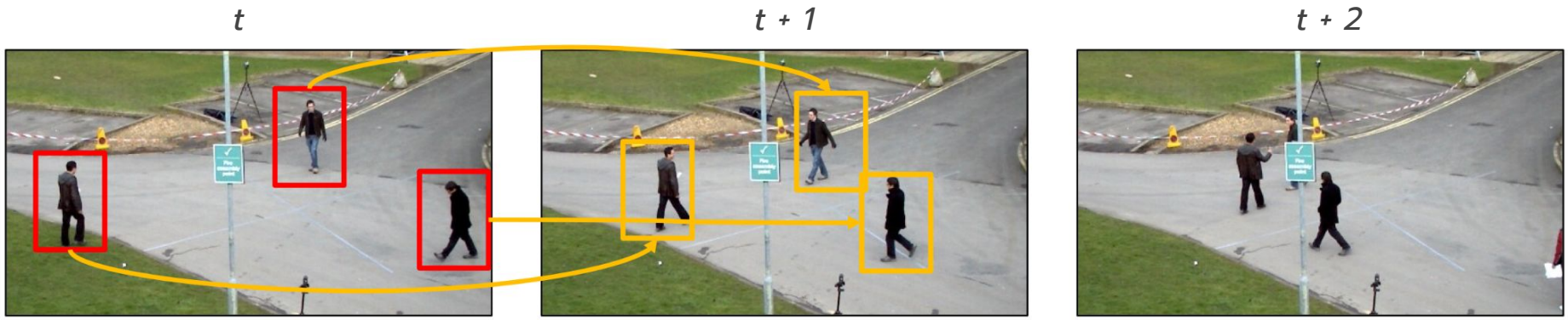


1. Track initialization (e.g. using a detector)

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Online Tracking



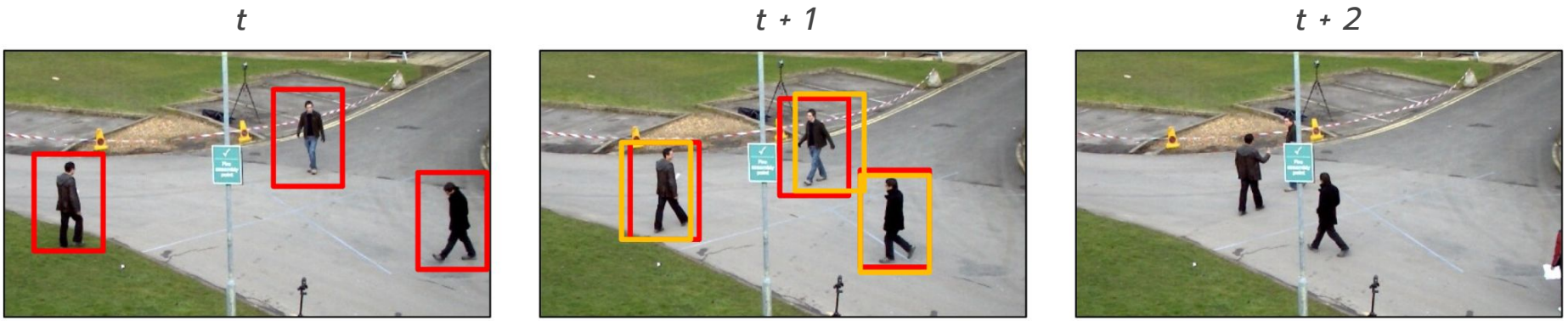
1. Track initialization (e.g. using a detector)

2. Prediction of the next position (motion model)
    - Classic: Kalman filter
    - Nowadays: Recurrent architecture
    - For now: we will assume a constant velocity model

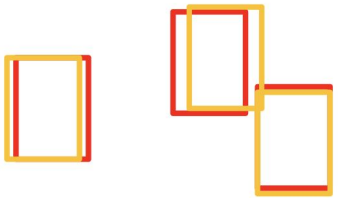
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Online Tracking



