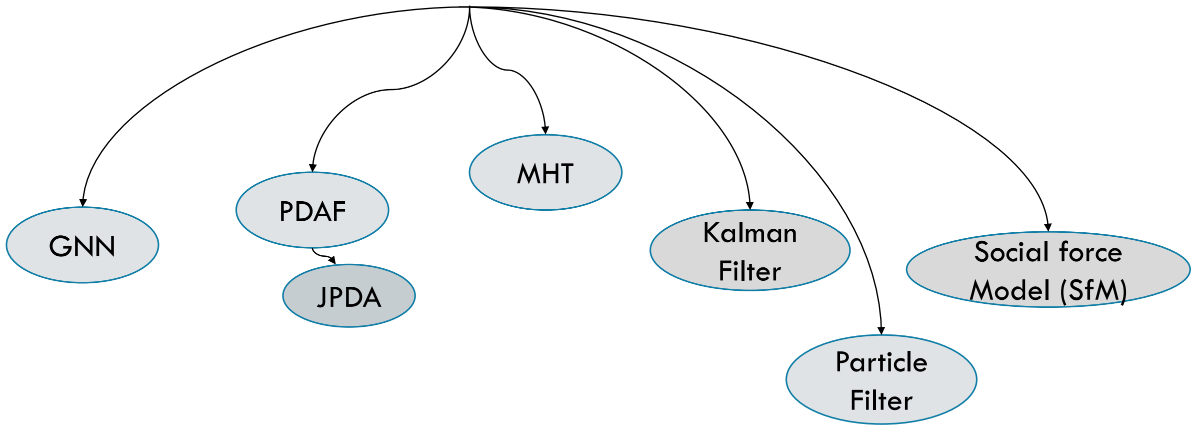
DATA ASSOCIATION: PROBABILISTIC

Techniques that incorporate probabilistic models to associate measurements with objects in tracking systems

- ❑ Consider the likelihood of a measurement originating from a specific track and selects the most likely association based on a probabilistic framework
- ❑ Evaluate the probability of each association being correct
- ❑ Find the globally optimal solution by considering all possible association hypotheses
- ❑ Based on Bayesian data association framework

DATA ASSOCIATION: PROBABILISTIC



DATA ASSOCIATION: PDAF

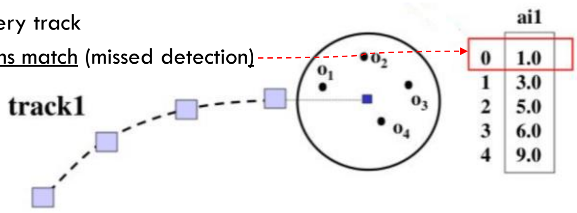
- ❑ PDAF = Probabilistic Data Association Filter
  - ➢ Designed to track a **single** target in the presence of false alarms and missed detections
  - ➢ Bayesian approach that computes the probability of track-measurement associations
  - ➢ Instead of making a single deterministic assignment, it computes the joint probabilities of various possible associations
  - ➢ Two assumptions:
    - a measurement can only have one source
    - no more than one measurement can originate from a target

- Robust to measurement noise, considers uncertainty in associations
- Chosen for scenarios with non-Gaussian measurement distributions, clutter, and situations requiring a more advanced treatment of data association

Can struggle in scenarios with significant clutter

DATA ASSOCIATION: PDAF

- ❑ General idea: instead of matching a single best observation to the track, it updates the track with a weighted average of all validated measurements
- ❑ Integrate **all** measurements in the validation gate
  - ➢ Calculate hypothesis pairs for every measurement for every track
  - ➢ Also consider the additional possibility that no observations match (missed detection)



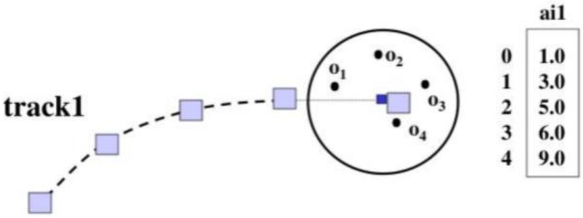
	ai1
0	1.0
1	3.0
2	5.0
3	6.0
4	9.0

- ❑ Weights are the individual association probabilities

DATA ASSOCIATION: PDAF

- The best matching "observation" is computed as a weighted combination of the predicted location and all candidate observation

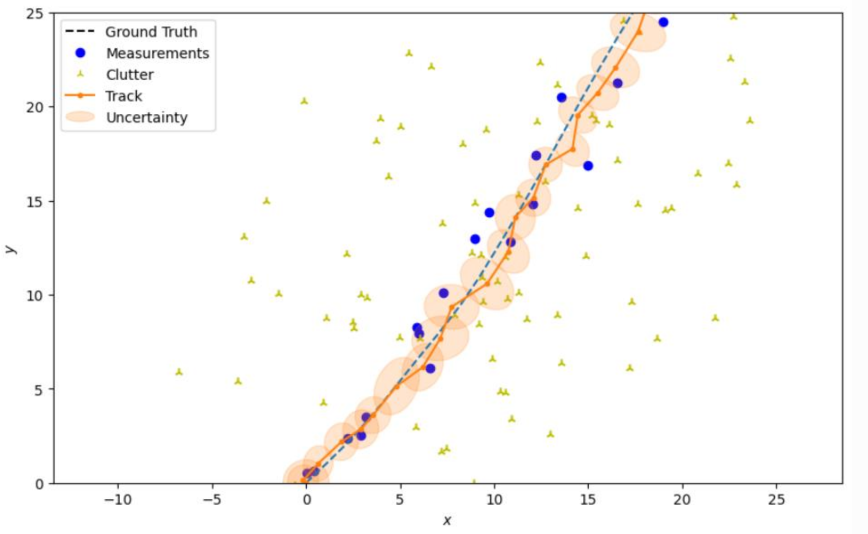
New location = 1/24 * predicted location +
3/24 * o1 + 5/24 * o2 + 6/24 * o3 + 9/24 * o4



	ai1
0	1.0
1	3.0
2	5.0
3	6.0
4	9.0

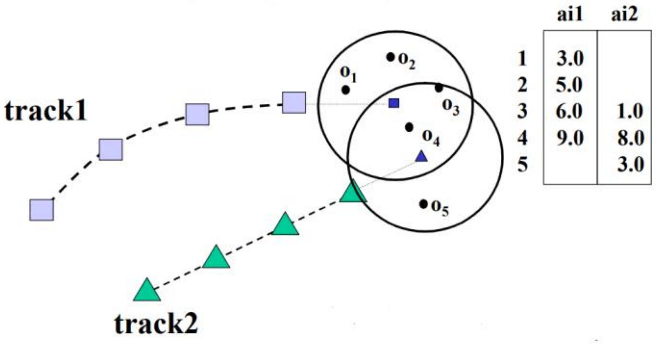
- Recursive algorithm computationally similar to the Kalman Filter (KF). *If the state or measurement equations are nonlinear, then PDAF is based on EKF*
  - State prediction, State covariance prediction, Innovation prediction, Kalman gain are the same as in the standard KF
  - only difference is in the use of the combined innovation (Gaussian mixture) in the state update, and the increased covariance of the update state

DATA ASSOCIATION: PDAF EXEMPLE



DATA ASSOCIATION: PDAF -> JPDA

- When gating regions overlap, the same observations can contribute to updating both trajectories
- The shared observations introduce a coupling into the decision process
  - Doing PDAF on each one independently is no optimal, since observations in overlapping gate regions will be counted more than once (contribute to more than one track)
- Solution: **JPDA** = Joint Probabilistic Data Association
  - Extension of the PDAF



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

DATA ASSOCIATION: JPDA

- The **measurement-to-target association probabilities** are computed jointly across the targets

$$\begin{aligned}
p(A \to x) = \bar{p}(A \to x \cap B \to None \cap C \to None)+ \\
+\bar{p}(A \to x \cap B \to None \cap C \to y)+ \\
+\bar{p}(A \to x \cap B \to None \cap C \to z)+ \\
+\bar{p}(A \to x \cap B \to y \cap C \to None)+ \\
+\bar{p}(A \to x \cap B \to y \cap C \to z)+ \\
+\bar{p}(A \to x \cap B \to z \cap C \to None)+ \\
+\bar{p}(A \to x \cap B \to z \cap C \to y)
\end{aligned}$$

$p(A \to x) = probability\ of\ track\ A\ being\ assosiated\ with\ measurement\ x$

$A, B, C = tracks$ $x, y, z = measurement$ $None = missed\ detection$

$\bar{p}(multi - hypothesis) = normalized\ probability\ of\ the\ multi - hypothesis$

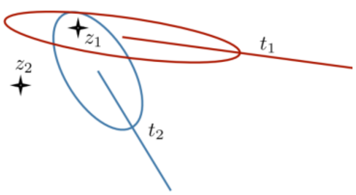
- Each possible (non-conflicting) assignment becomes a hypothesis with an associated probability
- The state estimation is done
  - **Decoupled** : separately for each target as in PDAF, resulting in JPDAF
  - **Coupled** : using a stacked state vector, resulting in JPDACF

DATA ASSOCIATION: MHT

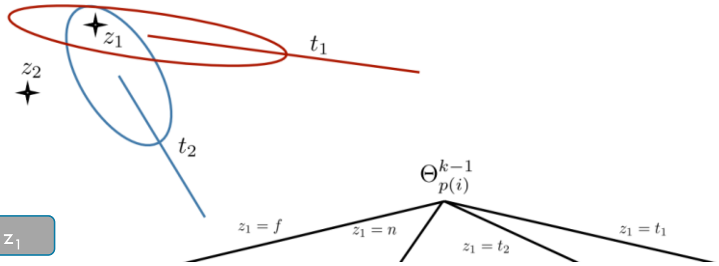
- Reason about the associations of sequences measurements with tracks and false alarm
- Evaluate the probability of association hypotheses
- Optimal Bayesian solution
- Algorithm steps:
  - State and measurement prediction
  - Hypotheses generation
  - Hypotheses probability evaluation
  - State update
  - Hypotheses management (i.e. elimination, creation)
- Computionnally expensive with expotentionally growning number of hypothesis
  - Pruning strategies, K-best hypoyhesis

DATA ASSOCIATION: MHT

- A **<u>hypothesis</u>** $\theta_i^k = \{\theta_{p(i)}^{k-1}, \theta_{c(i)}(k)\}$ at time k = history of assignments sets $\theta = \{\theta_{obs}, \theta_{track}\}$ to time k
  - $\theta_{obs} = \{z_1, \dots, z_{m(k)}\}$ set of measurement associations
  - $\theta_{track} = \{l_1, \dots, l_{t(k)}\}$ set of track labels
- Hypotheses are generated recursively in a tree-based structure
  - Unlikely branch are avoided by validation gating
  - Exponential growth of the trees
  - Only a subset of hypotheses are generated in practice
- The probability of an hypothesis can be calculated using Bayes ru



DATA ASSOCIATION: MHT HYPOTHESIS GENERATION



Considering z_1

$\theta_{obs} = \{z_1, \dots, z_{m(k)}\}$ set of measurement associations, where a measurement is either associated to track $z_i = t$, treated as a new track $z_i = n$, or a false alarm $z_i = f$

