Lecture 9: Multi-robot mutual localization

B3M33MRS — Aerial Multi-Robot Systems

Doc, Ing. Martin Saska, Dr. rer. nat.

Labs: Ing. Tomáš Báča, Ph.D

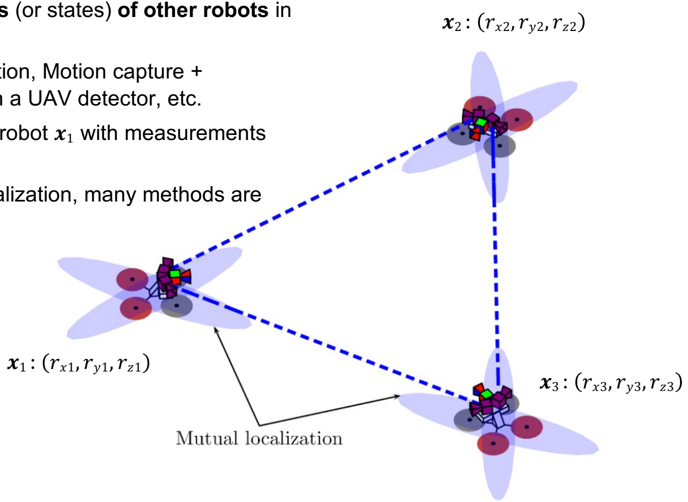
Multi-Robot Systems group, Faculty of Electrical Engineering
Czech Technical University in Prague

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

MRS MULTI-ROBOT SYSTEMS GROUP

What is multi-robot mutual localization?

- **A system that determines positions** (or states) **of other robots** in the environment
- Various setups: GNSS + communication, Motion capture + communication, onboard camera with a UAV detector, etc.
- For example, a system that provides robot $\boldsymbol{x}_1$ with measurements $L_1 = \{(r_{x2}, r_{y2}, r_{z2}), (r_{x3}, r_{y3}, r_{z3})\}$
- Although different from robot self-localization, many methods are similar

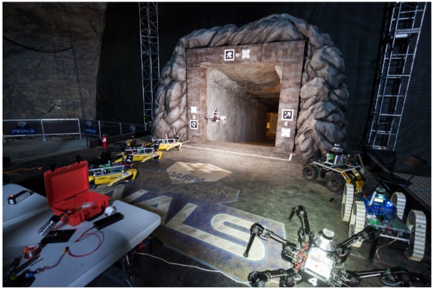


Why is mutual localization important?

- Collision avoidance
- Cooperation/coordination/collaboration
  - ✓ Formation control, swarming
- Mutual physical interaction
  - ✓ Aerial interception

Almost all multi-robot scenarios require mutual localization!

Source: MRS archive

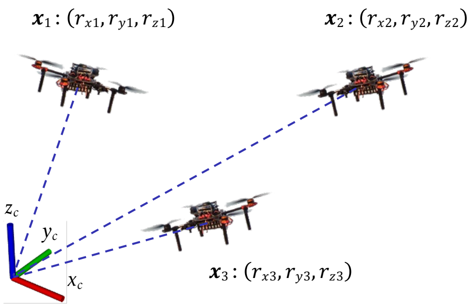


Clasification of multi-robot mutual localization

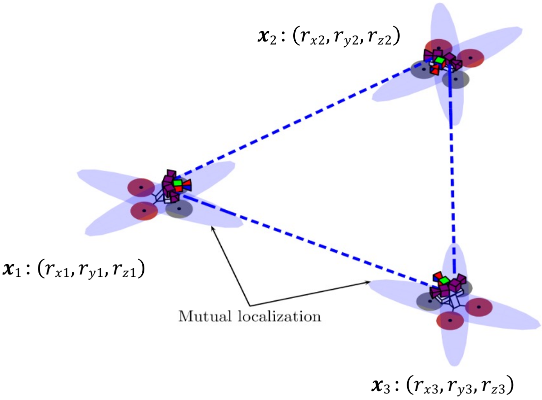
- Frame of reference
  - ✓ Absolute vs. relative localization
  - ✓ Requiring communication vs. communication-less
- Markers
  - ✓ Marker-based vs. marker-less localization
  - ✓ Active markers vs. passive markers
- Time of flight-based localization (UWB)
- Map sharing-based localization

- **Absolute localization:** localization in a common coordinate frame using an external source

- **Relative localization:** localization using only onboard sensors
- Each UAV with its own coordinate frame



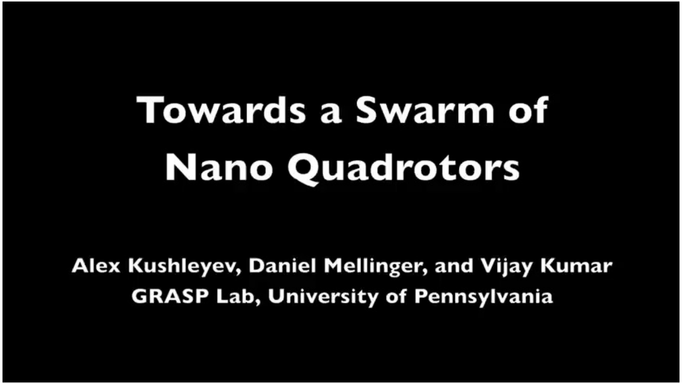
common frame



Absolute (global) localization

- GNSS-based localization (GPS, Galileo, GLONASS …)
- Motion Capture (Vicon, Optitrack, Qualisys …)
- Map (or landmark) merging
- Pre-installed radio beacons

