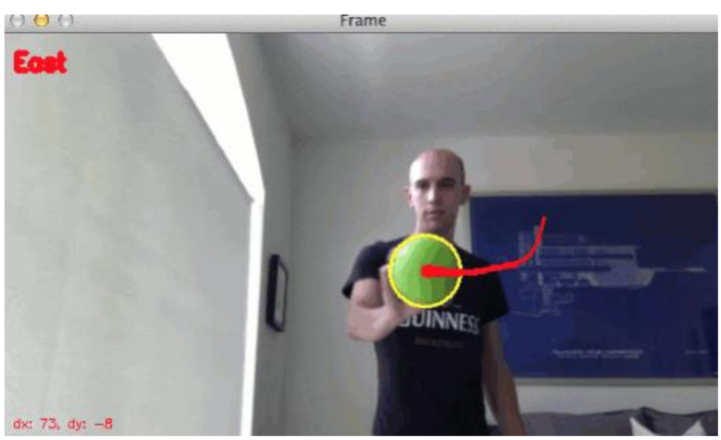
OBJECT TRACKING

EPITA 2024

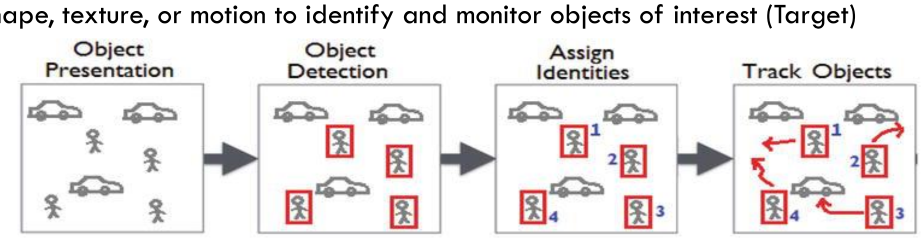
IS IT CLEAR, WHAT TRACKING IS?



IS IT CLEAR, WHAT TRACKING IS?

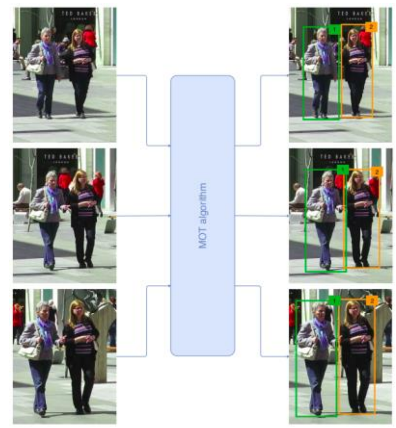
WHAT IS OBJECT TRACKING?

- Processes of locating specific objects, maintaining their identities and yielding their individual trajectories within a sequence of images or frames in a video stream
- The primary goal: determine the object's position and often its trajectory as it moves through the scene over time
- Object tracking algorithms employ computer vision techniques and may utilize features like color, shape, texture, or motion to identify and monitor objects of interest (Target)



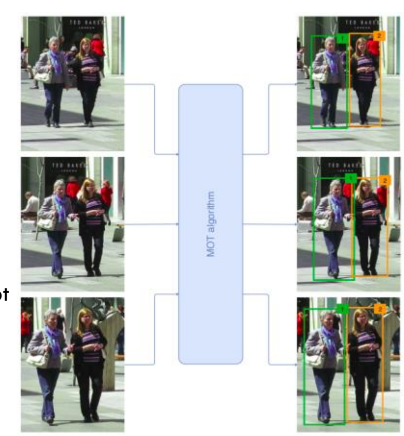
OBJECT TRACKING COMPONENTS

- **Detection**
  - In each frame, detect the presence of the target object
  - This is typically done by drawing a bounding box around the object (or by representing the object as a set of points a set of keypoints)
- **Feature Extraction**
  - Capture information about the object's appearance, such as color, texture, shape, or keypoints (e.g., SIFT, ORB, or deep learning-based features)
- **Matching**
  - Compare the features of the detected candidates with the features of the target object from the initial frame
  - Use a matching or similarity measure to determine which candidate is the most likely continuation of the tracked object



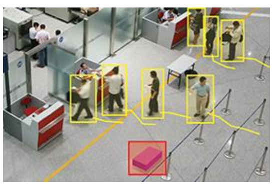
OBJECT TRACKING COMPONENTS

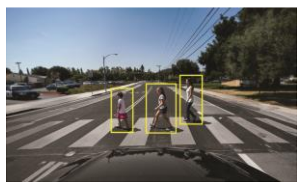
- **Localization**
  - Use motion models or predictive methods to estimate the object's new location
- **Filtering and Smoothing**
  - Apply filtering techniques (e.g. Kalman filter or particle filter) to improve tracking accuracy and handle noise or uncertainties in the data
- **Updating**
  - Update the position and orientation of the tracked object
  - Update the features and appearance model of the tracked object to adapt to changes in illumination, scale, or appearance
- **Termination**
  - Decide when to stop tracking. Use predefined criteria, such as the object leaving the frame or a tracking confidence threshold being exceeded



APPLICATIONS

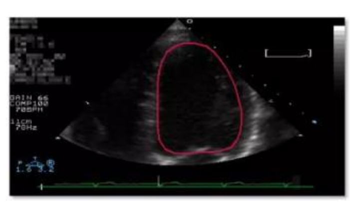
- **Video Surveillance** : to monitor and track individuals or objects of interest within a camera's field of view (public places, airports, banks, and smart homes)
- **Autonomous Vehicles:** to detect and track pedestrians, vehicles, and obstacles in real-time, ensuring safe navigation for self-driving cars and other autonomous vehicles





APPLICATIONS

- **Augmented Reality (AR):** to enhance experiences in gaming, education, and navigation
- **Human-Computer Interaction:** to enable gesture recognition and hand tracking
- **Robotics:** to locate and manipulate objects in various environments, such as in manufacturing, warehouse automation, and healthcare settings
- **Healthcare:** to monitor patient movements and vital signs, as well as to track medical equipment and supplies in hospitals



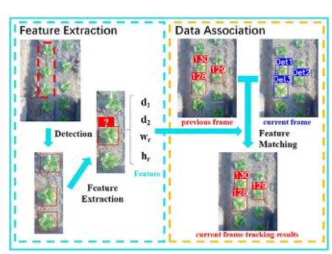
APPLICATIONS

- **Drones and Aerial Surveillance:** to follow and capture images or video footage of moving targets
- **Sports Analytics:** to monitor and analyze the movements of players and the ball during games

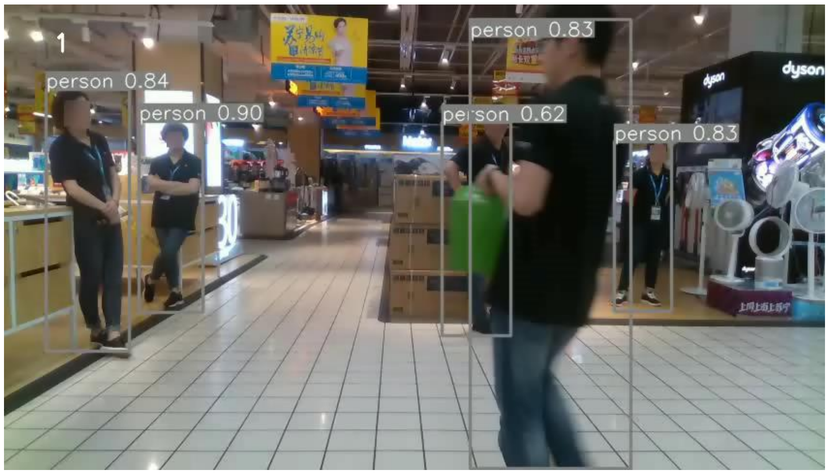
APPLICATIONS

- **Wildlife Conservation:** to monitor and study the behavior of animals in the wild, helping with wildlife conservation efforts
- **Industrial Automation:** to track the movement of products and components, ensuring quality control and process efficiency in manufacturing and assembly lines
- **Agriculture:** to monitor crop health, track the movement of livestock, and automate tasks such as harvesting and spraying in precision agriculture





OBJECT TRACKING CHALLENGES



OBJECT TRACKING CHALLENGES

- Low Resolution/ Limited resolution *(eg. Video captured from a low-end phone)*

- Scale Variation *(Scale of objects)*

Ration of the bounding boxes of the first frame and the current frame is out of the range