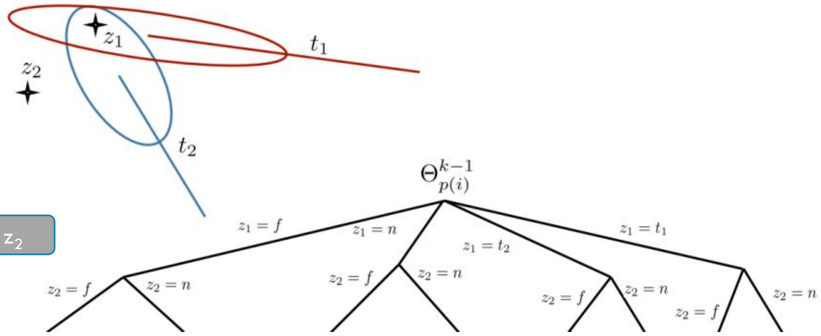
$\theta_{track} = \{l_1, \dots, l_{t(k)}\}$ set of track labels, where a track can be

matched $l_i = m$

occluded $l_i = o$

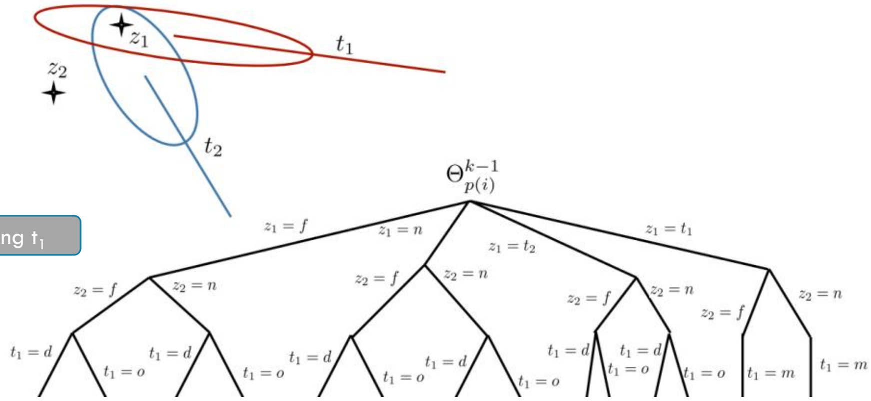
or deleted $l_i = d$

DATA ASSOCIATION: MHT HYPOTHESIS GENERATION



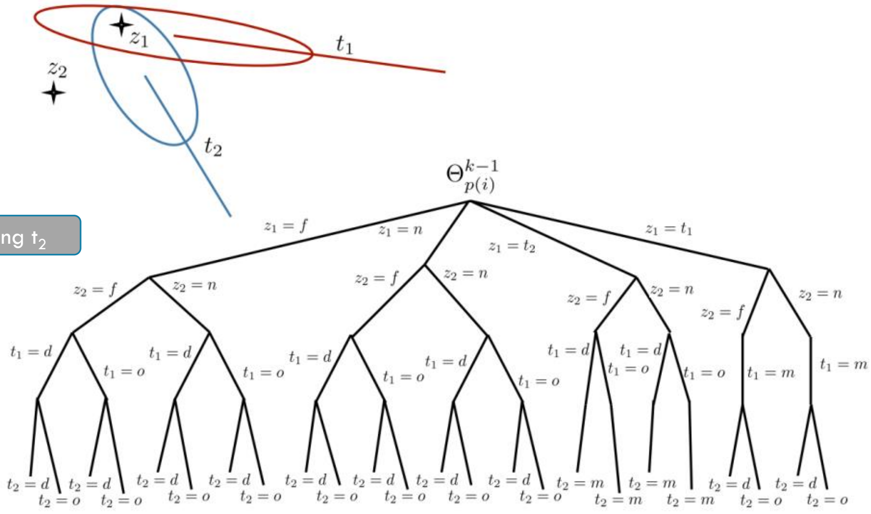
Considering z_2

DATA ASSOCIATION: MHT HYPOTHESIS GENERATION



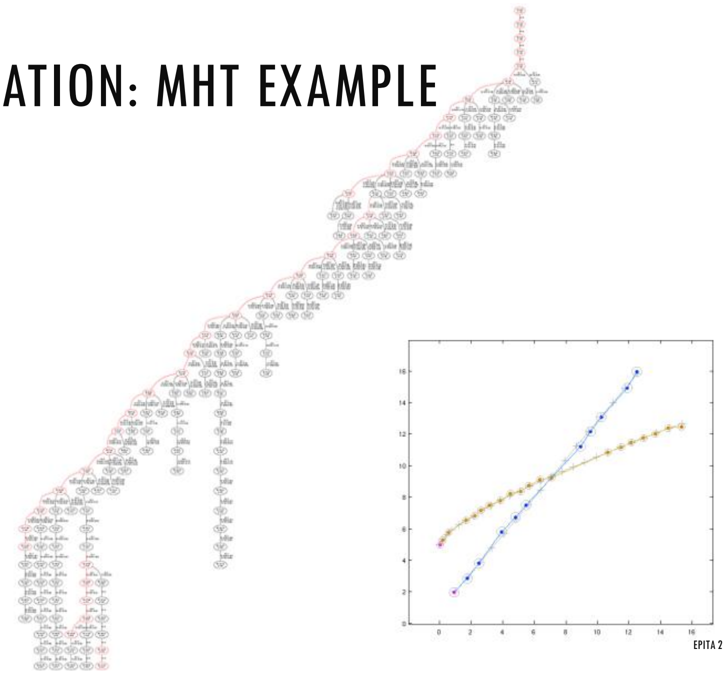
Considering t_1

DATA ASSOCIATION: MHT HYPOTHESIS GENERATION



Considering t_2

DATA ASSOCIATION: MHT EXAMPLE



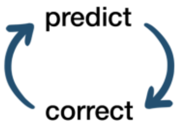
DATA ASSOCIATION: KALMAN FILTER (KF)

❑ Used to estimate the position of linear system by assuming that the errors are Gaussian

❑ Basic idea: Using the prior knowledge of the state, the filter makes a forward state of predicts the next state

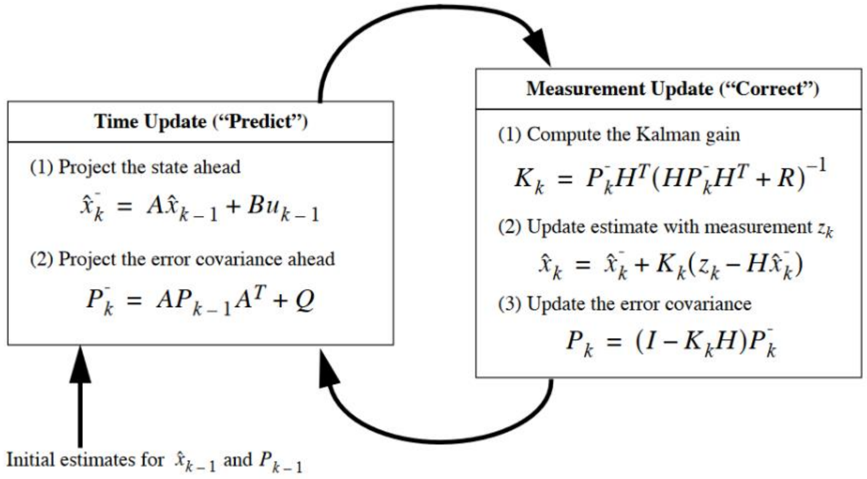
❑An iterative process that continuously refines its state estimate based on new measurements and predictions

❑ Two-step process for estimating state
- ➢ Prediction (using the model)
- ➢ Correction (using the measurements)



❑ For non–linear dynamic models:
- ➢ EKF (Extended Kalman Filter)
- ➢ UKF (Unscented Kalman Filter)

DATA ASSOCIATION: KALMAN FILTER (KF)



Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

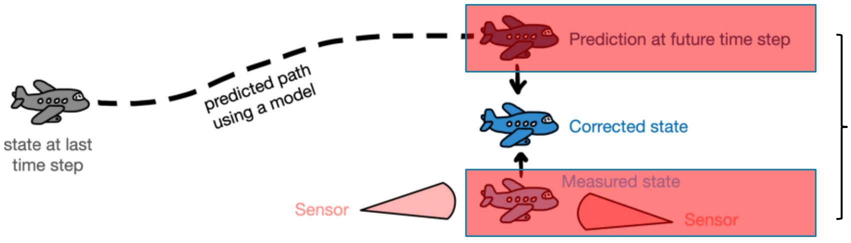
(3) Update the error covariance

$$P_k = (I - K_k H)P_k^-$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

DATA ASSOCIATION: KALMAN FILTER (KF)

- ❑ Predict step
  - ➢ Responsible for propagating the state vector into the future using the motion model (linear or non-linear)
- ❑ Correct/Update step
  - ➢ Blends the current prediction with a current measurement to get the corrected estimated state

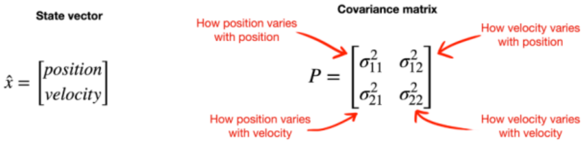


Two estimates of system state which are likely different !

How can we combine the two based on their relative uncertainties?

DATA ASSOCIATION: KALMAN FILTER (KF)

- ❑ Recap: A Kalman filter combines a noisy measurement with a flawed prediction to create an optimal state estimate
- ❑ We can use covariance as a measure of how much uncertainty there is in the noisy measurement, the flawed model, and the final estimation
- ❑ Since covariance is multi-dimensional, it's mathematically easier to maintain it in matrix form



State vector

$$\hat{x} = \begin{bmatrix} position \\ velocity \end{bmatrix}$$

Covariance matrix

$$P = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 \end{bmatrix}$$

How position varies with position

How velocity varies with position

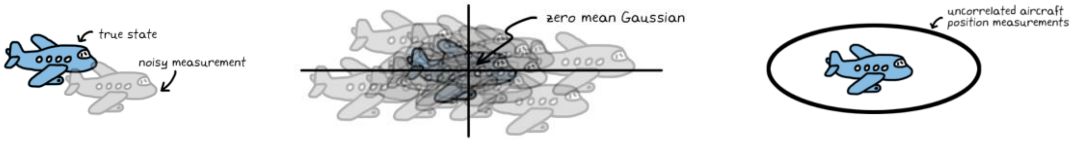
How position varies with velocity

How velocity varies with velocity

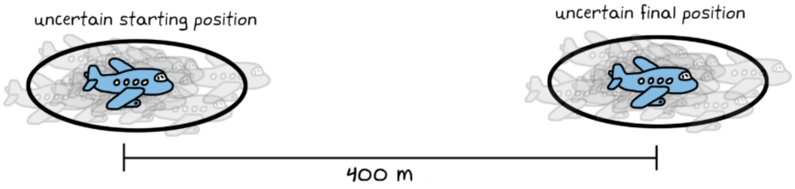
- ❑ A Kalman filter uses three different covariance matrices (measurement, model, final estimation) in order to maintain an estimate of the system state

DATA ASSOCIATION: KALMAN FILTER (KF)

- ❑ THE SENSORS AREN'T PERFECT => **Measurement noise covariance matrix** (R)
  - ➢ Capture the expected uncertainty that you have with the sensor measurements

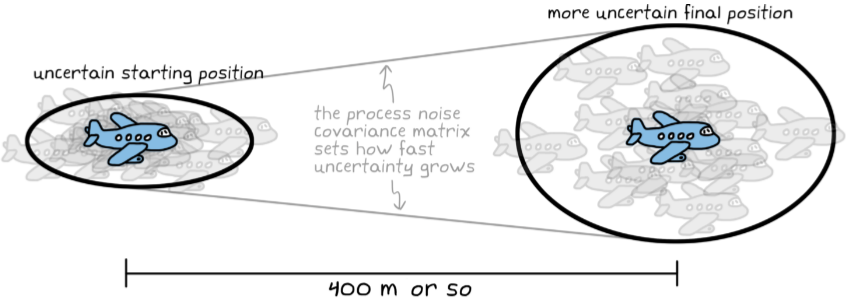


- ❑ THE PROCESS MODEL ISN'T PERFECT
  - ➢ The initial state before we even start the prediction process has error => **Prediction error covariance matrix** (P)



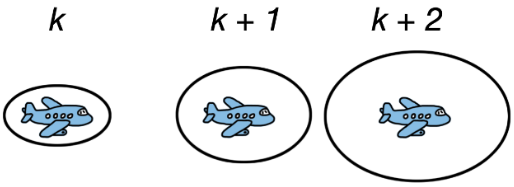
DATA ASSOCIATION: KALMAN FILTER (KF)