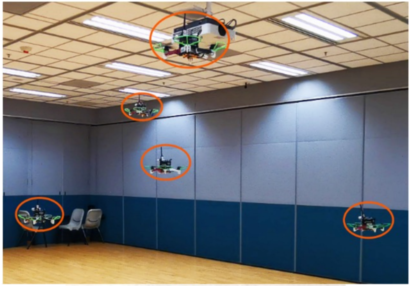
Source: archive of GRASP lab at UPENN

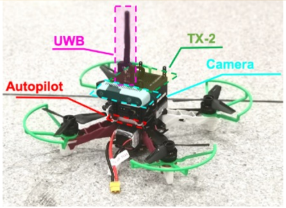


Source: archive of GRASP lab at UPENN

Relative localization

- Marker detection (visual, IR, UV, etc.)
- Radio-based multilateration (UWB)
- Marker-less detection (CNNs, RADAR, LiDAR, acoustic, etc.)
- Hybrid approaches





Xu, Hao, et al., "Decentralized visual-inertial-UWB fusion for relative state estimation of aerial swarm", *IEEE ICRA* 2020.

Frame of reference: Absolute vs. relative localization

- **Absolute localization**

+ Usually little or no drift
+ Easy to use from the user's perspective
+ Can be extremely accurate and low-delay
+ Simplifies multi-robot planning
− **Often relies on existing infrastructure** (satellites, MoCap cameras, the beacons)
− **Requires communication between robots** to share the measurements/the map

- **Relative localization**

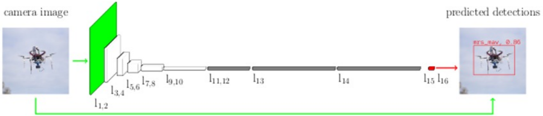
+ **no reliance on infrastructure**
+ **communication not necessary**
+ may work almost in any environment
− accuracy/drift varies
− can suffer from false positives → **lower reliability**

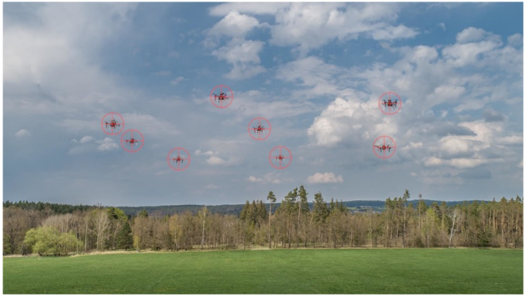
Markers: Marker-based vs. marker-less localization

- **Marker-based localization:** group members equipped with distinct markers
- Visual markers for cameras, IR markers for MoCap, UWB beacons, active/passive markers

Source: MRS archive

- **Marker-less localization:** group members localized without markers
- detection based on visual appearance, 3D shape, noise, radio reflectivity, …





Types of multi-robot mutual localization: Markers

- **Marker-based localization**

\+ Simpler to implement
\+ Usually **more reliable**
\+ Usually **more accurate**
\+ Easier to estimate relative orientation
− **Not always applicable** (e.g. non-cooperating targets)
− Markers take up space & weight (active markers also energy)
− Often difficult to localize from an arbitrary direction

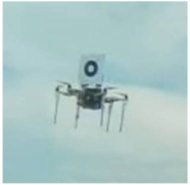
- **Marker-less localization:**

\+ **Can be used for non-cooperating targets**
\+ Bio-inspired approach
− A harder recognition problem to solve
− **More complex** approaches required
− Usually less robust and less accurate

Passive markers

- WhyCON
- WhyCode
- ArUco
- QR
- Color-based markers
- Other fiducials

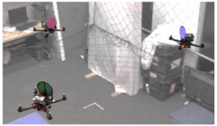
+ Simple to implement
+ Cheap and easy to install
− Dependent on light conditions
− Large size for significant range
− Heavy, non-aerodynamic, complicated maintenance
− Directional dependent



Source: MRS archive

Source: Ulrich's archive

Evangeliou, N., Chaikalis, D., Tsoukalas, A., & Tzes, A., "Visual Collaboration Leader-Follower UAV-Formation for Indoor Exploration", *Frontiers in Robotics and AI, 8, 2021.*

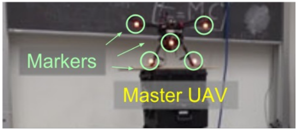


Tron, R., Thomas, J., Loianno, G., Polin, J., Kumar, V. & Daniilidis, K., "Vision-based formation control of aerial vehicles", *RSS, 2014.*