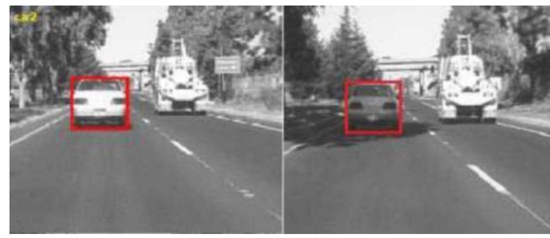
- Change the target position (*Pose, articulation)*

During the movement, target may be rotated, deformed …

OBJECT TRACKING CHALLENGES

❑ Illumination Variation:

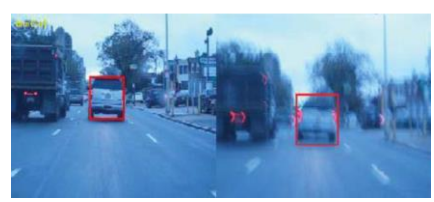
Illumination in the target region is significantly changed



❑ Background Clutters

Background near the target has a similar color or texture as the target

OBJECT TRACKING CHALLENGES



❑ Non-Linear motion

❑ Fast Motion

❑ Occlusion

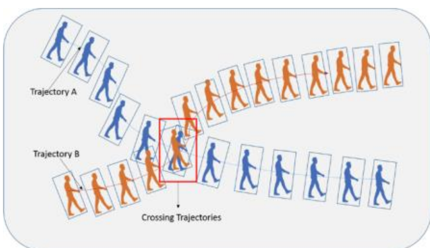
<u>**Most critical challenge !**</u>

Target is partially or fully occluded => ID switches or fragmentation of trajectories

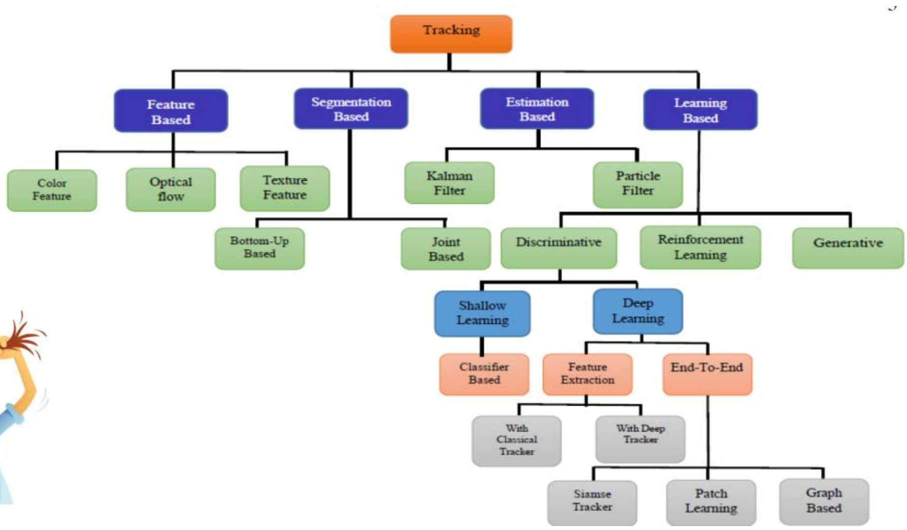
OBJECT TRACKING CHALLENGES

- Initialization and termination of tracks
- Similar appearance *(Same color of clothes, accessories etc.)*
- Interaction among multiple objects

MOT



HOW CLASSIFY OBJECT TRACKING METHODS ?



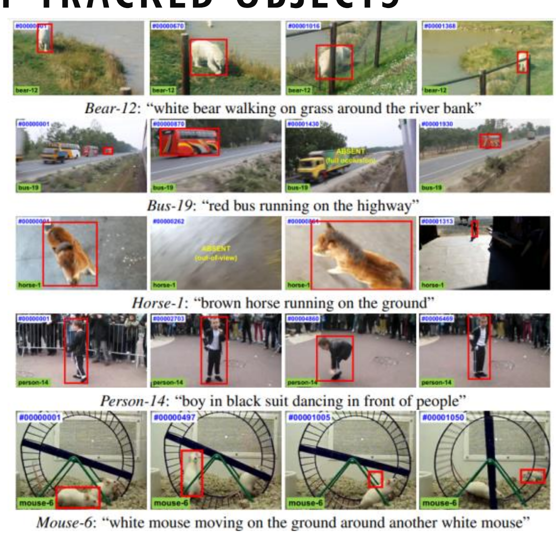
TYPES OF OBJECT TRACKING

Object tracking can be divided into types based on different properties such as the number of target objects, the number of camera viewpoints, initialization method, the way to process the data or the output type

- **Single-Object Tracking (SOT)** vs. **Multi-Object Tracking (MOT)**
- **Single-Camera Tracking (SCT)** vs. **Multi-Camera Tracking (MCT)**
- **Detection-Based tracking (DBT)** vs. **Detection-Free Tracking (DFT)**
- **Online tracking** vs. **Offline tracking**
- **Stochastic tracking** vs. **Deterministic tracking**

CATEGORIZATION: NUMBER OF TRACKED OBJECTS

- Single-Object Tracking (SOT)
  - Locating and tracking **ONE** target object throughout an input video, while all other objects are ignored
  - It faces challenges such as handling appearance changes, scale variations, and occlusions
  - Typically used when monitoring or analyzing a single target is the primary objective



Bear-12: "white bear walking on grass around the river bank"

Bus-19: "red bus running on the highway"

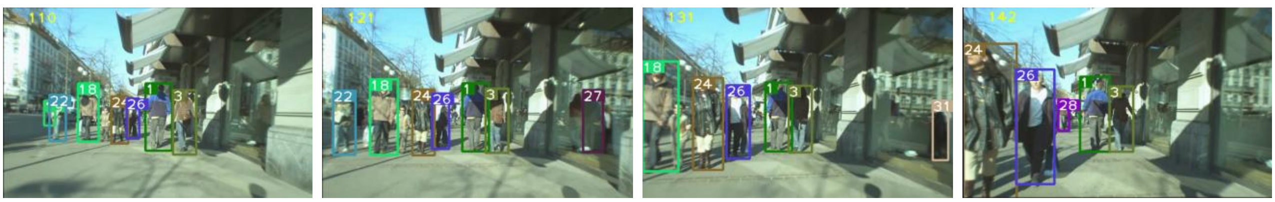
Horse-1: "brown horse running on the ground"

Person-14: "boy in black suit dancing in front of people"

Mouse-6: "white mouse moving on the ground around another white mouse"

CATEGORIZATION: NUMBER OF TRACKED OBJECTS

- Multiple-Object Tracking (MOT)
  - Locating and tracking **MULTIPLE** objects simultaneously throughout an input video
  - Objects can be of the same or different classes
  - MOT is more complex than SOT because it involves handling occlusions (when objects overlap), tracking objects with similar appearances, and maintaining identity consistency for each tracked object
  - Deep learning-based approaches are currently the mainstream in this work



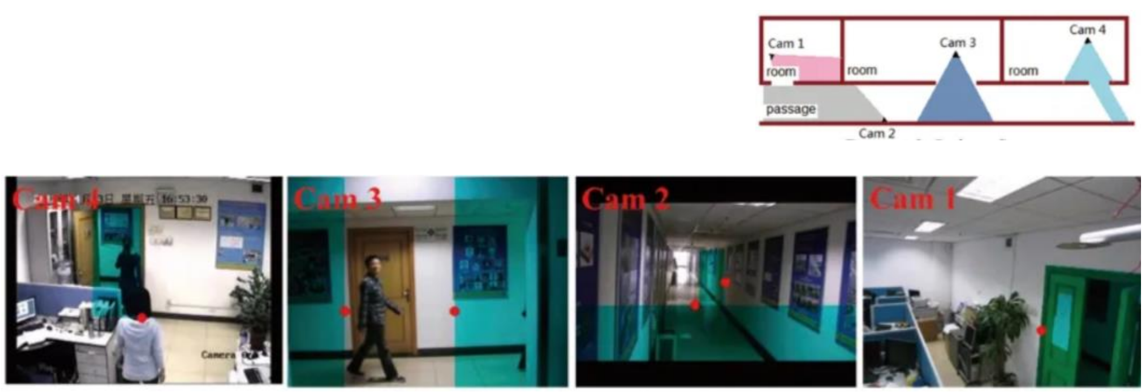
CATEGORIZATION: CAMERA'S NUMBER

- **SCT (Single-camera Tracking):**
  - Uses a single camera to capture the video sequence. All tracking data is derived from this single perspective
  - Limited to the field of view of the single camera. Blind spots or occlusions might lead to lost tracking
  - Generally simpler in terms of data processing since it's dealing with data from a single source
  - May be vulnerable to tracking errors due to occlusions, drastic changes in lighting, or fast-moving objects
  - The viewpoint in the video stream may be fixed in the case of stationary camera or varied in the case of moving camera
    - Static camera: Camera motion compensation is not necessary but a motion model predicting how a target object is moving in the upcoming frames is usually required
    - Mobile camera: Both camera motion compensation and object motion model are required
  - No need to reconcile different viewpoints or manage identities across cameras