- THE PROCESS MODEL ISN'T PERFECT
  - The initial state before we even start the prediction process has error => **Prediction error covariance matrix** (P)
  - The model isn't perfect! The act of predicting into the future causes additional uncertainty => **Process noise covariance matrix** (Q)


more uncertain final position
uncertain starting position
the process noise covariance matrix sets how fast uncertainty grows
400 m or so

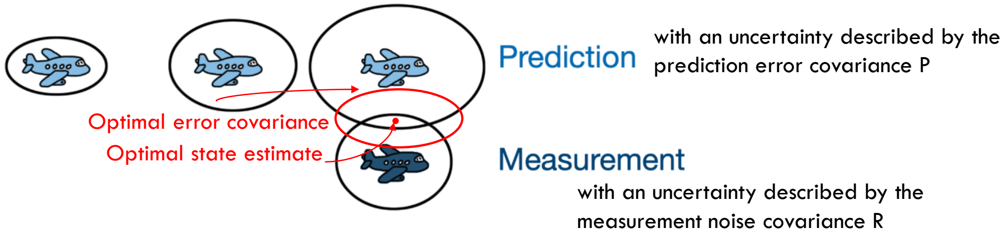
DATA ASSOCIATION: KALMAN FILTER (KF)

- HOW DO WE MANAGE ALL OF THIS?
  - Time step k: Start with an initial state and its associated error covariance (**P**)
  - Time step k+1: Propagate the state and error covariance into the future with a model of the system => The error covariance grows based on the specified process noise covariance (**Q**)
  - Time step k + i : Continue propagating the prediction each time step



DATA ASSOCIATION: KALMAN FILTER (KF)

- When a noisy measurement is available …


Prediction
with an uncertainty described by the prediction error covariance P
Optimal error covariance
Optimal state estimate
Measurement
with an uncertainty described by the measurement noise covariance R

- **Optimal corrected state:** Multiply the two Gaussian distributions together!
- Once a Gaussian, always a Gaussian (if linear)
  - Gaussian probability distribution maintains its Gaussian shape when subjected to linear operations

DATA ASSOCIATION: KALMAN FILTER (KF)

A FAST OVERVIEW OF MULTIPLYING GAUSSIANS

prediction **measurement** **new Gaussian**

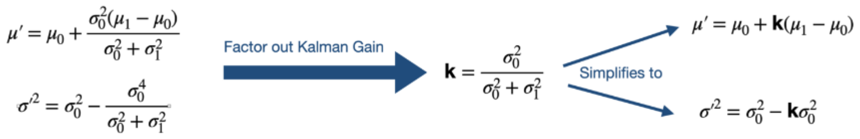
$$\left(\frac{1}{\sigma_0\sqrt{2\pi}}e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}}\right)\cdot\left(\frac{1}{\sigma_1\sqrt{2\pi}}e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\right)=\left(\frac{1}{\sigma'\sqrt{2\pi}}e^{-\frac{(x-\mu')^2}{2\sigma'^2}}\right)$$

The question: What does the new mean, **μ'**, and new variance, **σ'^2**, look like for the resulting distribution?

$$\mu' = \mu_0 + \frac{\sigma_0^2(\mu_1-\mu_0)}{\sigma_0^2+\sigma_1^2} \qquad \sigma'^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2+\sigma_1^2}$$

DATA ASSOCIATION: KALMAN FILTER (KF)

- Optimal Kalman gain (K)
  - A common multiplier that we can factor out to simplify the equations for the new mean and deviation
  - K value is a scale between 0 and 1
  - K reflects the relative uncertainty in the prediction versus the measurement



$$\mu' = \mu_0 + \frac{\sigma_0^2(\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2}$$

$$\sigma'^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2}$$

Factor out Kalman Gain

$$\mathbf{k} = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2}$$

Simplifies to

$$\mu' = \mu_0 + \mathbf{k}(\mu_1 - \mu_0)$$

$$\sigma'^2 = \sigma_0^2 - \mathbf{k}\sigma_0^2$$

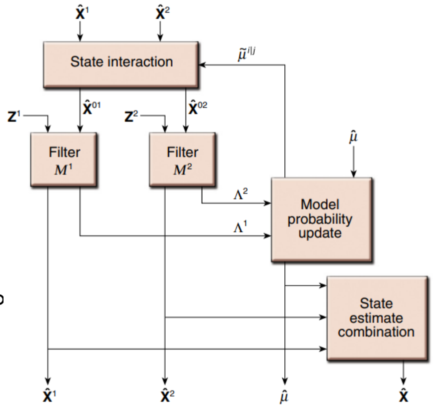
WE'RE READY FOR THE FILTER EQUATIONS !

DATA ASSOCIATION: PARTICLE FILTER (PF)

- Particle Filter = Sequential Monte Carlo
  - Recursive Monte Carlo statistical calculation method
- Principe:
  - Representing the probability distribution of an object's state using a set of particles, and then updating and refining this distribution over time based on observed measurements
- Well-suited for tracking scenarios where the underlying motion model or measurement model is non-linear and/or non-Gaussian
  - Preferred for systems where the dynamics are complex or poorly known!
- PF can handle more complex and arbitrary relationships between state variables and measurements

Increased complexity due to multiple models

DATA ASSOCIATION: IMM



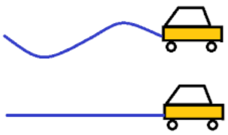
- ❑ Interactive Multiple Models (IMM)
  - ➢ Provide a structure to efficiently manage multiple filter models
  - ➢ Combine multiple motion models to adapt to changes in an object's motio
    - ▪ Deal with the multiple motion models in the Bayesian framework in the Bayesian framework
  - ➢ It is not a single motion model but a framework that switches between multiple motion models based on the object's behavior
  - ➢ **Principe**: use set of M elementar filters at each time, each filter corresponds to a specific motion model. Final state is obtained by merging the results of all filters according to the distribution probability
  - ➢ Useful when an object's motion behavior change over time or when object dynamics are not well-defined

Adaptable to changing dynamics, effective in handling model uncertainties

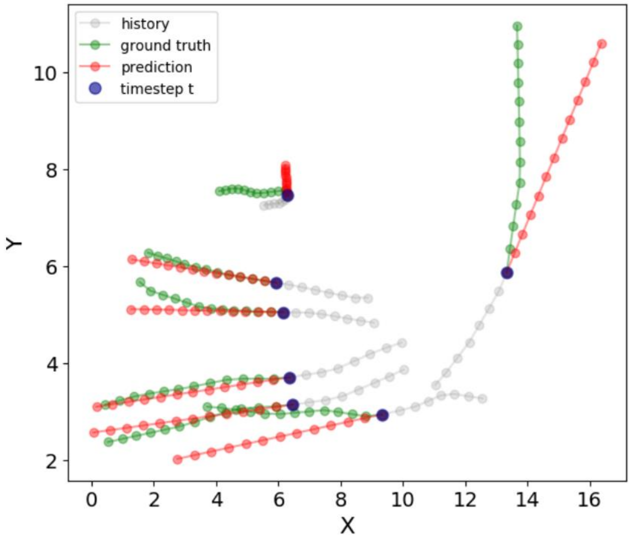
BUT It increases of computational effort and complexity of algorithme due to multiple models!

MOTION MODEL

- ❑ Mathematical representation of how an object's state or position changes over time
- ❑ Capture the dynamic behavior of an object
- ❑ Estimate potential position of objects in the futur frame, thereby reducing the search space
  - ➢ It plays a crucial role in predicting the future location of the object between measurements and in making informed decisions about where the object is likely to be
- ❑ The choice of motion model depends on the characteristics of the tracked object and the specific tracking scenario



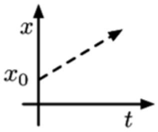
MOTION MODEL



MOTION MODEL

- ❑ LINEAR MOTION MODEL: Commonly used to explain the object's dynamics!
  - ➢ **Constant Velocity Model (CV)**
    - Assume that the object moves with a constant velocity in a straight line
    - The state includes position (x, y) and velocity (vx, vy)
    - This model is suitable for tracking objects with relatively smooth and linear motion

$$x = x_0 + v_0 t$$



State transition matrix

$$F_{CV} = \begin{bmatrix} 1 & dT & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dT \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cycle time

Process noise covariance matrix

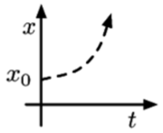
$$Q_{CV} = \begin{bmatrix} \alpha \frac{dT^4}{4} & \alpha \frac{dT^3}{2} & 0 & 0 \\ \alpha \frac{dT^3}{2} & \alpha dT^2 & 0 & 0 \\ 0 & 0 & \beta \frac{dT^4}{4} & \beta \frac{dT^3}{2} \\ 0 & 0 & \beta \frac{dT^3}{2} & \beta dT^2 \end{bmatrix}$$

Spectral density of process noise

MOTION MODEL

- ❑ LINEAR MOTION MODEL
  - ➢ Constant Velocity Model (CV)
  - ➢ **Constant Acceleration Model**
    - Consider that the object's velocity changes linearly over time due to constant acceleration
    - The state includes position (x, y), velocity (vx, vy), and acceleration (ax, ay)
    - It is useful for tracking objects with more complex motion, such as vehicles

$$x = x_0 + vt + \frac{a_0 t^2}{2}$$



- ➢ **Constant Turn (CT) Model**
  - Motion of a vehicle can usually be modeled as moving by circle segments

- ❑ NON-LINEAR MOTION MODEL
  - ➢ Capture complex object motion accurately

INTERACTION MODEL

- ❑ Interation model = Mutal motion model
  - ➢ Goal: Capture the influence of an object on other objects
  - ➢ In the crowd scenery, an object would experience some “force” from other agents and objects
- ❑ Types:
  - ➢ Social force models (SfM)
  - ➢ Crowd motion pattern models

INTERACTION MODEL: SFM

Target behavior is modeled based on:

- ❑ **Individual force** (conidered for each individual in a group)
  - ➢ Fidelity: one should not change his desired destination
  - ➢ Constancy: one should not suddenly change his momentum, including speed and direction
- ❑ **Group force** (whole group)
  - ➢ Attraction : individuals moving together as a group should stay close
  - ➢ Repulsion: individuals moving together as a group should keep some distance away from others to make all members comfortable
  - ➢ Coherence: individuals moving together as a group should move with similar velocity

MOT EVALUATION

- ❑ Output of the MOT:
  - ➢ what objects are present in each frame (detection)
  - ➢ where they are in each frame (localisation)
  - ➢ and whether objects in different frames belong to the same or different objects (association) => UNIQUE ID
- ❑ Type of tracking errors (the most commonly used error types):
  - ➢ Detection error
    - Tracker predicts detections that don’t exist in the ground-truth, or fails to predict detections that are in the ground-truth
  - ➢ Localisation error
    - It occurs when predited detection (prDets) are not perfectly spatially aligned with ground truth detection (gtDets)
  - ➢ Association error
    - Tracker assigns the same ID to two detections which have different ground truth IDs (gtIDs), or when it assigns different IDs to two detections which should have the same ground truth ID (gtID)

MOT EVALUATION

- ❑ How to assess the tracker's performance ?
  - ➢ Compare its output to a ground-truth set of tracking result
  - ➢ Evaluate the similarity between the given ground-truth and the tracking results
  - ➢ Problem not well defined = > There are many different ways of scoring such a similarity
  - ➢ The choice of evaluation metric is extremely important, as the properties of the metric determine how different errors contribute to a final score

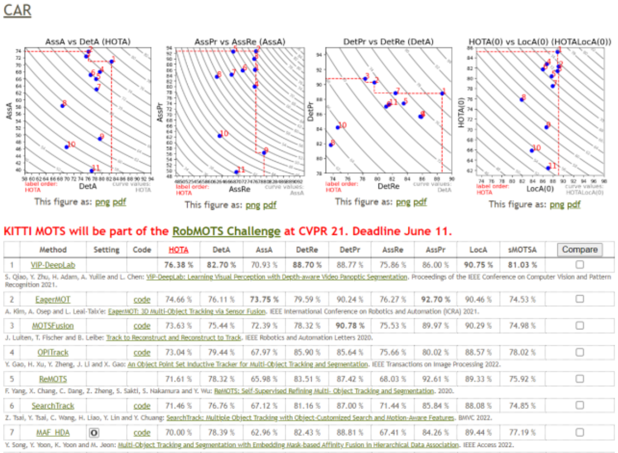
BUT :