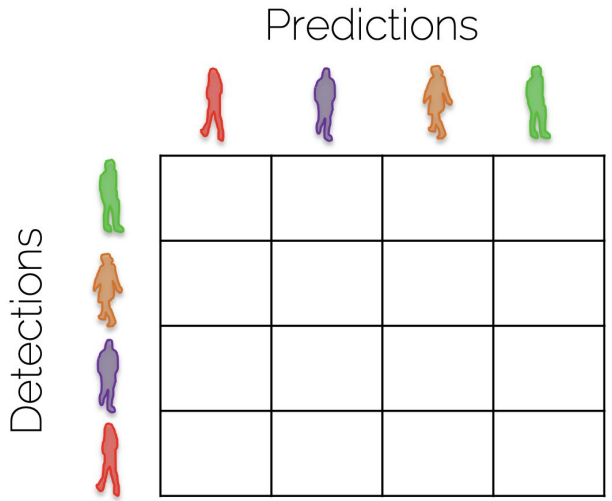
1. Track initialization (e.g. using a detector)
2. Prediction of the next position (motion model)
3. Matching predictions with detections (appearance model)



Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)



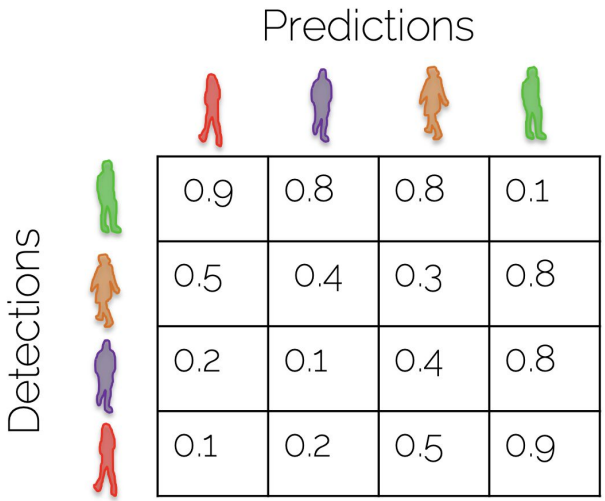
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., “Computer Vision III: Detection, Segmentation and Tracking (CV3DST)”,the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)

Bipartite matching

- Define distances between boxes (e.g., IoU, pixel distance, 3D distance)



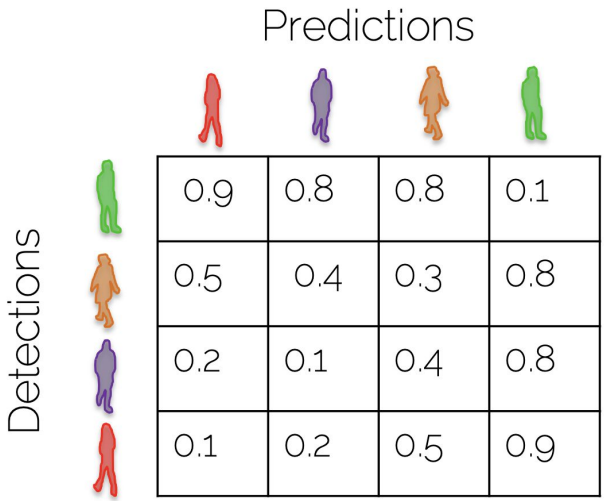
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)

Bipartite matching

- Define distances between boxes (e.g., IoU, pixel distance, 3D distance)
- Solve the unique matching with e.g., the Hungarian algorithm*



Demo: http://www.hungarianalgorithm.com/solve.php

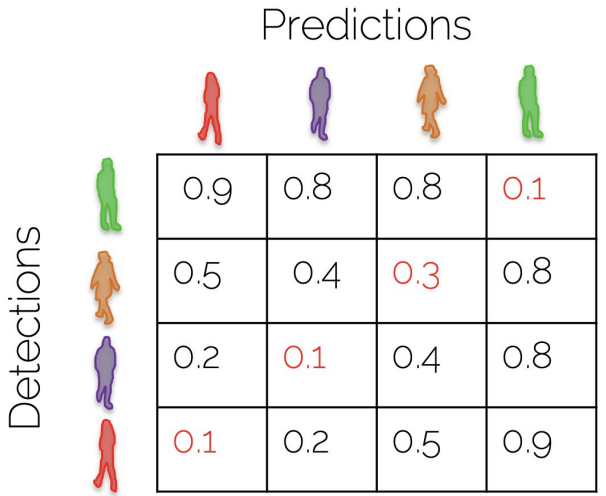
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)

Bipartite matching

- Define distances between boxes (e.g., IoU, pixel distance, 3D distance)
- Solve the unique matching with e.g., the Hungarian algorithm*
- Solutions are the unique assignments that minimize the total cost