CATEGORIZATION: CAMERA'S NUMBER

- **MCT (Multiple-Camera Tracking):**
  - Utilizes multiple cameras, often placed at different viewpoints, to capture video sequences simultaneously.
  - The data from cameras must by fused to provide a more comprehensive tracking result => Can mitigate the effects of blind spots and occlusions
  - More complex and more challenging due to the need to calibrate multiple cameras, synchronize their data, and reconcile differences in viewpoints or object appearances
  - Divided into:
    - overlapping fields-of-view (FOVs) : exploited in a small observation area (e.g. a room)
    - non-overlapping FOVs
  - Requires mechanisms to ensure consistent object identity across different camera views
    - Additional module called Re-IDentification (ReID) to compute the similarity among local tracks from different cameras (for non-overlapping FoVs)
    - Local track association step can leverage extra information such as ground-plane coordinates or moving patterns of objects to compute the similarity among local tracks from different cameras (for overlapping FoVs)

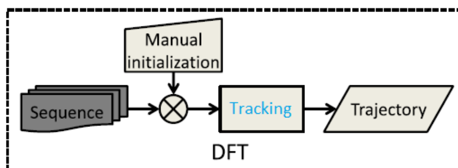
MCT (MULTIPLE-CAMERA TRACKING)



CATEGORIZATION: INITIALIZATION METHOD

Detection-Free-Tracking (DFT)

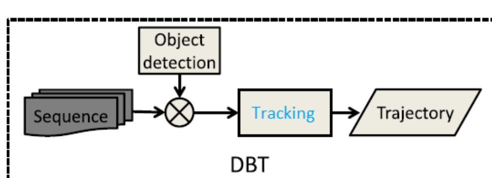
- **Initialization**: Identify fixed number of objects of interest in the initial frame (e.g., through manual user input or a region of interest)
- Cannot deal with the case that object appear !
- **Object Tracking**: Track the identified objects across frames using motion estimation, appearance modeling, and feature tracking (e.g., optical flow or keypoints) without explicitly detecting them
- May struggle in cases with occlusions, significant object appearance changes, or complex interactions between objects
- Can be robust in scenarios with consistent object motion and appearance



CATEGORIZATION: INITIALIZATION METHOD

Detection-Based Tracking (DBF) = Tracking-by-detection

- **Object Detection**: Detect objects in each frame (Utilizes object detectors)
- **Data Association**: Link detections across frames to form object tracks, often using methods like the Hungarian algorithm or data association techniques like Kalman filters or Particle filters
- Can handle complex scenarios with multiple objects and occlusions
- May be computationally expensive due to the need for object detection in each frame
- Vulnerable to detection errors, which can propagate into tracking errors
- **Better detection = Better tracking!**



CATEGORIZATION: DBT VS. DFT

Item	DBT	DFT
Initialization	Automatic, imperfect	Manual, perfect
Number of objects	Varying	Fixed
Applications	Specific type of objects	Any type of objects
Advantages	Ability to handle varying number of objects	Free of object detector
Drawbacks	Perfomance depends on object detection	Manual initialization

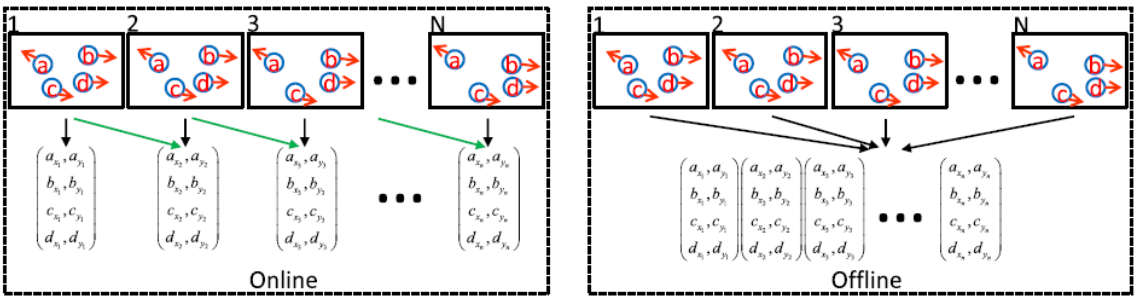
CATEGORIZATION: PROCESSING MODE

Object tracking can be performed in an online or offline manner, regardless of being SOT, MOT, SCT, or MCT

The difference is whether observations from future frames are utilized when handling the current frame !

- Online tracking = Sequential tracking
  - Image sequence is handled in a step-wise manner
  - Tracking only relies on the past information available up to the current frame
  - Suitable for real-time applications like surveillance, autonomous vehicles, and robotics
- Offline tracking
  - Observations from all the frames are required to be obtained in advance and are analyzed jointly to estimate the final output
  - Usually superior to online tracking in terms of accuracy
  - Due to computational and memory limitation, it is not always possible to handle all the frames at once => Split the data into shorter video clips

CATEGORIZATION: PROCESSING MODE



CATEGORIZATION: ONLINE VS. OFFLINE

Item	Online	Offline
Input	Up-to-time observations	All observations
Methodology	Gradually extend existing trajectories	Link observations into trajectories
Advantages	Suitable for online tasks	Obtain global optimal solution theoretically
Drawbacks	Suffer from shortage of observation	Delay in outputting final results

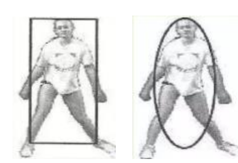
CATEGORIZATION: TYPE OF OUTPUT

They differ in their modeling of uncertainty => optimization methods

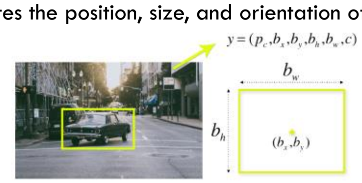
- Stochastic tracking
  - Explicitly models uncertainty. It recognizes that measurements may be noisy, and the object's true state is not known with certainty
  - More suitable for scenarios where measurements may be noisy or where the object's motion is uncertain
  - Example: Incorporate probabilistic models, such as Bayesian filters (e.g. Kalman filter) or Sequential Monte Carlo methods (Particle filter), to estimate the object's state while considering uncertainty in the measurements and object dynamics
  - **Different tracking results in different running times !**
- Deterministic tracking
  - Assumes no uncertainty and works well in scenarios with highly accurate measurements and well-defined object dynamics. It operates under the assumption that there is no randomness or uncertainty in the tracking process.
  - Example: Data association method like Hungarian algorithm will produce deterministic tracking results
  - **Tracking output is constant when running the methods multiple times !**

HOW REPRESENT OBJECTS IN OBJECT TRACKING?

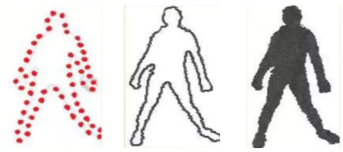
- **Keypoints and Feature Points**
  - Keypoints = distinctive points or regions within the object, such as corners or interest points
  - Keypoints serve as landmarks, and their positions are tracked over time
  - Center point or Set of Points (Classic Feature extraction methods or Deep Learning Features)



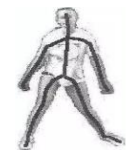
- Primitive Geometric Shapes: **Bounding Boxe**, Ellipse, *etc.*
  - Bounding box = rectangular region that encapsulates the target object
  - The simplest and most commonly
  - Tracking algorithm estimates the position, size, and orientation of this box in each frame



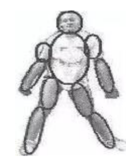
HOW REPRESENT OBJECTS IN OBJECT TRACKING?