- ❑ Different Detection Representations
  - ➢2D bounding boxes, 3D bounding boxes, segmentation masks, point estimates in 2D or 3D, and human pose skeletons ….
  - ➢ In general tracking evaluation metrics are specific representation -aware

MOT EVALUATION

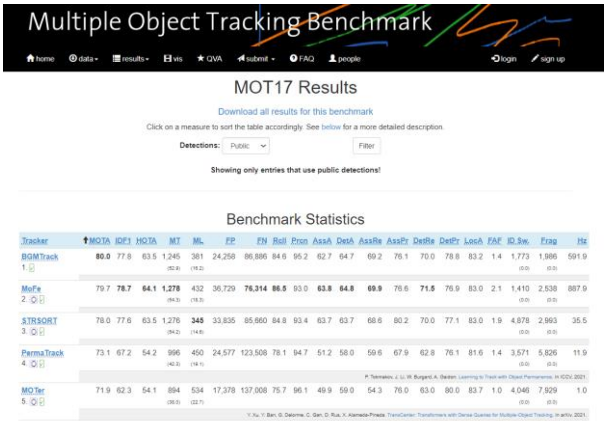
- ❑ Use Standard Datasets
  - ➢ Evaluate trackers on well-established benchmark datasets like ke OTB (Object Tracking Benchmark), VOT (Visual Object Tracking), and MOT (Multiple Object Tracking), KITTI
- ❑ Ground Truth Annotations
  - ➢ Reliable ground truth annotations are essential for accurate evaluation. Ensure that ground truth data aligns with the tracking objectives
- ❑ Tracking Challenges
  - ➢ RobMOT
  - ➢ MOT Challenge
  - ➢…

MOT EVALUATION: TRACKING CHALLENGE



KITTI MOTS will be part of the RobMOTS Challenge at CVPR 21. Deadline June 11.

	Method	Setting	Code	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA	sMOTSA	Compare
1	VIP-DeepLab			**76.38 %**	**82.70 %**	70.93 %	**88.70 %**	88.77 %	75.86 %	86.00 %	**90.75 %**	**81.03 %**	□
S. Qiao, Y. Zhu, H. Adam, A. Yuille and L. Chen: VIP-DeepLab: Learning Visual Perception with Depth-aware Video Panoptic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2021.													
2	EagerMOT		code	74.66 %	76.11 %	**73.75 %**	79.59 %	90.24 %	76.27 %	**92.70 %**	90.46 %	74.53 %	□
A. Kim, A. Osep and L. Leal-Taix'e: EagerMOT: 3D Multi-Object Tracking via Sensor Fusion. IEEE International Conference on Robotics and Automation (ICRA) 2021.													
3	MOTSFusion		code	73.63 %	75.44 %	72.39 %	78.32 %	**90.78 %**	75.53 %	89.97 %	90.29 %	74.98 %	□
J. Luiten, T. Fischer and B. Leibe: Track to Reconstruct and Reconstruct to Track. IEEE Robotics and Automation Letters 2020.													
4	OPITrack		code	73.04 %	79.44 %	67.97 %	85.90 %	85.64 %	75.66 %	80.02 %	88.57 %	78.02 %	□
Y. Gao, H. Xu, Y. Zheng, J. Li and X. Gao: An Object Point Set Inductive Tracker for Multi-Object Tracking and Segmentation. IEEE Transactions on Image Processing 2022.													
5	ReMOTS			71.61 %	78.32 %	65.98 %	83.51 %	87.42 %	68.03 %	92.61 %	89.33 %	75.92 %	□
F. Yang, X. Chang, C. Dang, Z. Zheng, S. Sakti, S. Nakamura and Y. Wu: ReMOTS: Self-Supervised Refining Multi- Object Tracking and Segmentation. 2020.													
6	SearchTrack		code	71.46 %	76.76 %	67.12 %	81.16 %	87.00 %	71.44 %	85.84 %	88.08 %	74.85 %	□
Z. Tsai, Y. Tsai, C. Wang, H. Liao, Y. Lin and Y. Chuang: SearchTrack: Multiple Object Tracking with Object-Customized Search and Motion-Aware Features. BMVC 2022.													
7	MAF_HDA	O	code	70.00 %	78.39 %	62.96 %	82.43 %	88.81 %	67.41 %	84.26 %	89.44 %	77.19 %	□
Y. Song, Y. Yoon, K. Yoon and M. Jeon: Multi-Object Tracking and Segmentation with Embedding Mask-based Affinity Fusion in Hierarchical Data Association. IEEE Access 2022.													



MOT17 Results

Download all results for this benchmark

Click on a measure to sort the table accordingly. See below for a more detailed description.

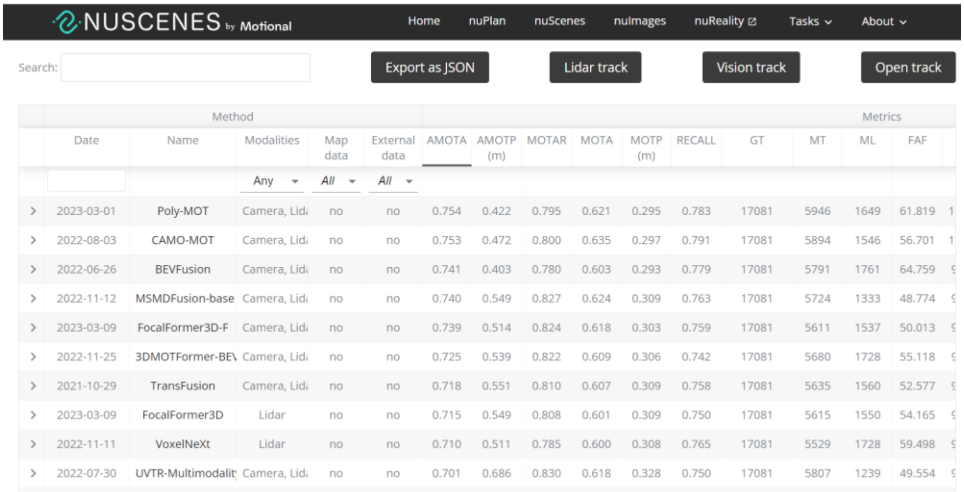
Detections: Public   Filter

Showing only entries that use public detections!

Benchmark Statistics

Tracker	MOTA	IDF1	HOTA	MT	ML	FP	FN	Rcll	Prcn	AssA	DetA	AssRe	AssPr	DetRe	DetPr	LocA	FAF	ID Sw.	Frag	Hz
BGMTrack 1.	**80.0**	77.8	63.5	1,245 (52.9)	381 (16.2)	24,258	86,886	84.6	95.2	62.7	64.7	69.2	76.1	70.0	78.8	83.2	1.4	1,773 (0.0)	1,986 (0.0)	591.9
MoFe 2.	79.7	**78.7**	**64.1**	**1,278** (54.3)	432 (18.3)	36,729	**76,314**	**86.5**	93.0	**63.8**	**64.8**	**69.9**	76.6	**71.5**	76.9	83.0	2.1	1,410 (0.0)	2,538 (0.0)	887.9
STRSORT 3.	78.0	77.6	63.5	1,276 (54.2)	**345** (14.6)	33,835	85,660	84.8	93.4	63.7	63.7	68.6	80.2	70.0	77.1	83.0	1.9	4,878 (0.0)	2,993 (0.0)	35.5
PermaTrack 4.	73.1	67.2	54.2	996 (42.3)	450 (19.1)	24,577	123,508	78.1	94.7	51.2	58.0	59.6	67.9	62.8	76.1	81.6	1.4	3,571 (0.0)	5,826 (0.0)	11.9
P. Tokmakov, J. Li, W. Burgard, A. Gaidon: Learning to Track with Object Permanence. In ICCV, 2021.																				
MOTer 5.	71.9	62.3	54.1	894 (38.0)	534 (22.7)	17,378	137,008	75.7	96.1	49.9	59.0	54.3	76.0	63.0	80.0	83.7	1.0	4,046 (0.0)	7,929 (0.0)	1.0
Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, X. Alameda-Pineda: TransCenter: Transformers with Dense Queries for Multiple-Object Tracking. In arXiv, 2021.																				

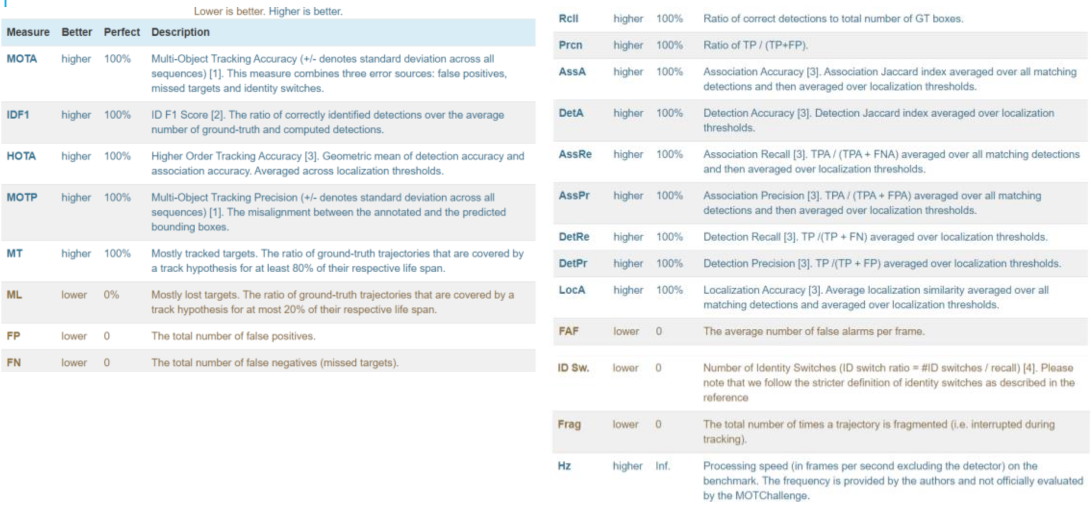
MOT EVALUATION: TRACKING CHALLENGE



Search:

	Method					Metrics										
	Date	Name	Modalities	Map data	External data	AMOTA	AMOTP (m)	MOTAR	MOTA	MOTP (m)	RECALL	GT	MT	ML	FAF	
			Any	All	All											
›	2023-03-01	Poly-MOT	Camera, Lidi	no	no	0.754	0.422	0.795	0.621	0.295	0.783	17081	5946	1649	61.819	1
›	2022-08-03	CAMO-MOT	Camera, Lidi	no	no	0.753	0.472	0.800	0.635	0.297	0.791	17081	5894	1546	56.701	1
›	2022-06-26	BEVFusion	Camera, Lidi	no	no	0.741	0.403	0.780	0.603	0.293	0.779	17081	5791	1761	64.759	9
›	2022-11-12	MSMDFusion-base	Camera, Lidi	no	no	0.740	0.549	0.827	0.624	0.309	0.763	17081	5724	1333	48.774	9
›	2023-03-09	FocalFormer3D-F	Camera, Lidi	no	no	0.739	0.514	0.824	0.618	0.303	0.759	17081	5611	1537	50.013	9
›	2022-11-25	3DMOTFormer-BE\	Camera, Lidi	no	no	0.725	0.539	0.822	0.609	0.306	0.742	17081	5680	1728	55.118	9
›	2021-10-29	TransFusion	Camera, Lidi	no	no	0.718	0.551	0.810	0.607	0.309	0.758	17081	5635	1560	52.577	9
›	2023-03-09	FocalFormer3D	Lidar	no	no	0.715	0.549	0.808	0.601	0.309	0.750	17081	5615	1550	54.165	9
›	2022-11-11	VoxelNeXt	Lidar	no	no	0.710	0.511	0.785	0.600	0.308	0.765	17081	5529	1728	59.498	9
›	2022-07-30	UVTR-Multimodalit	Camera, Lidi	no	no	0.701	0.686	0.830	0.618	0.328	0.750	17081	5807	1239	49.554	9