Demo: http://www.hungarianalgorithm.com/solve.php

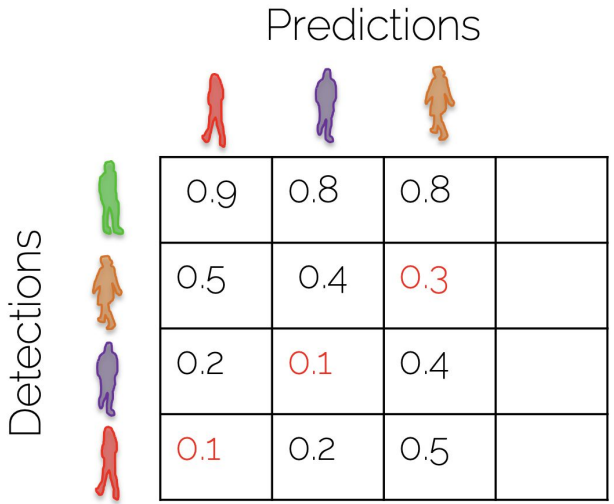
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)

Bipartite matching

- What happens if we are missing a prediction?



Demo: http://www.hungarianalgorithm.com/solve.php

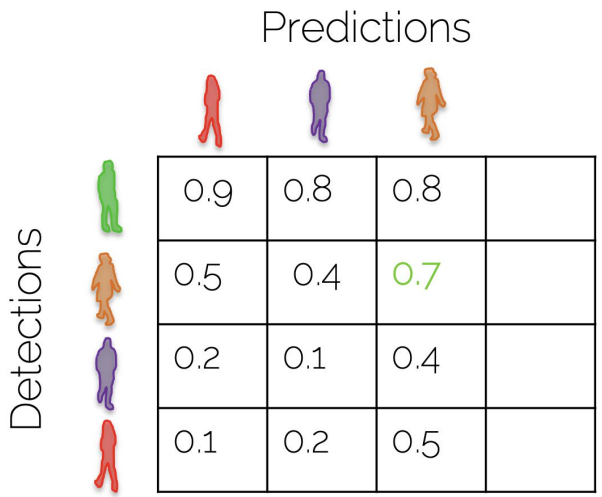
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)

Bipartite matching

- What happens if we are missing a prediction?
- What happens if no prediction is suitable for the match?



no sutiable matchig

see neext slide :

Demo: http://www.hungarianalgorithm.com/solve.php

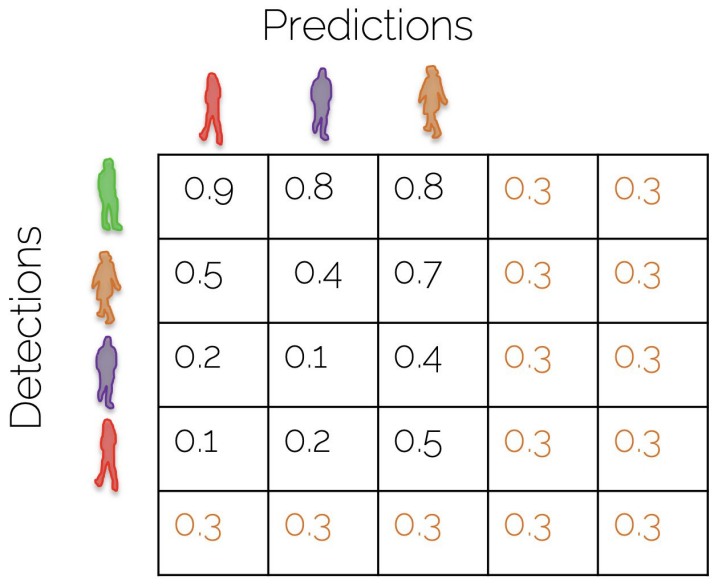
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)

Bipartite matching

- What happens if we are missing a prediction?
- What happens if no prediction is suitable for the match?
- Introducing extra nodes with a threshold cost