- **Shape models / Object Silhouette or Contour**
  - Contour representation defines the boundary of the object, object defined as a set of connected lines or curves
  - The region inside of the contour = Silhouette of the object



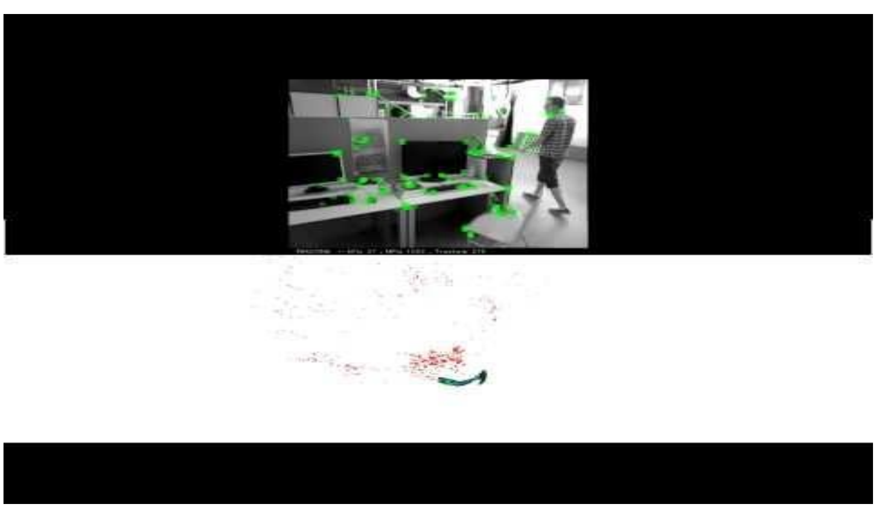
- Squeletal models



- Articulated Shape Models
  - Composed of body parts that are held togheter with joints

- Appearance models
  - Capture the visual appearance of the object through histograms of color, texture, or other visual features

EXEMPLES: POINT/KEYPOINT TRACKING



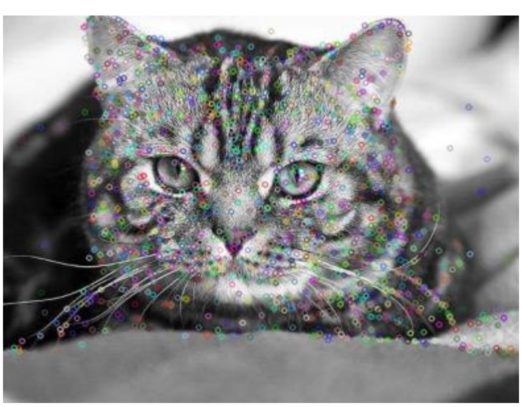
EXEMPLES: SQUELETON TRACKING

EXEMPLES: OBJECT (BOUNDING BOX + CONTOURS) TRACKING

KEYPOINTS

- Keypoints = Interest Points = Feature Points
  - Distinctive locations or regions in an image that have unique characteristics. It stands out due to local variations in intensity, texture, color, or other visual attributes
  - Is robust to certain tranformations such as scaling, rotation, and changes in lighting conditions
  - Can be reliably found across multiple images of the same scene
- Keypoint extraction is typically followed by feature description
  - Local image information in a region around the point

KEYPOINTS



EXAMPLE OF INTEREST POINT DETECTORS

- **Harris Corner Detection**
  - Identifies key points by analyzing local intensity changes in an image. It looks for regions where intensity gradients change in multiple directions, indicating the presence of corners
- **Shi-Tomasi Corner Detection**
  - An improvement over Harris corner detection, offering better localization accuracy
  - Selects keypoints based on the smallest eigenvalues of the gradient matrix
- **Features from Accelerated Segment Test (FAST)**
  - Identifies keypoints by comparing the brightness of a central pixel to its surrounding pixels
- **Sale-Invariant Feature Transform (SIFT)**
  - Analyzes the local image gradients and applies Gaussian filtering to identify distinctive points
- **Learned detectors**
  - SuperPoint

KEYPOINTS DESCRIPTION

- **Histogram of Oriented Gradients (HOG)**
  - Captures the distribution of local gradient orientations in an image patch around a key point
  - Size of HOG features depends on the parameters used in the feature extraction process, such as cell and block sizes
- **SIFT**
  - Represents the distribution of gradient magnitudes and orientations in a region around a key point
  - Descriptor size: 128
- **Speeded-Up Robust Features (SURF)**
  - Uses the sum of Haar wavelet responses to represent the intensity information around a key point
  - Descriptor size: 64
- **Binary Robust Independent Elementary Features (BRIEF)**
  - Captures the intensity comparisons between pairs of pixels in an image patch around a key point
  - **Oriented FAST and Rotated BRIEF (ORB)** = FAST + BRIEF

SIFT

- Is known for its invariance to scale, rotation, and illumination changes
- Simplified and generalized flowchart of the key steps involved in the SIFT keypoint detection process:
  - Scale-Space Pyramid Generation
  - Difference of Gaussians (DoG) Pyramid
  - Keypoint Localization

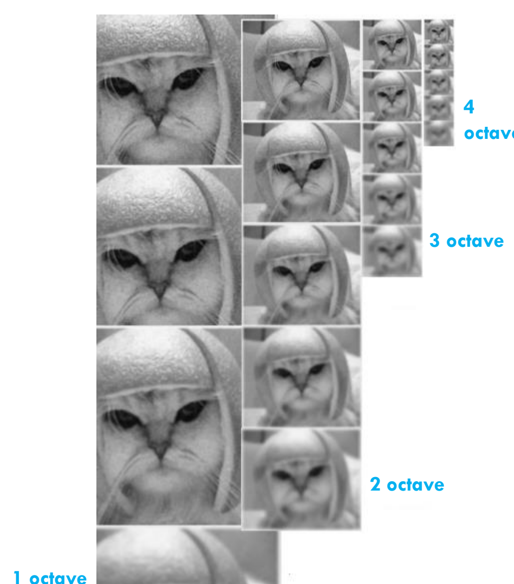
  Detector

  - Keypoint Descriptor Calculation
    - Create a local image patch for each key point based on orientation
    - Compute a gradient histogram for the patch
    - Generate the SIFT descriptor from the gradient histogram

  Descriptor

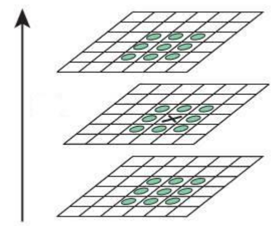
SIFT: SCALE-SPACE PYRAMID

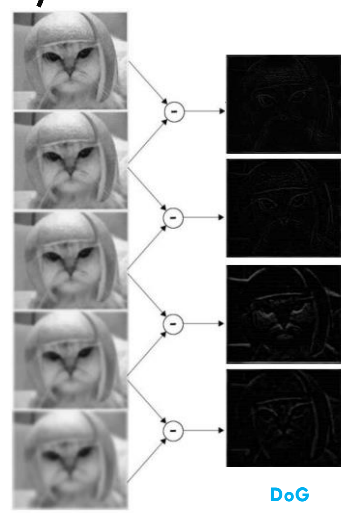
- Create a series of blurred downsampled images at different scales
- Determine the number of octaves and scales per octave



SIFT: DIFFERENCE OF GAUSSIANS (DOG) PYRAMID

- Compute the difference between adjacent scales to create a DoG pyramid
- Identify local extrema (potential key points) in the DoG images
- Perform non-maximum suppression to retain strong extrema



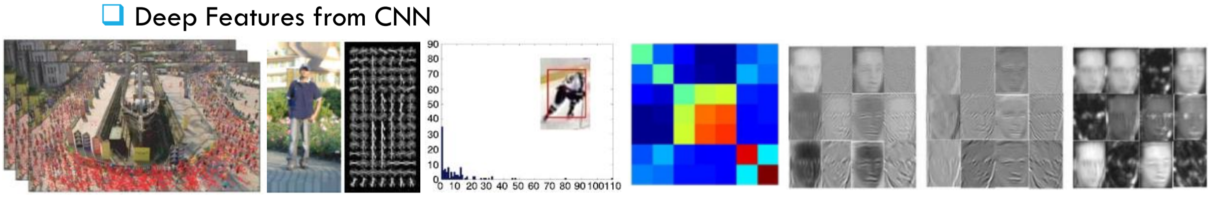


Gaussien DoG

VISUAL REPRESENTATION

- Describes object according to different kind of features => Exploite some similarity criterion by using those features
- Local features — Optial flow
- Region features
  - Zero-order type : Values of pixels are not compared — Color histogram, Raw pixel template, …
  - First-order type — Gradient-based representation like HoG, …
  - Up-to-second-order type — Region covariance matrix

  (First-order and up-to-second-order type: Discrepancy values amond pixels are computed)
- Deep Features from CNN