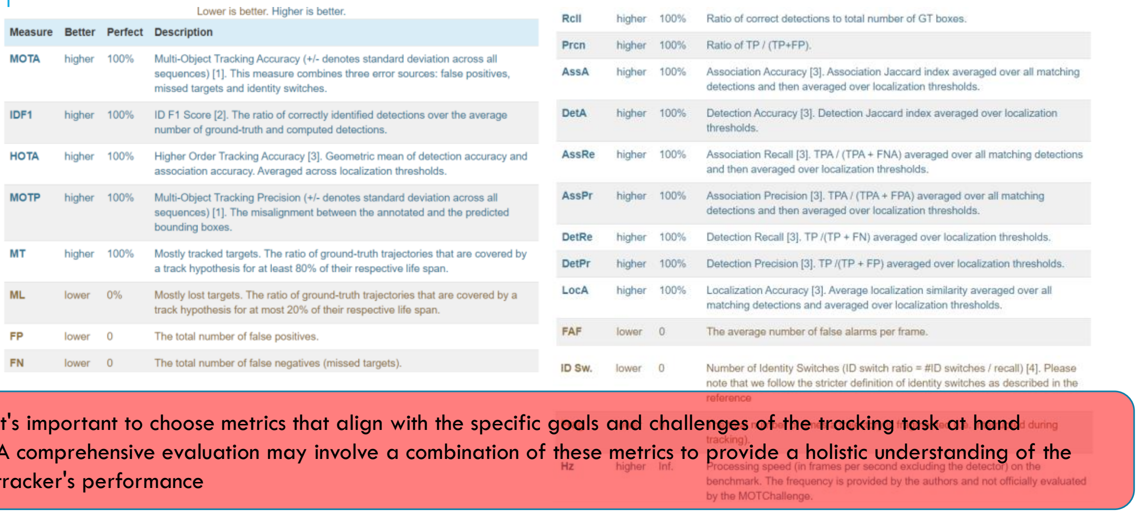
MOT EVALUATION: METRICS



Lower is better. Higher is better.

Measure	Better	Perfect	Description
MOTA	higher	100%	Multi-Object Tracking Accuracy (+/- denotes standard deviation across all sequences) [1]. This measure combines three error sources: false positives, missed targets and identity switches.
IDF1	higher	100%	ID F1 Score [2]. The ratio of correctly identified detections over the average number of ground-truth and computed detections.
HOTA	higher	100%	Higher Order Tracking Accuracy [3]. Geometric mean of detection accuracy and association accuracy. Averaged across localization thresholds.
MOTP	higher	100%	Multi-Object Tracking Precision (+/- denotes standard deviation across all sequences) [1]. The misalignment between the annotated and the predicted bounding boxes.
MT	higher	100%	Mostly tracked targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.
ML	lower	0%	Mostly lost targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.
FP	lower	0	The total number of false positives.
FN	lower	0	The total number of false negatives (missed targets).
Rcll	higher	100%	Ratio of correct detections to total number of GT boxes.
Prcn	higher	100%	Ratio of TP / (TP+FP).
AssA	higher	100%	Association Accuracy [3]. Association Jaccard index averaged over all matching detections and then averaged over localization thresholds.
DetA	higher	100%	Detection Accuracy [3]. Detection Jaccard index averaged over localization thresholds.
AssRe	higher	100%	Association Recall [3]. TPA / (TPA + FNA) averaged over all matching detections and then averaged over localization thresholds.
AssPr	higher	100%	Association Precision [3]. TPA / (TPA + FPA) averaged over all matching detections and then averaged over localization thresholds.
DetRe	higher	100%	Detection Recall [3]. TP /(TP + FN) averaged over localization thresholds.
DetPr	higher	100%	Detection Precision [3]. TP /(TP + FP) averaged over localization thresholds.
LocA	higher	100%	Localization Accuracy [3]. Average localization similarity averaged over all matching detections and averaged over localization thresholds.
FAF	lower	0	The average number of false alarms per frame.
ID Sw.	lower	0	Number of Identity Switches (ID switch ratio = #ID switches / recall) [4]. Please note that we follow the stricter definition of identity switches as described in the reference
Frag	lower	0	The total number of times a trajectory is fragmented (i.e. interrupted during tracking).
Hz	higher	Inf.	Processing speed (in frames per second excluding the detector) on the benchmark. The frequency is provided by the authors and not officially evaluated by the MOTChallenge.

MOT EVALUATION: METRICS



Lower is better. Higher is better.

Measure	Better	Perfect	Description
MOTA	higher	100%	Multi-Object Tracking Accuracy (+/- denotes standard deviation across all sequences) [1]. This measure combines three error sources: false positives, missed targets and identity switches.
IDF1	higher	100%	ID F1 Score [2]. The ratio of correctly identified detections over the average number of ground-truth and computed detections.
HOTA	higher	100%	Higher Order Tracking Accuracy [3]. Geometric mean of detection accuracy and association accuracy. Averaged across localization thresholds.
MOTP	higher	100%	Multi-Object Tracking Precision (+/- denotes standard deviation across all sequences) [1]. The misalignment between the annotated and the predicted bounding boxes.
MT	higher	100%	Mostly tracked targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.
ML	lower	0%	Mostly lost targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.
FP	lower	0	The total number of false positives.
FN	lower	0	The total number of false negatives (missed targets).
Rcll	higher	100%	Ratio of correct detections to total number of GT boxes.
Prcn	higher	100%	Ratio of TP / (TP+FP).
AssA	higher	100%	Association Accuracy [3]. Association Jaccard index averaged over all matching detections and then averaged over localization thresholds.
DetA	higher	100%	Detection Accuracy [3]. Detection Jaccard index averaged over localization thresholds.
AssRe	higher	100%	Association Recall [3]. TPA / (TPA + FNA) averaged over all matching detections and then averaged over localization thresholds.
AssPr	higher	100%	Association Precision [3]. TPA / (TPA + FPA) averaged over all matching detections and then averaged over localization thresholds.
DetRe	higher	100%	Detection Recall [3]. TP /(TP + FN) averaged over localization thresholds.
DetPr	higher	100%	Detection Precision [3]. TP /(TP + FP) averaged over localization thresholds.
LocA	higher	100%	Localization Accuracy [3]. Average localization similarity averaged over all matching detections and averaged over localization thresholds.
FAF	lower	0	The average number of false alarms per frame.
ID Sw.	lower	0	Number of Identity Switches (ID switch ratio = #ID switches / recall) [4]. Please note that we follow the stricter definition of identity switches as described in the reference
Frag	lower	0	The total number of times a trajectory is fragmented (i.e. interrupted during tracking).
Hz	higher	Inf.	Processing speed (in frames per second excluding the detector) on the benchmark. The frequency is provided by the authors and not officially evaluated by the MOTChallenge.

- It's important to choose metrics that align with the specific goals and challenges of the tracking task at hand
- A comprehensive evaluation may involve a combination of these metrics to provide a holistic understanding of the tracker's performance

MOT EVALUATION

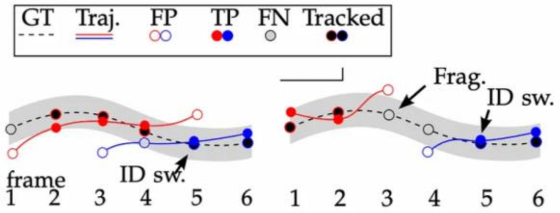
- Most metrics require a definition of a <u>similarity score</u> (S) between two detections
  - Constrained to be between 0 (there is no overlap between detections) and 1(perfectly align)
  - The most commonly used similarity metric
    - for 2D boxes, 3D boxes and segmentation masks: **IoU/ Jaccard Index**
    - for point representations: **a score of one minus the Euclidean distance**

$$\text{Jaccard Index} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}| + |\text{FP}|}$$

- Bijective (one-to-one) Matching computed between gtDets and prDets
  - Calculate a matching score between all pairs of gtDet and prDet
  - Use the Hungarian algorithm for finding the matching that optimizes the sum of the matching score
  - Outputs:
    - **True Positive TP** (gtDets and prDets that are matched together) : correct predictions
    - **False Negative FN:** gtDets that are not matched = *ground truth exists but prediction was missed*
    - **False Positive FP**: prDets that are not matched (extra predictions) = *tracker prediction exists for no ground truth tracker*

MOT EVALUATION: METRICS

- Identity Switch (IDSW)
  - Measures association error
  - IDSW occurs when a tracker wrongfully swaps object identities or when a track was lost and reinitialised with a different identity = two or more objects tracks are swapped as they pass close to each other
  - An IDSW is a TP which has a prID that is different from the prID of the previous TP (that has the same gtID)
- Fragmentation
  - It occurs when a track suddenly stops getting tracked but ground truth track still exists



MOT EVALUATION: METRICS

❑ IDF1 (Identification Metrics)

- Perform at a <u>trajectory level</u>
  - Calculate a bijective mapping between the sets of gtTrajs and prTrajs to determine which trajectories are present
  - Hungarian algorithm selects which trajectories to match to minimize the sum of the number of IDFP and IDFN
  - Matched when S ≥ α of trajectories
- **IDF1 = the ratio of correctly identified detections over the average number of ground-truth and computed detections**
- **IDF1** combines **IDP(ID Precision)** and **IDR(ID Recall)**
- It emphasizes Association accuracy rather than detection

$$\text{ID-Recall} = \frac{|\text{IDTP}|}{|\text{IDTP}| + |\text{IDFN}|}$$

$$\text{ID-Precision} = \frac{|\text{IDTP}|}{|\text{IDTP}| + |\text{IDFP}|}$$

$$\text{IDF1} = \frac{|\text{IDTP}|}{|\text{IDTP}| + 0.5\,|\text{IDFN}| + 0.5\,|\text{IDFP}|}$$

MOT EVALUATION: METRICS

❑ MOTA (Multi-Object Tracking Accuracy)

- Measures three types of tracking errors: the detection errors of FNs and FPs, as well as the association error of Identity Switch (IDSW)

$$MOTA = 1 - \frac{|FN| + |FP| + |IDSW|}{|gtDet|}$$

- Matches are done at a <u>local detection level</u>
  - A bijecting mapping is constructed between prDets(predicted detection)and gtDets(ground truth detection)
  - TP = a pair consisting of a gtDet and a prDet, for which the localisation similarity S is greater than or equal to the threshold α
- MOTA doesn't include a measure of localization error

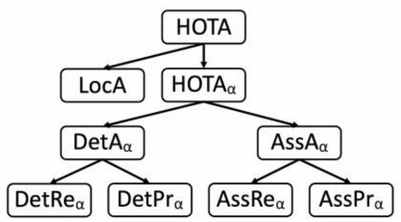
❑ MOTP (Multi-Object Tracking Precision)

- Measure localisation accuracy of the detector
- It provides little information about actual performance of the tracker

$$MOTP = \frac{1}{|TP|} \sum_{TP} S$$

MOT EVALUATION: METRICS

- HOTA (Higher Order Tracking Accuracy)
  - Builded upon the MOTA metric
  - Matching (bijective) occurs at a <u>detection level </u>(similar to MOTA)
  - HOTA decomposes into a family of sub-metrics
    - Localization Accuracy (LocA)
    - Detection Accuracy (DetA)
    - Association Accuracy (AssA)



  - A single unified metric that explicitly evaluates three aspects of tracking : ***accurate detection, association and localization***
  - HOTA is calculated at a number of different localization thresholds α
  - The final HOTA score is the average of the $HOTA_\alpha$ scores calculated at each threshold

$$\mathrm{HOTA} = \int_0^1 \mathrm{HOTA}_\alpha \, d\alpha \approx \frac{1}{19} \sum_{\alpha \in \left\{ \substack{0.05,\ 0.1,\ \ldots \\ 0.9,\ 0.95} \right\}} \mathrm{HOTA}_\alpha$$