Demo: http://www.hungarianalgorithm.com/solve.php

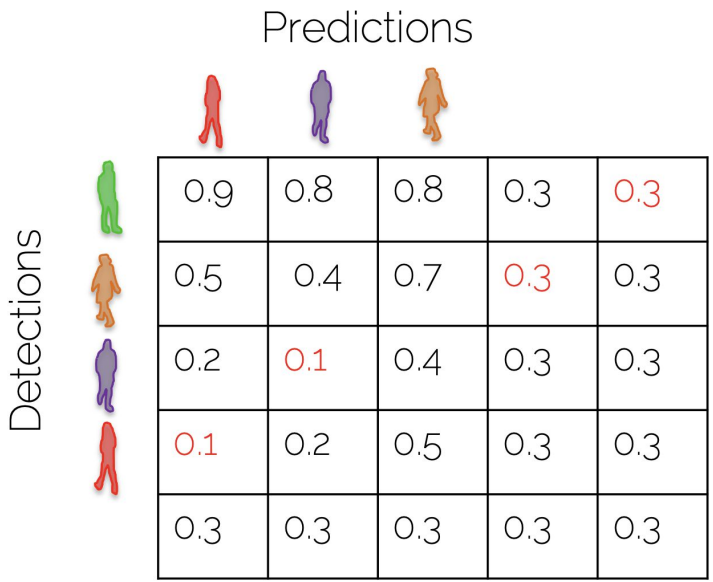
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Matching predictions with detections (appearance model)

Bipartite matching

- What happens if we are missing a prediction?
- What happens if no prediction is suitable for the match?
- Introducing extra nodes with a threshold cost
- Apply Hungarian
- Result: two detections have no matched prediction

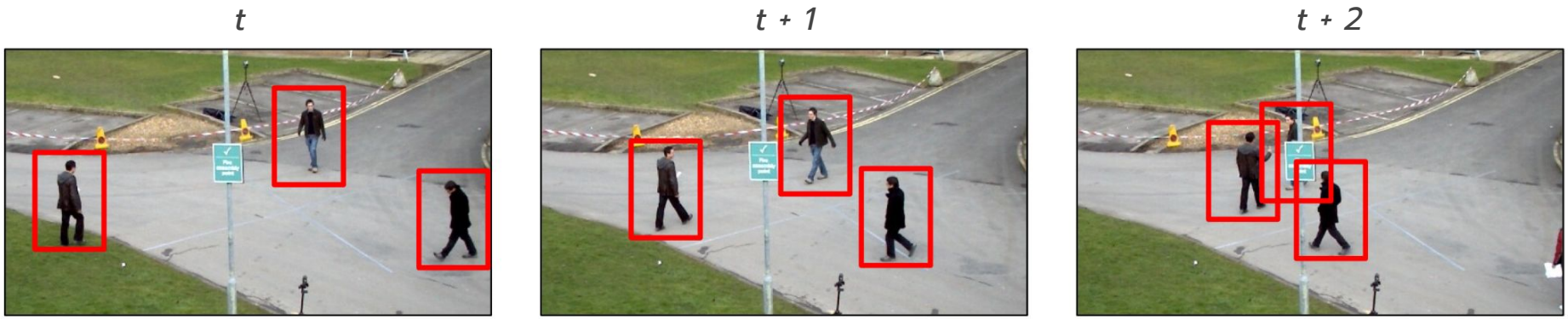


Demo: http://www.hungarianalgorithm.com/solve.php

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Offline Tracking : Frame-by-Frame

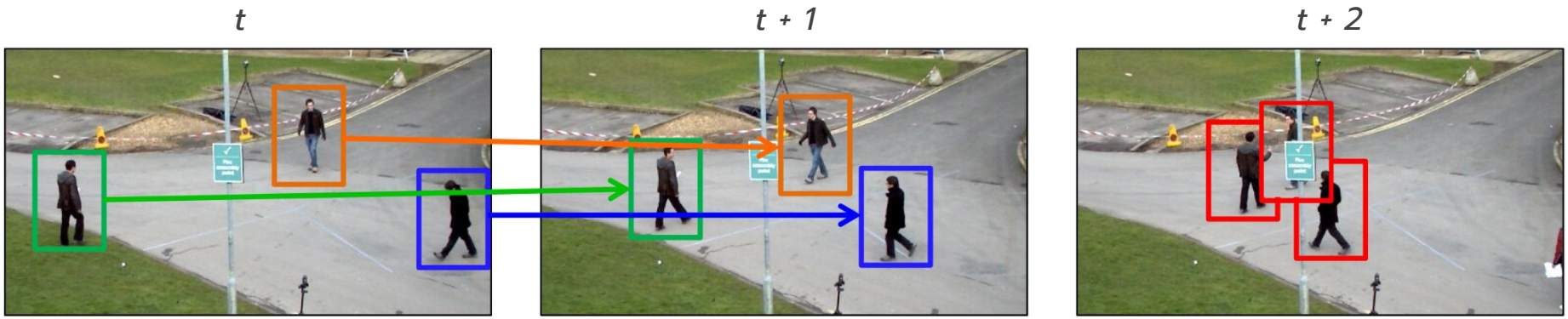


1. Track initialization (e.g. using a detector)

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Offline Tracking : Frame-by-Frame