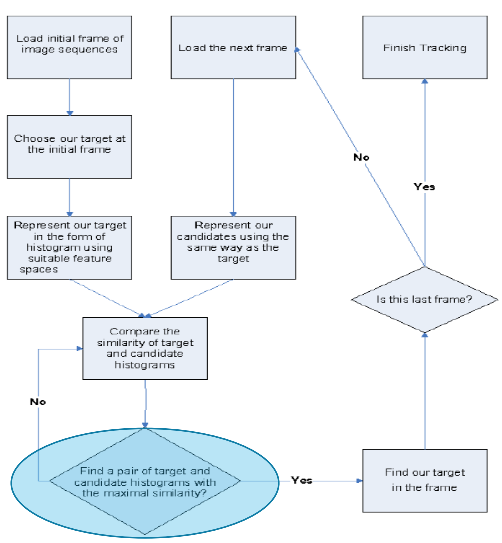


Optial flow HoG Color Covariance matrix Gabor wavelets

OBJECT APPAREANCE MODEL

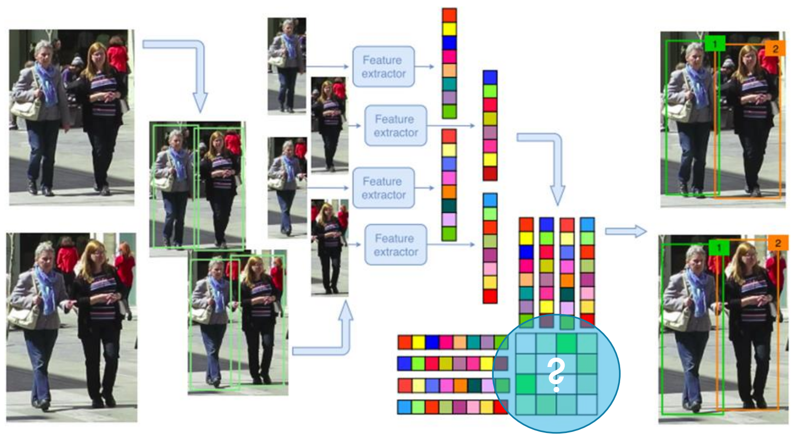
- Single cue
  - One type of visual feature
- Multiple cues
  - Combine diffrent types of features
  - Concatenation : Different kinds of features can be concatenated
  - Summation: Values from different features are balanced with weights
  - Product : Values are multiplied
  - …

FLOWCHART OF THE TRACKING USING HISTOGRAM

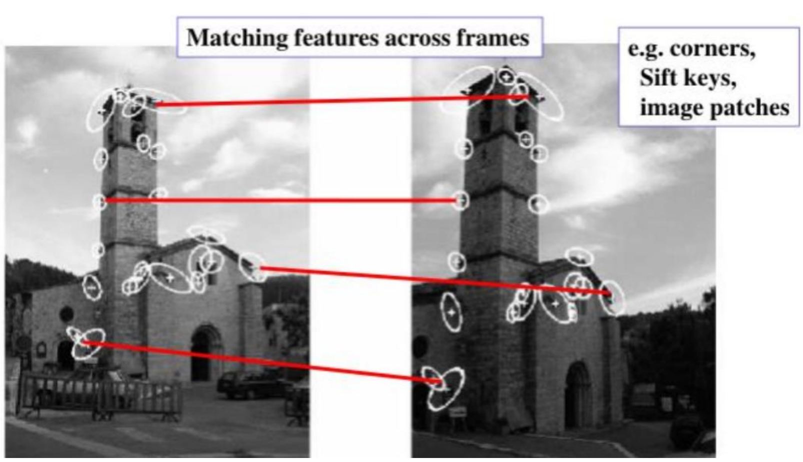


FLOWCHART OF RE-ID BASED OBJECT TRACKING

❑ Goal: Compute the affinity between two observations

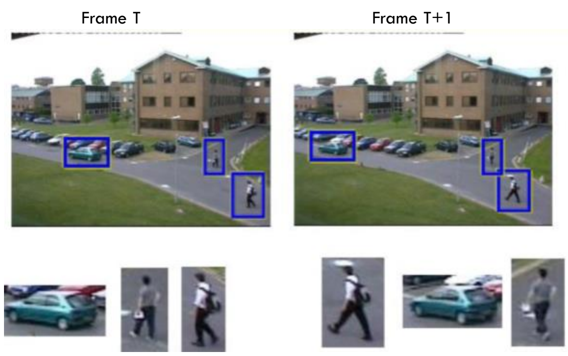


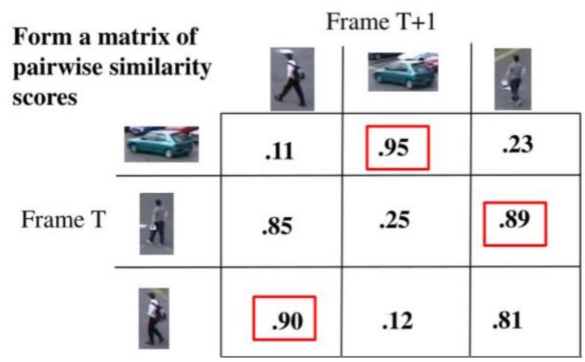
CORRESPONDENCE PROBLEM



CORRESPONDENCE PROBLEM

Match up detected blobs across video frames





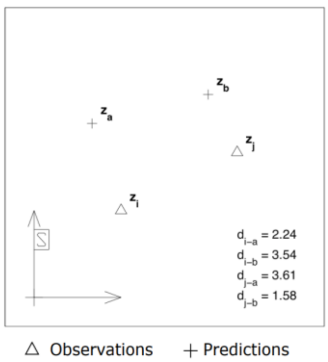
SOME SIMILARITY MEASURE/METRICS…

- **Correlation-based** : A high correlation indicates a strong match
  - Correlation Coefficient
  - Normalized Cross-Correlation (NCC)
- **Distance measurements**
  - Euclidean Distance (L2 Norm)
  - Mahalanobis Distance: takes into account the covariance between variables when tracking objects in multi-dimensional space
  - Manhattan Distance (L1 Norm): Measures the sum of absolute differences between the coordinates of two points. It is suitable for tracking on grids or when motion is constrained to vertical and horizontal movements
  - Minkowski Distance
  - Cosine Distance or Dissimilarity: Measures the cosine of the angle between two vectors
  - Hausdorff Distance: used for tracking objects with complex shapes
  - Jaccard Distance: used for comparing sets of features
- Histogram-Based Metrics : Measures the amount of overlap between the histograms. i.e. Histogram Intersection and Bhattacharyya distance

  A smaller area indicates a better match, meaning that the two histograms have a higher degree of overlap

SOME SIMILARITY MEASURE/METRICS…

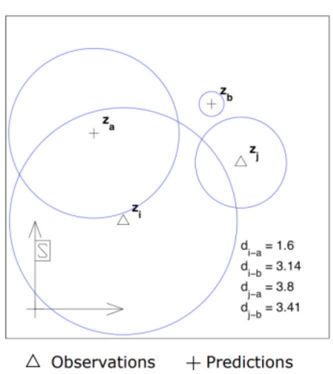
- **Euclidian distance**
  - Takes into account: ✓ Position ✗ Uncertainty ✗ Correlations



It seems that i-a and j-b belong together !

SOME SIMILARITY MEASURE/METRICS…

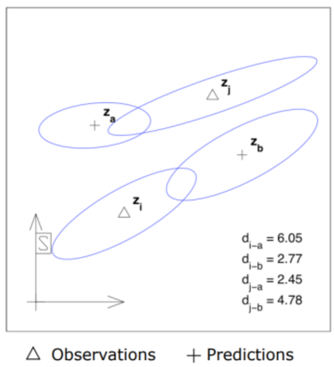
- **Mahalanobis distance** with **diagonal** covariance matrices
  - Takes into account: ✓ Position ✓ Uncertainty ✗ Correlations



Now, i-b is "closer" than j-b !

SOME SIMILARITY MEASURE/METRICS…

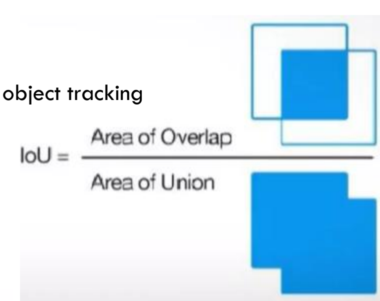
- **Mahalanobis distance**
  - Takes into account: ✓ Position ✓ Uncertainty ✓ Correlations



It's actually i-b and j-a that belong together!

IOU/ JACCARD INDEX

- IoU:
  - Widely used evaluation metric in object detection, image segmentation, and object tracking
  - Calculated as the ratio of the intersection area to the union area
  - Quantifies the overlap between two sets or regions
  - Ranges from 0 to 1
- Jaccard index = 1 - IoU



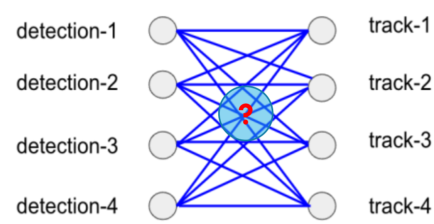
DATA ASSOCIATION

Data association is the process of associating uncertain measurements to known tracks!