MOT EVALUATION: METRICS

- Localisation accuracy score (LocA)

Where S(c) is a spatial similarity score between the prDet and gtDet which make up the TP

$$\mathrm{LocA} = \int_0^1 \frac{1}{|\mathrm{TP}_\alpha|} \sum_{c \in \{\mathrm{TP}_\alpha\}} \mathcal{S}(c) \, d\alpha$$

- Detection Accuracy Score (DetA)

$$\mathrm{DetA}_\alpha = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|}$$

- Association Accuracy Score (AssA)

$$\mathrm{AssA}_\alpha = \frac{1}{|\mathrm{TP}|} \sum_{c \in \{\mathrm{TP}\}} \mathcal{A}(c)$$

MOT EVALUATION: METRICS

- HOTA score $HOTA_{\alpha}$
  - We call this a ‘double Jaccard’ formulation
  - Association score A = another Jaccard metric, but this time over the association concepts of TPAs/FPAs/FNAs

$$\mathcal{A}(c) = \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FNA}(c)| + |\mathrm{FPA}(c)|}$$

$$\begin{aligned} \mathrm{HOTA}_{\alpha} &= \sqrt{\frac{\sum_{c \in \{\mathrm{TP}\}} \mathcal{A}(c)}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|}} \\ &= \sqrt{\mathrm{DetA}_{\alpha} \cdot \mathrm{AssA}_{\alpha}} \end{aligned}$$

- HOTA = geometric mean of a detection score and an association score

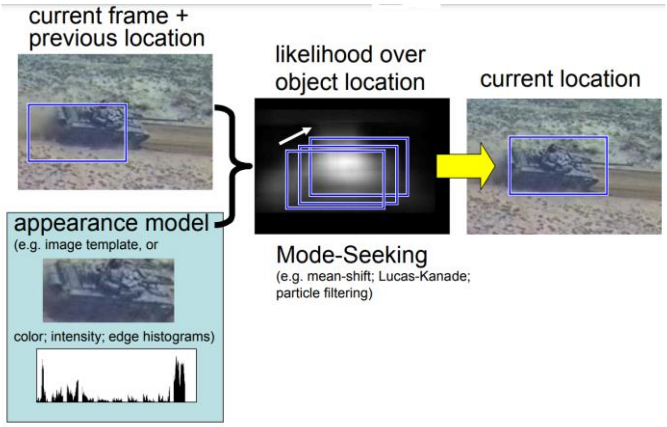
MOT EVALUATION: EFFICIENCY METRICS

- Frames Per Second (FPS)
  - Speed of the tracker evaluated by measuring how many frames it processes per second
- Processing time
  - Elapsed time to process a sequence of frames, indicating the computational efficiency of the tracker

CATEGORIZATION : TRACKING APPROACH

- **Model-Based Tracking**
  - Utilizes a model of the object being tracked. The model is usually constructed from prior knowledge or learning
  - Examples: Kalman filters, Particle filters, Markov Models, …
- **Feature-Based Tracking**
  - Identifies and tracks distinct features or keypoints within the object or its surroundings
  - Examples: Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), ORB, but also CNN-based features (i.e. superpoints), …
- **Appearance-Based Tracking**
  - Focuses on the appearance of the object, often using templates or appearance models
  - Examples:, Template matching, Correlation Filters, Mean-shift, …
- **Deep Learning-Based Tracking**
  - Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Transformers
  - Examples: Siamese Networks, GOTURN, FairMOT…

APPAREANCE-BASED TRACKING



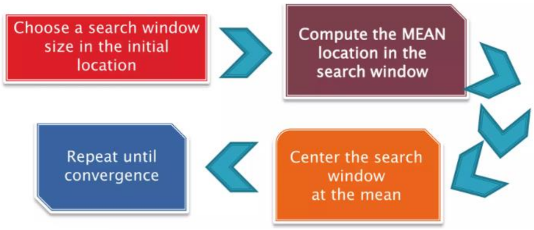
TEMPLATE MATCHING

- Technique for finding small parts of an image which match a template image
- Steps:
  - Template definition = small image containing the pattern or object
- Sliding the Template
  - Slide the template over the image and computing a similarity measure at each position
- Similarity measures (examples)
  - SSD (Sum of Squared Differences) : measures the squared pixel-wise differences between the template and the image region
  - NCC (Normalized Cross-Correlation): measures the correlation between the intensity values of the template and the image
- Finding best match : Position where similarity measure is maximized

TRACKING MATCHING THAT LEVERAGE TEMPLATE MATCHING PRINCIPLES

- Correlation Filter-based Trackers
  - Mean Shift, KCF (Kernelized Correlation Filter)
- Template Matching with Machine Learning
  - Siames Networks : learn a similarity network between the target template and the region
- Discriminative Correlation Filter (DCF)
  - GOTURN
- Template Matching with Keypoints
  - Kanade-Lucas-Tomasi (KLT)
- Feature-Based Tracker's with Templates
  - MIL (Multiple Instance Learning)

MEAN SHIFT



Choose a search window size in the initial location

Compute the MEAN location in the search window

Center the search window at the mean

Repeat until convergence

- Steps (Iterative process) :
  - Initialization
    - Choose a starting point within a data space. This point represent initial estimate
  - Define Kernel
    - Kernel function determines the shape or the search window or region around the current estimate
  - Compute MEAN shift vector
    - It is a weighted average of the data points within the search window
  - Update Estimate
    - Shift in the direction of the mean shift vector
  - Convergence check
    - Compare the mean shift vector’s magnitude with a predefined threshold
  - Result
    - Peak of the probability density function

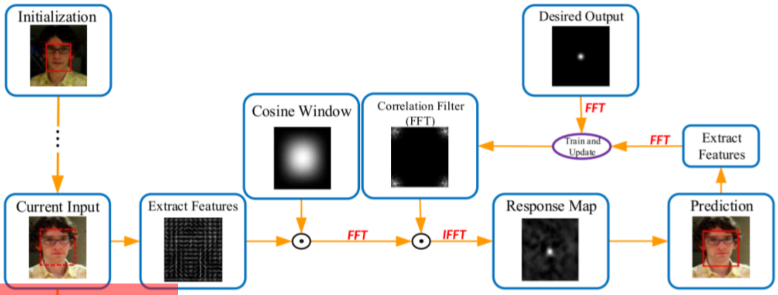
In the object tracking:
- Objects appearance is defined by histograms
- Mean Shift locates the peak of the color distribution of the target object in consecutive frames

CORRELATION FILTER-BASED TRACKERS

- Used in object tracking for the first time in 2010
  - High speed and precision, but it faces the challenges of boundary effect and scale effect
- Exemples of CF-based trackers:
  - MOSSE [10]
  - CSK [12]
  - KCF [15]
  - DSST [16]
  - BACF [17]

CF-based tracking algorithms are improved by template updating strategy, feature improvement and area detection

KCF TRACKER

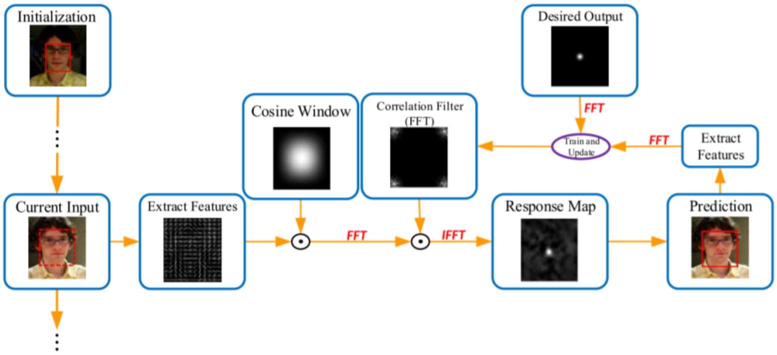


KCF may face challenges when target object is havy occluded or under goes significant deformations

Correlation filter training and updating procedures, all performed in the frequency domain, which is critical to achieve a high-frame-rate tracker.

- ❑ Generative model: it learns the correlation between the target object's appearance and the image region
- ❑ In the first frame, an initial correlation filter is trained based on the ground truth bounding box
- ❑ For each following frame:
  - ➢ Various local features are extracted and filtered by a cosine window in order to smooth the boundary effects
  - ➢ A response map is generated efficiently with a fast Fourier transform (FFT)
  - ➢ The position with the maximum value in this map is predicted as the new location of the target object
  - ➢ The new location of the object is used to update the correlation filter

KCF TRACKER

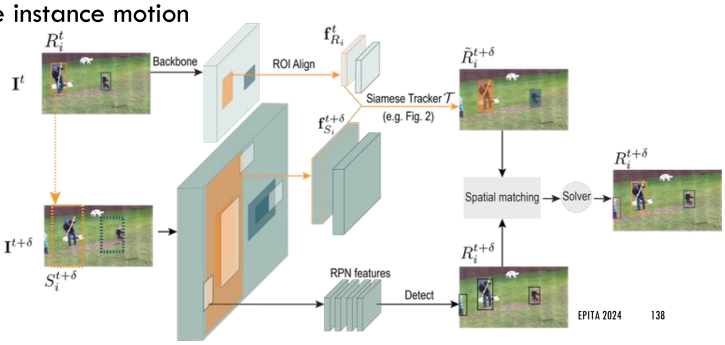


KCF may face challenges when target object is havy occluded or undergoes significant deformations

- KCF is known for its speed and efficiency, making it suitable for real-time object tracking
- It is robust for changes in scale, translation, and illumination

JOINT DETECTION AND TRACKING METHODS: SIAMMOT [21]

- ❑ Siamese architectures trained to determine the similarity between the target object and regions in subsequent frames
- ❑ Combine a region-based detection network (Faster-RCNN) with motion model
- ❑ Template matching to estimate instance motion
  - ➢ Siamese tracker searches instances at frame $I^{t+\delta}$ in a contextual window around its location at frame I^t (Rt) to generate R˜ t+δ
  - ➢ SiamMOT contains a motion model that tracks each detected instance from time t to t + δ
  - ➢ Detection network outputs a set of detected instances $R^{t+\delta}$
  - ➢ Spatial matching

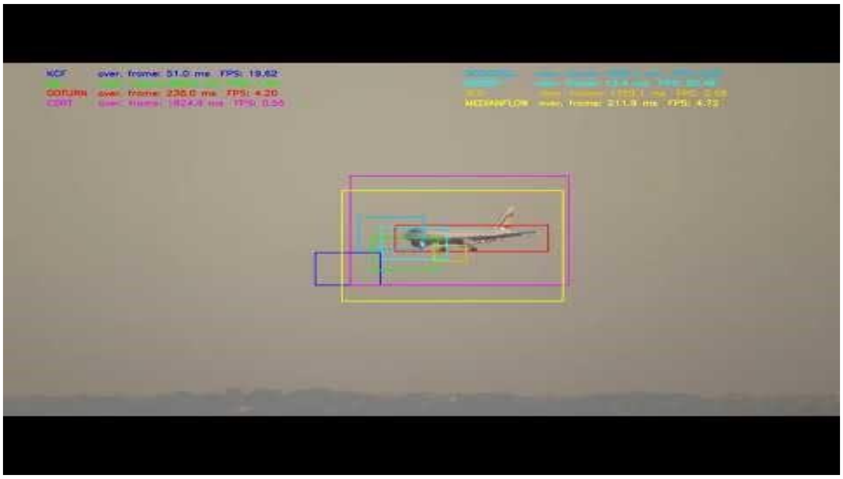


GOTURN [16]

- ❑ GOTURN = Generic Object Tracking Using Regression Networks
- ❑ Use a CNN to predict the target's bounding box displacement, relying on template matching principles
- ❑ Based Siamese architecture
- ❑ Trained in the supervised manner
  - ➢ It learns to predict the bounding box offset (regression)! Instead of learning an appareance model like KCF
- ❑ Adapted to track generic objects