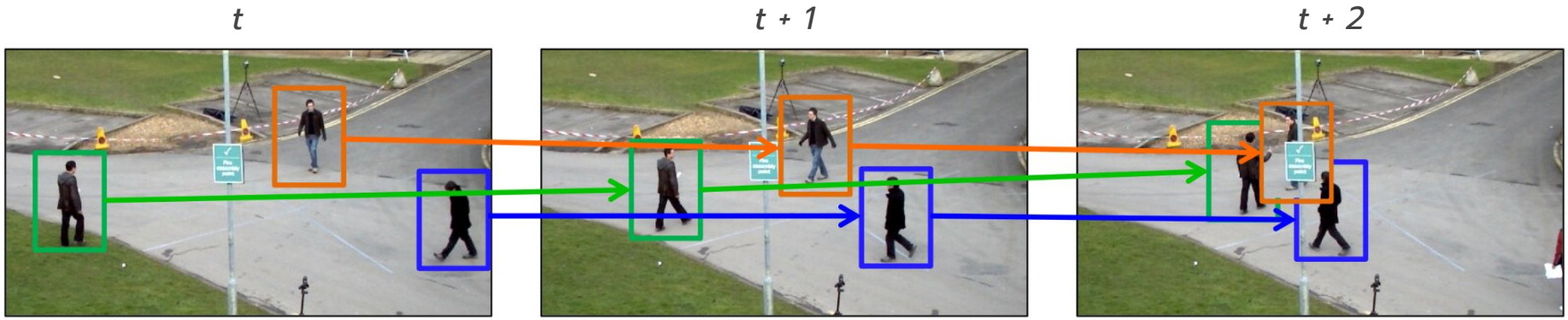
1. Track initialization (e.g. using a detector)

2. Matching detections at *t* with detections at *t+1*

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Offline Tracking : Frame-by-Frame



1. Track initialization (e.g. using a detector)

2. Matching detections at *t* with detections at *t+1*
3. Repeat for every pair of frames

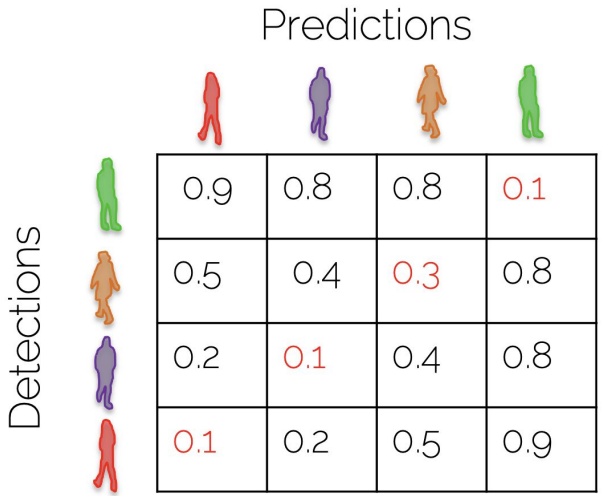
Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Frame-by-Frame

Bipartite matching

- Define distances between boxes (e.g., IoU, pixel distance, 3D distance)
- Solve the unique matching with e.g., the Hungarian algorithm*
- Solutions are the unique assignments that minimize the total cost



Demo: http://www.hungarianalgorithm.com/solve.php

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/

Multiple Object Tracking

Frame-by-Frame

Problems with frame-by-frame tracking

- Cannot recover from errors. If a detection is missing from a frame, we have to end the trajectory.
- All decisions are essentially local
- Hard to recover from errors in the detection step

Solution: find the minimum cost solution for ALL frames and ALL trajectories

⇒ Graph-based MOT

Slide adopted from: Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep., "Computer Vision III: Detection, Segmentation and Tracking (CV3DST)",the Technical University of Munich, https://dvl.in.tum.de/teaching/cv3dst-ss20/