Given: N tracked target trajectories and M new sensor observations

Problem: deciding which target generated which observation

- Key component of MOT with tracking-by-detection strategy
  - Link existing tracks and new detections at each frame so that it forms trajectories of multiple objects
  - Produce sequences of detections with unique identities
- However, when the association is ambiguous, then the assignment decisions become coupled and much harder to solve



DATA ASSOCIATION

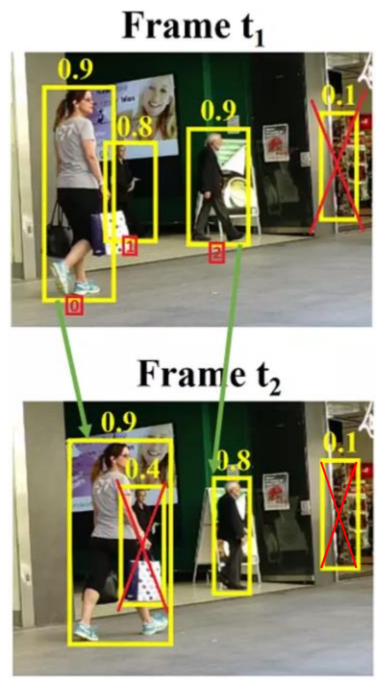
Frame t1

Frame t2

DATA ASSOCIATION

PROBLEM TYPES:

- Target detection
- Single or multiple targets
- False alarm model and rates
  - They can come from sensor imperfections or detector failures
- Single or multiple sensors
- Track creation, maintenance, and deletion

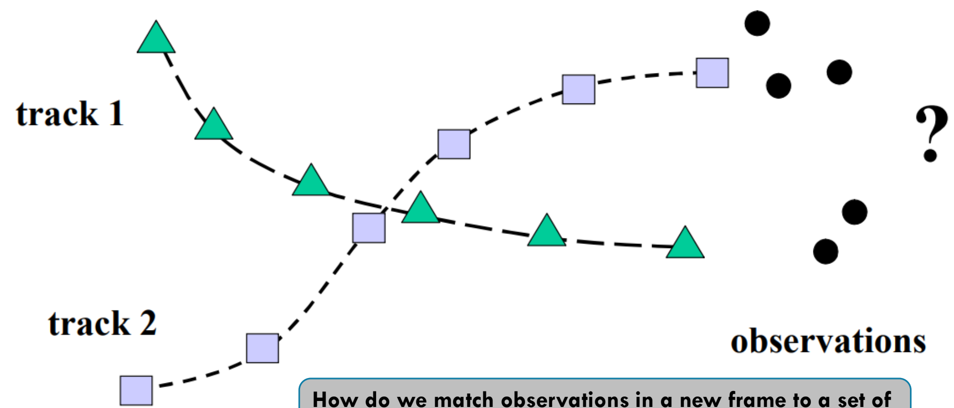


DATA ASSOCIATION

What makes this problem difficult ?

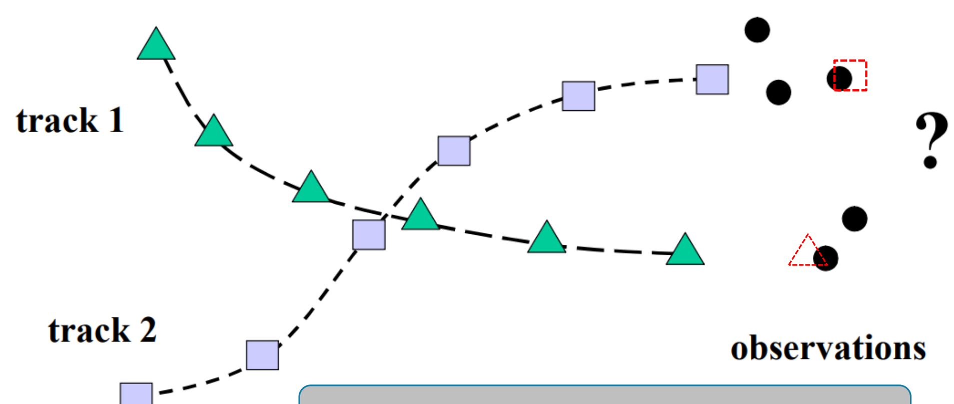
- Multiple targets
- False alarms
- Detection uncertainty (occlusions, sensor failures, …)
- Ambiguities (several measurements in the gate)

DATA ASSOCIATION: TRACK MATCHING



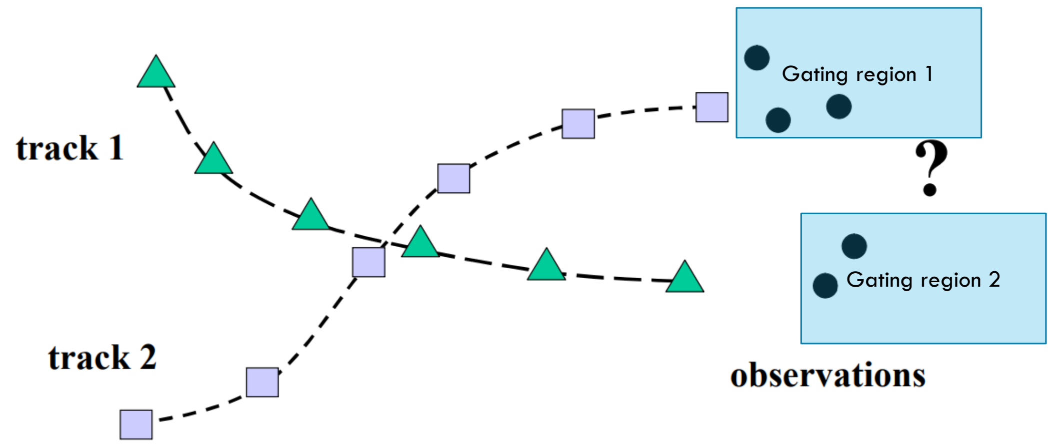
How do we match observations in a new frame to a set of tracked trajectories?

DATA ASSOCIATION: TRACK MATCHING



Predict next target position along each track ….

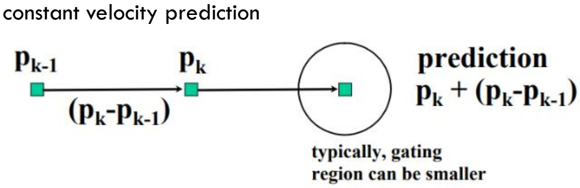
DATA ASSOCIATION: GATING



DATA ASSOCIATION: GATING

- A method of pruning matches that are geometrically unlike from the start
- Allows us to decompose matching into smaller subproblems
- Validation gate: Simpler Prediction/Gating:
  - Constant position + bound on maximum inter-frame motion

  - Three frame constant velocity prediction



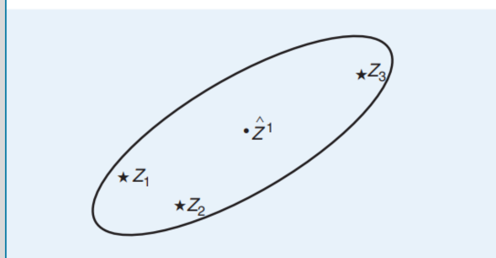
DATA ASSOCIATION: VALIDATION GATE

Validation gate/Validation region

- The area around the predicted measurement $\widehat{z^1}$ in which pairing are accepted
  - Region of acceptance such that 100(1-α)% of true measurements are rejected

Typical values for α are 0.95 or 0.99

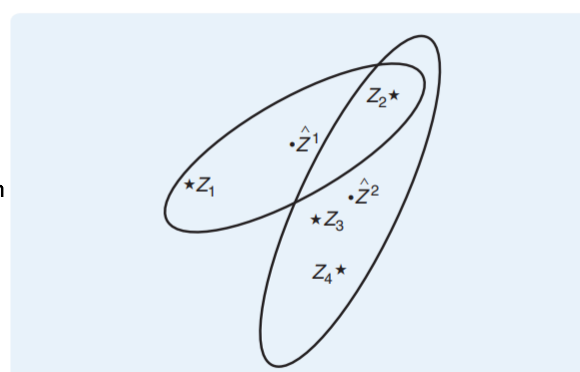
- The shape of the validation gate is a hyper-ellipsoid
  - ➢ The elliptical shape of the validation region is a consequence of the assumption that the error in the target's predicted measurement is Gaussian



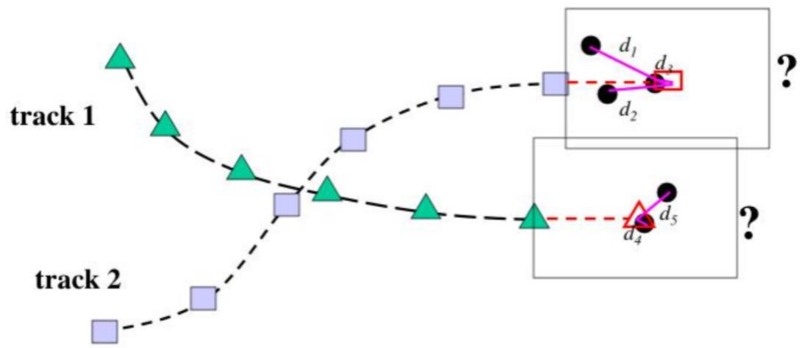
DATA ASSOCIATION: VALIDATION GATE

MOT case:

- ❑ The validation regions are the ellipses centered at the predicted measurements $\widehat{z_1}$ and $\widehat{z_2}$
- ❑ Each of the measurements in the validation region of one of targets could have originated from the corresponding target or from clutter Clutter = False alarm
- ❑ Two targets with a measurement z_2 in the intersection of their validation regions
- ❑ In this case, **joint association** events must be considered!