OPENCV TRACKERS

- May face challenges in scenarios with occlusions, abrupt changes in scale, or non-rigid deformations
- It might struggle when multiple objects share similar color distributions

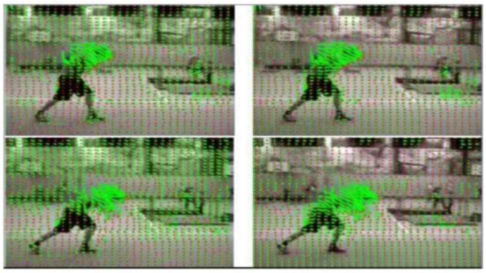

KCF over. frame: 51.0 ms FPS: 19.62
GOTURN over. frame: 236.0 ms FPS: 4.20
CSRT over. frame: 1824.8 ms FPS: 0.55
MEDIANFLOW over. frame: 211.9 ms FPS: 4.72

KLT TRACKER

- ❑ Firstly published in 1981 as an image registration method
- ❑ Closely related to optical flow estimation
- ❑ Improved many times, most importantly Carlo Tomasi
- ❑ Goal:
  - ➢ Find the parameters that allow the reduction in dissimilarity measurements between feature points

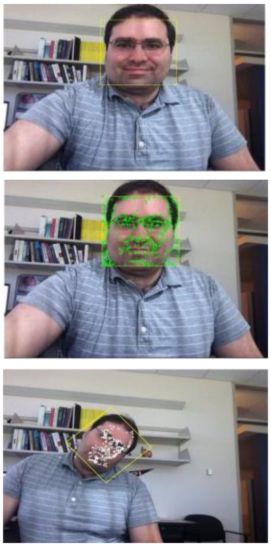
KLT TRACKER: OPTICAL FLOW

- ❑ Characterize the motion of pixels from one image to the next
  - ➢ Dense Optical Flow: Algorithm calculates displacement for all image
  - ➢ Sparse Optical Flow: Algorithm estimates displacement tension for a selective number of pixels in an image
- ❑ Compute velocity field
  - ➢ Pixels are associated velocity vector indicating the direction and magnitude of its motion
- ❑ Assumptions:
  - ➢ Brightness constancy (brightness of the point does not change between frames)
  - ➢ Spatial smoothness (neighboring pixels have similar motion)



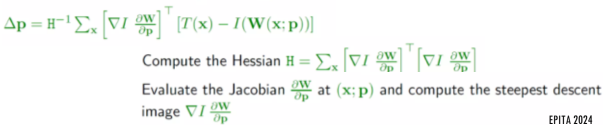
KLT TRACKER STEPS

- ❑ ROI select
  - ➢ For good conditioning patch must be textured/structured enough. Uniform patch = No information!
- ❑ Extract feature points in the first frame => eg. Harris corners
- ❑ Track the object:
  - ➢ For each Harris point compute motion (translation affine) between consecutive frames
  - ➢ Link motion vectors to get a track



KLT TRACKER: HOW TO ESTIMATE ALIGMENT?

- ❑ Set all allowable warps W(x;p), where p is a vector of parameters, For translations : $W(x;p) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$
- ❑ The best alignment minimizes image dissimilarity: $\sum_x [I(W(x;p)) - T(x)]^2$
  => <u>Non linear optimization!</u>
  - It is assumed that some p is known and best increment Δp is sought
  - The modified problem is solved with respect to Δp: $\sum_x [I(W(x;p + \Delta p)) - T(x)]^2$
  - Linearized by performing first order Taylor expansion : $\sum_x [I(W(x;p)) + \nabla \frac{\partial w}{\partial p} \Delta p - T(x)]^2$
  - p gets updated p ← p+ Δp
  - Until Δp≤π



$$\Delta \mathbf{p} = \mathrm{H}^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\top} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$$

Compute the Hessian $\mathrm{H} = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\top} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]$

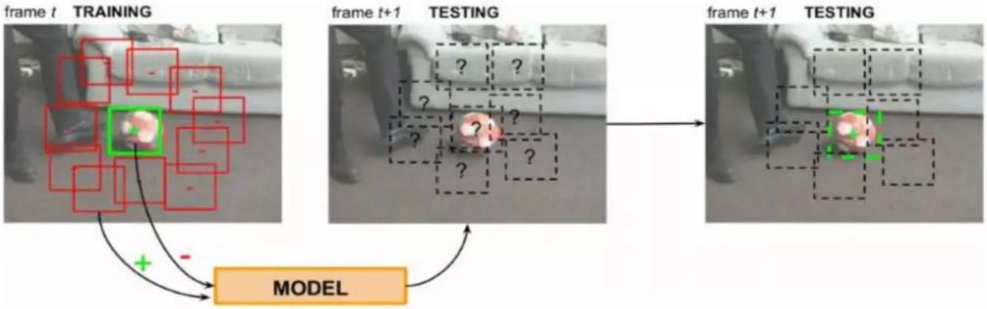
Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x};\mathbf{p})$ and compute the steepest descent image $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$

WHAT ABOUT DEEP LEARNING? OR CNN FOR TRACKING?

- ❑ Features learning
  - ➢ Deep learning models are adept at learning hierarchical and abstract features directly from raw data, eliminating the need for handcrafted feature engineering
- ❑ Representation Learning
  - ➢ Deep trackers learn to represent objects in a feature space, enabling robustness to variations in object appearance, scale, and pose
- ❑ End-to-End Learning
  - ➢ Trackers designed to operate in an end-to-end manner, directly predicting the object's state or location without explicit intermediate steps

DEEP LEARNING-BASED TRACKING

- ❑ Learn online a binary classifier (+ is object, -background)
- ❑ Redect the object at every frame + update the classifier



REID

- ❑ ReID integration
  - ➢ **Recovery from Occlusions:** In cases where an object is temporarily occluded and reappears in the field of view, ReID aids in re-establishing the identity of the object, preventing tracking failures
  - ➢ **Cross-Camera Tracking:** ReID helps in associating the same person or object as it moves between cameras by comparing the appearance features across different views
- ❑ Re-ilentification (ReID) relies on discriminative features ReID features robust to changes in lighting, pose, shape

- ❑ Principe:
  - ➢ Build classifier over database
  - ➢ Strip the final classification layer. Assuming a classical architecture, left with a dense layer producing a single feature vector, waiting to be classified. That feature vector becomes our "appearance descriptor" of the object
  - ➢ Pass all detected features from the image to this network (model) => [N,1] dimensional feature vector
  - ➢ How close their appearance is ? = > Compare the vectors produced by the re-id network for successive images

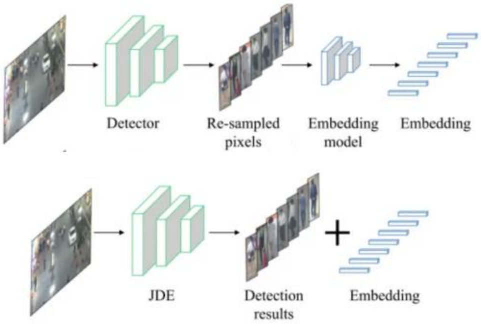
REID

- ❑ Image-ReID datasets
  - ➢ Market1501, DukeMTMC, CUHK03, VIPeR, PRID, SensReID
- ❑ Models
  - ➢ ResNet, Inception, Xception, …
  - ➢ Lighweight models
    - MobileNet, ShuffleNet, SqueezeNet, …
  - ➢ ReID- specific models
    - ResNet, OSNet, OSNet-AIN, MLFN, PCB, MuDeep, …

CNN-BASED TRACKING APPROCHES CATEGORISATION

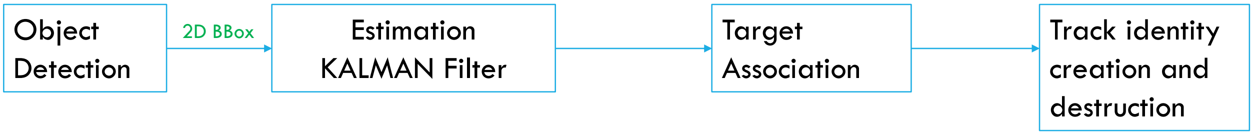
According to the diffrent implementations of detection, embedding extraction, motion prediction:

- ❑ Seperate detection and embedding (SDE)
  - ➢ Exploit two separate models; one for object detection and another for generating visual appearance features, which are trained separately on different tasks
- ❑ Joint detection and embedding (JDE)
  - ➢ Simultaneously generate the location and apparence features of targets in a single forward pass
- ❑ Joint Detection and Tracking (JDT)



When object detection and re-identification are performed separately, they can not benefit each other!

SEPERATE DETECTION AND EMBEDDING: DEEP SORT [18]



Object Detection

2D BBox

Estimation KALMAN Filter

Target Association

Track identity creation and destruction

Using Linear Constant Velocity model

IoU, Deep Appearence Descriptor

Each detected object is represented by a state vector consisting of:

- *the center location*
- *Scale and aspect ratio of its bounding box*
- *velocity and the rate of change in its scale*

Cost Matrix = weighed sum of Mahalanobis distance & appeareance distance (cosine distance)

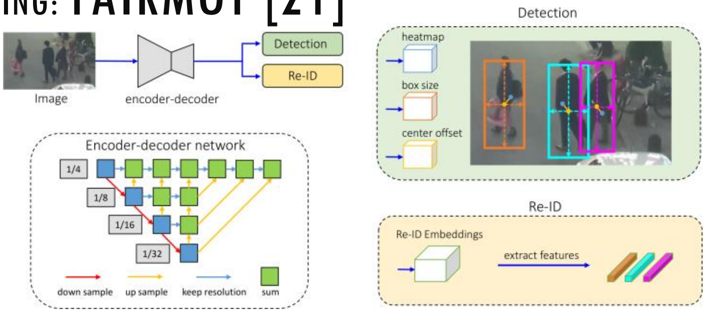
Assignement:

- Hungarian algorithm
- Cascade matching: tracklets with a smaller age are compared and associated with new detections before those with a larger age

BYTETRACK [22]

- Exploit YOLOX, a recent object detector in the YOLO series, to detect persons
- Simple object association algorithm called BYTE
  - Principe: keep non-background low score boxes (typically discarded after the initial filtering) for a secondary association step between previous frame and next frame based on their similarities with tracklets.
  - Require a motion model, which is a Kalman filter, to predict the location of existing tracklets in the next frame
  - IOU-based distance for object association
  - Gives priority to detection boxes with high confidence scores
  - Bounding box association with Hungarian algorithm

JOINT DETECTION AND EMBEDDING: FAIRMOT [21]



Key IDEA:

Two homogeneous branches, detection and ReID branches

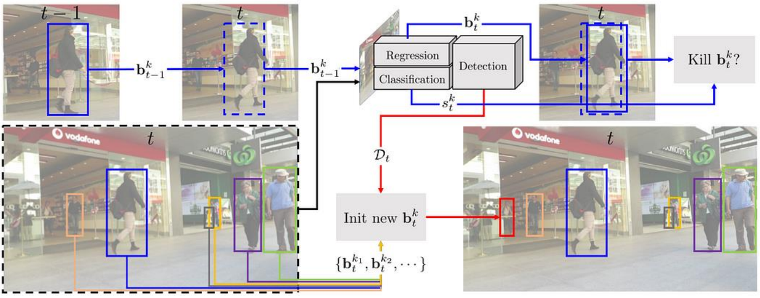
- ❑ **Detection branch:** three parallel heads appended to DLA-34 [ResNet-34 with Deep Layer Aggregation (DLA)] to estimate:
  - ➢ Heatmaps => Heatmap head estimates the object’s center
  - ➢ Object center offsets => Object center offsets head aims to localize object centers more precisely
  - ➢ Bounding box sizes => Bounding box head is responsible for estimating the height and width of objects
- ❑ **ReID branch:** a convolution layer with 128 kernels on top of backbone features to extract features for each location

JOINT DETECTION AND TRACKING METHODS: TRACKTOR [19]