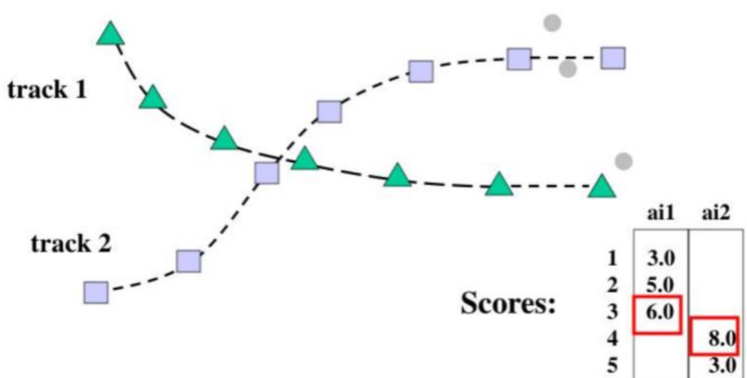


DATA ASSOCIATION: TRACK MATCHING



Match score based on distance to predicted location. Could also incorporate similarity of appearance (e.g. color histogram similarity, patch correlation)

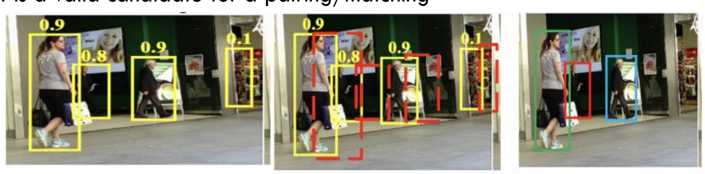
DATA ASSOCIATION: TRACK MATCHING



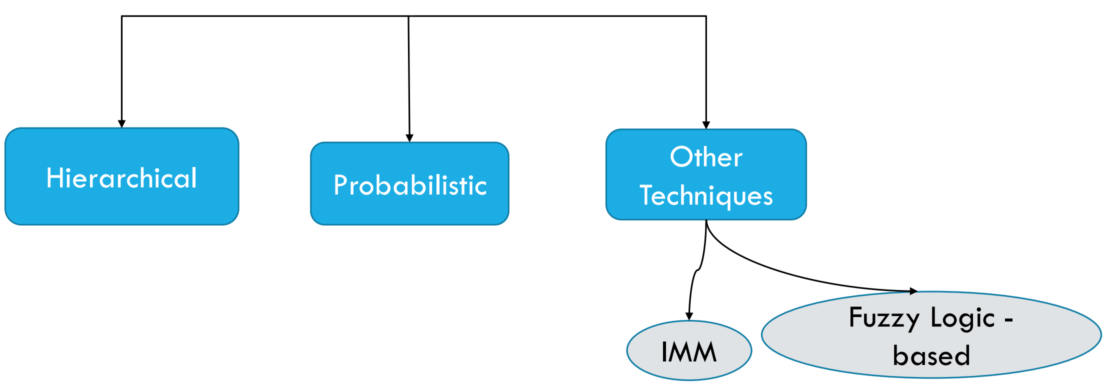
Determine the best match and extend the trajectory

DATA ASSOCIATION: OVERALL PROCEDURE

- Make observations (= measurements)
  - Measurements = raw data or the output of some target detector (e.g. people detector)
- Model object appearance
- Predict the measurements from the tracks
  - This yields an area in sensor space where to expect an observation. The area is called validation gate and is used to narrow the search
- Check if a measurement lies in the gate
  - If yes, then it is a valid candidate for a pairing/matching



DATA ASSOCIATION: APPROCHES



DATA ASSOCIATION: HIERARCHICAL

Tracklet Assosiation

Greedy Data Association

Hungarian Algorithm based

LSTM-based Data Association

Greedy beacause:

- Assosiation decisions made at each time step based on local information, without considering the global optimality of the solution

NN Matching

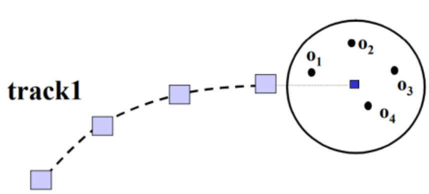
Thresholding

- The most commun!

- It predicts the assignment for each measurement to one target at a time
- The input of the network: a pairwise-distance matrix, which is an Euclidean distance matrix between the predicted state of target and measurement

DATA ASSOCIATION: NEAREST NEIGHBOR

- **Nearest Neighbor (NN) Matching**
  - Compute distance (according to the chosen similarity metrics) to all measurements
  - Choose "best" one to incorporate into track => Accept the closest measurement !
  - Update the track



	ai1	
1	3.0	
2	5.0	
3	6.0	
4	$\boxed{9.0}$	max

a_{i1} = score for matching observation i to track 1

Choose best match = max{a_{11}, a_{12}, a_{13}, a_{14}}

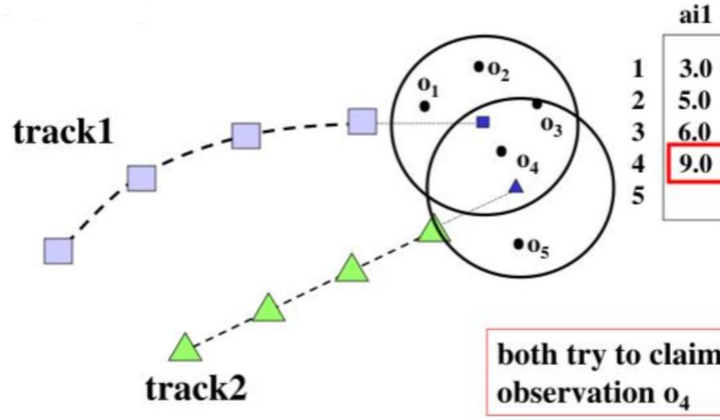
DATA ASSOCIATION: NEAREST NEIGHBOR

- Nearest Neighbor (NN) Matching
- **Nearest Neighbor with Threshold**
  - Extends the nearest neighbor strategy by incorporating a distance threshold: Detections within the threshold of a tracked object are associated with it, while those beyond the threshold are treated as new objects

Deterministic decisions about associations!

DATA ASSOCIATION: TRACK MATCHING

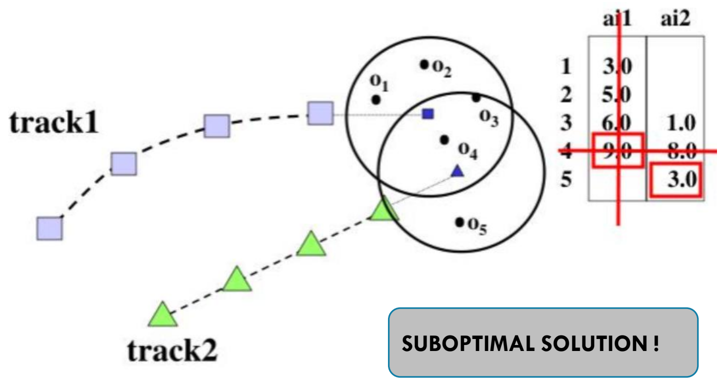
Problem: If we do that independently for each track, we could end up with contention for the same observation when gating regions overlap



	ai1	ai2
1	3.0	
2	5.0	
3	6.0	1.0
4	9.0	8.0
5		3.0

DATA ASSOCIATION: GREEDY (BEST FIRST) STRATEGY

- Assign observations to trajectories in decreasing order of goodness, making sure to not reuse an observation twice



SUBOPTIMAL SOLUTION !