- ❑ Key IDEAS:
  - ➢ Accomplish MOT only with a object detector
  - ➢ Directly uses the previous frame tracking boxes as region proposals and then applies the bounding box regression to provide tracking boxes on the current step, thus eliminating the box association procedure

- *Use bbox from last frame (t-1) to do RoI Pooling*
- *Regress offset to predict detection in the next frame*

- *The object’s identity is automatically transferred from frame t−1 to frame t*
- *New track is initialized if a detection has no substantial IOU with any bbox of the set of actif tracks*
- *Classification score of the new bbox position is used to kill potentially occluded tracks*

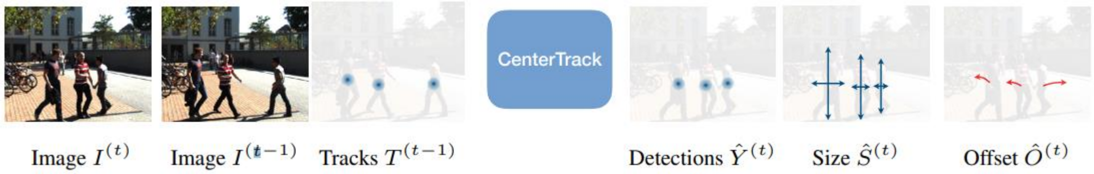


JOINT DETECTION AND TRACKING METHODS: TRACKTOR++

- ❑ BUT:
  - ➢ It relies on an assumption that target objects move only slightly between frames, which may hold in case of high frame rate sequences
- ❑ Tracktor++, extended Tracktor
  - ➢ First extension: the use of a motion model to handle the case of low frame rate sequences and the case of moving cameras. In which cases, a bounding box in frame t−1 might not overlap with its target object in frame t at all; consequently, the regression head does not have any clue to refine its location correctly
    - Bbox motion: most common one is constant velocity (CVM). Shift bbox from previous frame with velocity first, then regress offset
    - Camera motion compensation: : image registration to compensate for camera rotation and translaton by ECC (Enhanced Correlation Coefficient)
  - ➢ Second extension: exploit a ReID model to verify if the target object in frame t−1 really matches its refined location in frame t. If not, this tracklet of the target object is deactivated
    - Siamese network to extract embedded features

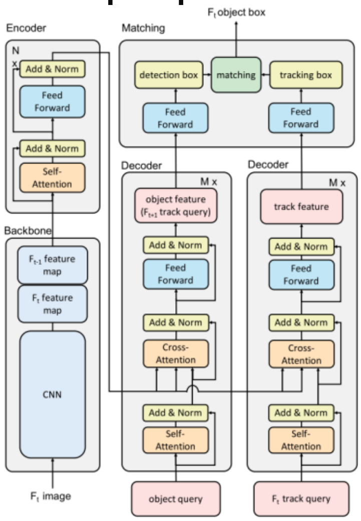
JOINT DETECTION AND TRACKING METHODS: CENTER TRACK [20]

- ❑ KEY IDEAS:
  - ➢ Each object is represented by a single point at the center of its bounding box
  - ➢ Detector is also trained to output an offset vector from the current object center to its center in the previous frame
  - ➢ Displacement is used to perform association
- ❑ Based on an object detector called CenterNet
  - ➢ Generate a low-resolution heatmap representing the chance to find an object's center (local maximum is considered the center of a detected object) and a size map representing the width and height of an object at each location



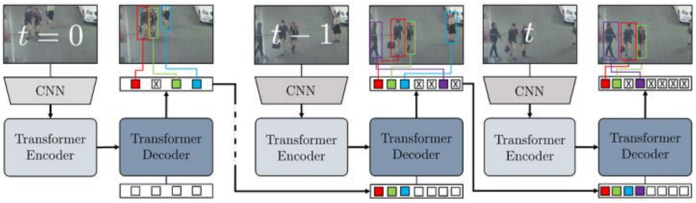
JOINT DETECTION AND TRACKING METHODS: TRANS TRACK [21]

- Architecture is extended from Detection Transformer (DETR)
- Steps:
  - Video frame is fed into a CNN backbone to compute a feature map
  - Feature maps of the current frame t and the previous frame t-1 are linearly projected and reshaped into a sequence of tokens
  - This sequence is processed by a Transformer encoder generates keys (in which feature representation has been enhanced)
  - Two parallel decoders:
    - Object detection: it takes learned object query as input and predicts detection boxes
      - ✓Object query: Set of learnable parameters, trained together with all other parameters in the network
    - Object propagation: it takes the object feature from previous frames "track query", as input and predicts the locations of the corresponding objects on the current frame

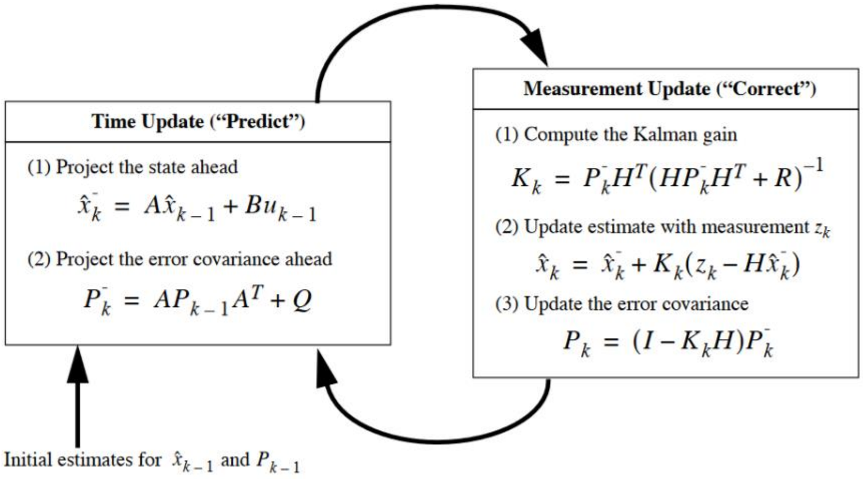


TRACKING WITH TRANSFORMER: TRACK FORMER [22]

- Jointly performs object detection and tracking-by-attention
- Similar to TransTrack but exploits only one Transformer decoder
- Architecture:
  - CNN for image feature extraction
  - Transformer encoder for image feature encoding with self-attention
  - Transformer decoder for decoding of queries with self- and encoder-decoder attention
  - Track association happens via attention in the Transformer decoder



KALMAN FILTER (KF)



Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

(3) Update the error covariance

$$P_k = (I - K_k H)P_k^-$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

KALMAN FILTER (KF)

❑ KF is intended to estimate the state of a system at time k, using the linear stochastic difference equation

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \qquad (1)$$

❑ KF is always paired with the measurement model, that describes a relation between the state and measurements at the current step k

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k \qquad (2)$$

A state transition matrix (nxn) relating the previous time step k-1 to the current state k

B control input matrix (nxl), applied to the optional control input

H transformation matrix (mxn) that transforms the state into the measurement domain

$\mathbf{w}_k$ process noise vectors with covariance Q

$\mathbf{v}_k$ measurement noise vector with the covariance R

State transition matrix A and control matrix B

Based on Kinematic equation, the relation between the postion x,y and valocity $\dot{x},\dot{y}$

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}\Delta t + 1/2\ddot{x}_{k-1}\Delta t^2 \\ y_{k-1} + \dot{y}_{k-1}\Delta t + 1/2\ddot{y}_{k-1}\Delta t^2 \\ \dot{x}_{k-1} + \ddot{x}_{k-1}\Delta t \\ \dot{y}_{k-1} + \ddot{y}_{k-1}\Delta t \end{bmatrix} \tag{3}$$

We can write eq.(3) into the form of matrix multiplication as follows:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & \frac{1}{2}(\Delta t)^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \ddot{x}_{k-1} \\ \ddot{y}_{k-1} \end{bmatrix} \tag{4}$$

The eq.(4) can be simplified as follows:

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & \frac{1}{2}(\Delta t)^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \mathbf{a}_{k-1} \tag{5}$$

KALMAN FILTER (KF)

- ❑ State transition matrix A and control matrix B

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

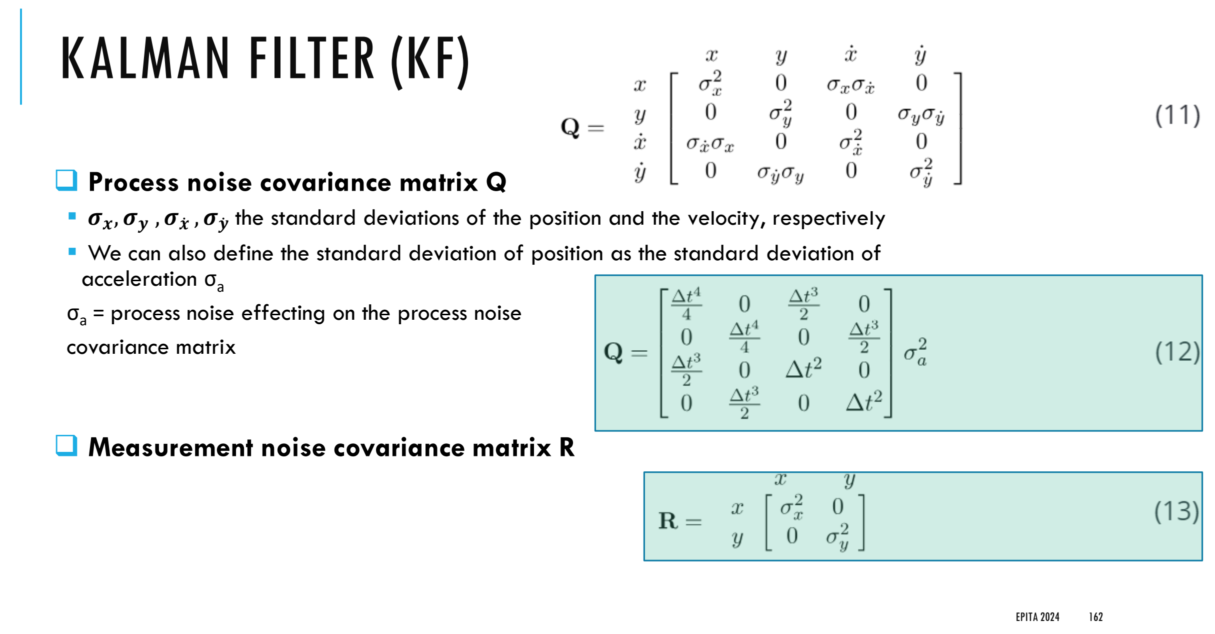
$$\mathbf{B} = \begin{bmatrix} \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & \frac{1}{2}(\Delta t)^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{7}$$

- ❑ Transformation matrix H

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k \tag{8}$$

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} + \mathbf{v}_k \tag{9}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$



KALMAN FILTER (KF)

$$\mathbf{Q} = \begin{array}{c|cccc} & x & y & \dot{x} & \dot{y} \\ \hline x & \sigma_x^2 & 0 & \sigma_x\sigma_{\dot{x}} & 0 \\ y & 0 & \sigma_y^2 & 0 & \sigma_y\sigma_{\dot{y}} \\ \dot{x} & \sigma_{\dot{x}}\sigma_x & 0 & \sigma_{\dot{x}}^2 & 0 \\ \dot{y} & 0 & \sigma_{\dot{y}}\sigma_y & 0 & \sigma_{\dot{y}}^2 \end{array} \qquad (11)$$

❑ Process noise covariance matrix Q

- $\boldsymbol{\sigma_x, \sigma_y, \sigma_{\dot{x}}, \sigma_{\dot{y}}}$ the standard deviations of the position and the velocity, respectively
- We can also define the standard deviation of position as the standard deviation of acceleration σ_a

σ_a = process noise effecting on the process noise covariance matrix

$$\mathbf{Q} = \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \sigma_a^2 \qquad (12)$$

❑ Measurement noise covariance matrix R

$$\mathbf{R} = \begin{array}{c|cc} & x & y \\ \hline x & \sigma_x^2 & 0 \\ y & 0 & \sigma_y^2 \end{array} \qquad (13)$$