DATA ASSOCIATION: NEAREST NEIGHBOR

1. **Simplicity:** NN Matching is easy to understand and implement, making it a good choice for simple tracking tasks
2. **Efficiency:** Computationally efficient and can process frames in real-time (**Real-time Tracking**)
3. **Low Overhead:** It does not require complex optimization algorithms, making it suitable for lightweight tracking tasks

1. **Sensitivity to Noise:** Highly sensitive to noisy detections, as a single outlier can lead to incorrect associations and tracking failures
2. **Ambiguity:** In cases where multiple objects are close to each other or have similar appearances, NN Matching can lead to ambiguous associations
3. **Lack of Global Context:** It does not consider global context or history when making associations. Each association is made independently for the current frame
4. **Occlusions:** When objects are occluded or reappear in the scene after a brief absence, it can struggle to maintain the correct associations

DATA ASSOCIATION: LINEAR ASSIGNEMENT PROBLEM

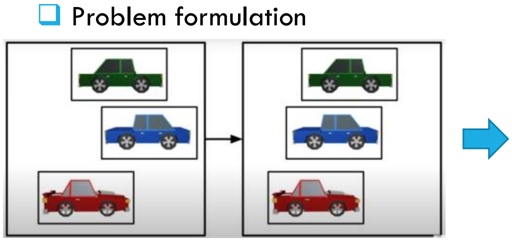
<u>Problem:</u> Choose a 1-1 correspondence that maximizes sum of match scores

	1	2	3	4	5
1	0.95	0.76	0.62	0.41	0.06
2	0.23	0.46	0.79	0.94	0.35
3	0.61	0.02	0.92	0.92	0.81
4	0.49	0.82	0.74	0.41	0.01
5	0.89	0.44	0.18	0.89	0.14

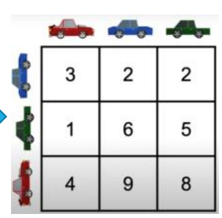
- **Hungarian algorithm**/ Kuhn-Munkres algorithm/Munkres assignment algorithm
  - It finds the best assignment that minimizes the total cost or maximizes the total benefit => Finds global cost minimum!
  - The algorithm always finds a solution
  - <u>Deterministic Nature:</u> It always produces the same solution for the same input !
  - It is suitable for small to moderately sized problems (Complexity (O(n^3)) !

DATA ASSOCIATION: HUNGARIAN ALGORITHM

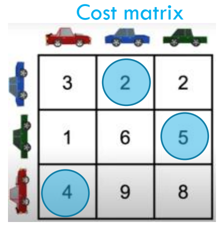
- Problem formulation



Detections Tracks



Cost matrix



- Distance between the centers of the boxes
- The overlap between the boxes
- The similarity of the features

…

How to proceed?

DATA ASSOCIATION: HUNGARIAN ALGORITHM

Step 1 : Row reduction

- Find the lowest cost entry in each row and sustract it from all other entries in that rows

3	2	2
1	6	5
4	9	8

3	2	2
1	6	5
4	9	8

1	0	0
0	5	4
0	5	4

Step 2 : Column reduction

- Find the lowest cost entry in each column and sustract it from all other entries in that columns.

This step ensures that there is at least one zero in each row

1	0	0
0	5	4
0	5	4

1	0	0
0	5	4
0	5	4

•If yes, the initial assignment is optimal, and the algorithm ends.
If the number of lines is not equal to N, **check if the number of marked zeros** (zeros that have been included in the cu

DATA ASSOCIATION: HUNGARIAN ALGORITHM

Step 3 : Test for an optimal assignement

- Cover the matrix with the minimum lines (horizontal or vertical) so that all zeros are covered

1	0	0
0	5	4
0	5	4

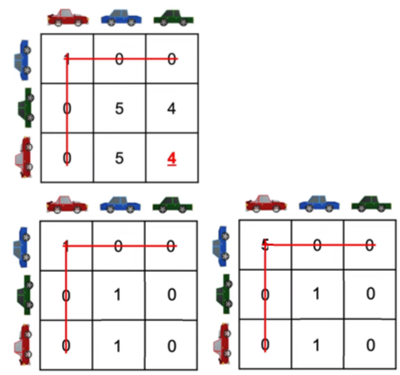
Step 4: Augmenting Path Search

- If there are N (the number of rows or columns) lines drawn, an optimal assignment of zeros is possible and the algorithm is finished
- If the number of lines is not equal to N, check if the number of marked zeros (zeros that have been included in the current assignment) equals the number of rows.
  - If it does, the initial assignment is optimal

DATA ASSOCIATION: HUNGARIAN ALGORITHM

Step 5: Shift zeros

- Find the smallest uncovered value in the cost matrix
- Subtract it from all uncovered values ..
- and add it to all values at the intersection of covered rows and columns



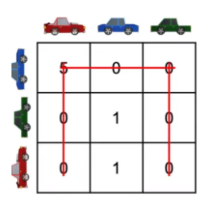
DATA ASSOCIATION: HUNGARIAN ALGORITHM

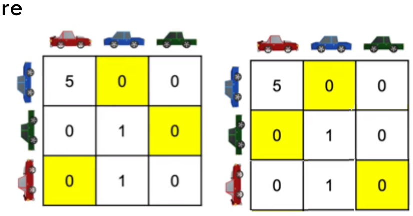
Step 6: Updating Assignment

- Go back to the Step 3
- Repeat the process until you have a complete assignment with the same number of lines as there are rows or columns

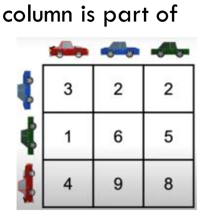
Step 7: Optimal Assignement

- Only one 0 per row or column is part of assignement





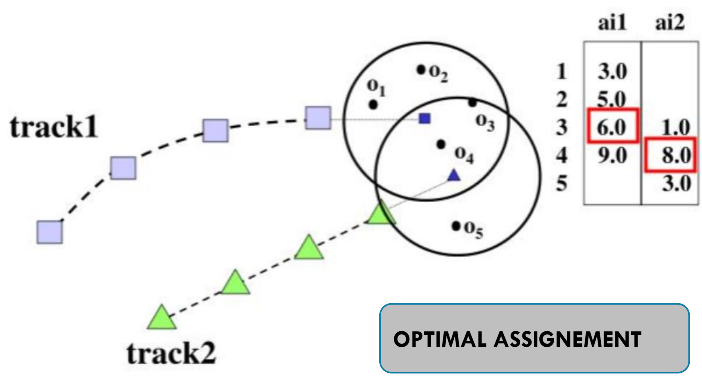
Cost = 2+5+4=11 Cost = 2+1+8=11



TWO POSSIBLE ASSIGNEMENTS !!

DATA ASSOCIATION: HUNGARIAN ALGORITHM

- Each track is forced to claim a diffrent observation



OPTIMAL ASSIGNEMENT

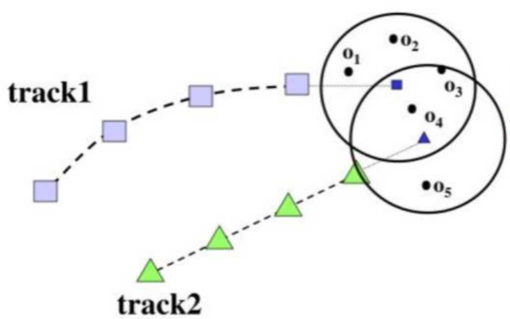
DATA ASSOCIATION: HUNGARIAN ALGORITHM

- **Hungarian algorithm**/ Kuhn-Munkres algorithm/Munkres assignment algorithm
  - It finds the best assignment that minimizes the total cost or maximizes the total benefit => Finds global cost minimum!
  - The algorithm always finds a solution
  - Deterministic Nature: It always produces the same solution for the same input !
  - It is suitable for small to moderately sized problems (Complexity (O(n^3)) !

1. Square Cost Matrix Requirement: Designed for square cost matrices => Number of detections must be equal to the number of tracks. Otherwise, padding the matrix with dummy values is necessary, which can affect the results
2. Computationally expensive : Cubic time complexity (O(n^3)) makes it less suitable for very large-scale problems
3. Space Complexity: O(n^2), which can be problematic for large cost matrices as it may require a significant amount of memory

DATA ASSOCIATION

Nearest Neighbor Greedy association



	ai1	ai2
1	3.0	
2	5.0	
3	6.0	1.0
4	$\boxed{9.0}$	8.0
5		$\boxed{3.0}$

Hungarian algorithm

	ai1	ai2
1	3.0	
2	5.0	
3	$\boxed{6.0}$	1.0
4	9.0	$\boxed{8.0}$
5		3.0