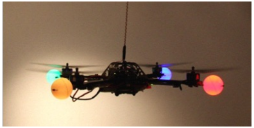
Active markers

- Monochrome visible light LEDs
- RGB LEDs
- Infrared LEDs
- Ultraviolet LEDs

+ Lighting-independent
+ Small image - typically points → less computationally demanding
+ Higher reliability
+ Smaller footprint
+ Usually less precise pose estimate
− Complex implementation (HW and SW)
− Power consumption

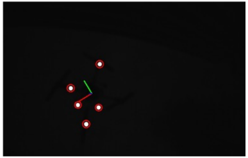


L. Teixeira, F. Maffra, M. Moos and M. Chli, “VI-RPE: Visual-Inertial Relative Pose Estimation for Aerial Vehicles”, *IEEE RAL, vol. 3, no. 4, pp. 2770-2777, 2018.*



Achtelik, M., Zhang, T., Kuhnlenz, K., & Buss, M., “Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors”, *IEEE ICMA, 2019.*

Source: MRS archive

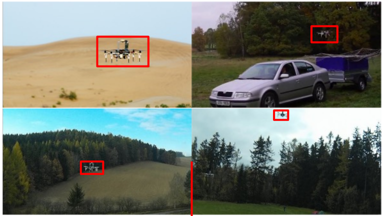


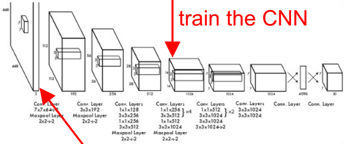
Faessler, M., Mueggler, E., Schwabe, K., & Scaramuzza, D., “A monocular pose estimation system based on infrared leds”, *IEEE ICRA, 2014.*

Marker-less relative localization

- Typically used if the target
  - ✓ is too small to carry markers
  - ✓ is non-cooperative (i.e. collision avoidance with non-team members)
  - ✓ is hostile (i.e. intruder/attacker drone)
- In comparison with marker-based methods, it is
  - ✓ harder to estimate full pose (e.g., orientation estimation)
  - ✓ harder to provide unique IDs of targets (can be mitigated by multi-target tracking)
  - ✓ less restrictive usage

Marker-less relative localization - visual detection





train the CNN

feed images to the CNN

Source of images: Redmon's and MRS archives

- Deep learning-based methods - e.g. using Convolutional Neural Network (CNN)

  + Straightforward: only needs a large dataset for training

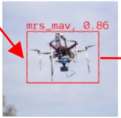
  − Assumes known visual appearance and size of targets

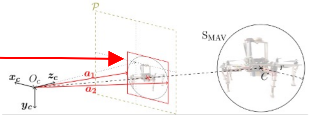
  − Deep learning is sensitive to domain changes (e.g. environment)

  − Unreliable distance estimation

  − Sensitive to sensor changes

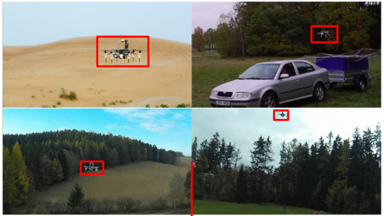
the CNN estimates bounding boxes



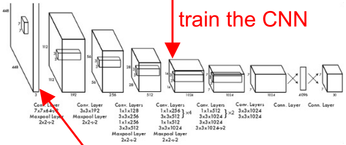


calculate relative positions of the targets using a calibrated camera projection model

Marker-less relative localization - visual detection



train the CNN

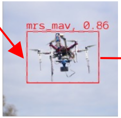


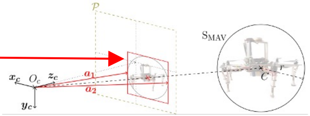
feed images to the CNN

Source of images: Redmon's and MRS archives

- Deep learning-based methods (a typical approach)
  - ✓ A large labeled dataset for training and validation
  - ✓ A neural network (usually a CNN such as YOLO)
- The main steps of the detection algorithm:
  - ✓ Train the CNN to detect the desired targets in images
  - ✓ Feed the CNN with camera images
  - ✓ Use the detections to estimate targets' relative positions

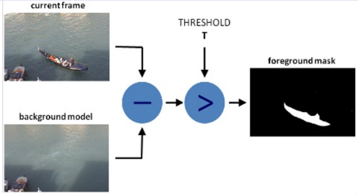
the CNN estimates bounding boxes





calculate relative positions of the targets using a calibrated camera projection model

Marker-less relative localization - visual detection



https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html

J. Li, D. H. Ye, … C. Bouman, "Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs)", *IROS, 2016.*

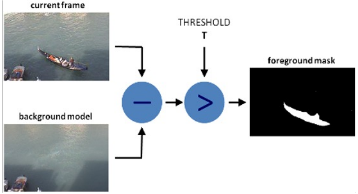
- Background subtraction steps:
  - ✓ Detect salient features in subsequent images
  - ✓ Calculate the transformation between corresponding features
  - ✓ Obtain the background model by transforming the 2nd image
  - ✓ Subtract the background model from the 1st image
  - ✓ Filter the resulting differential image
  - ✓ Detect and classify moving objects in the differential image
- Optical flow can also be used to remove the background
- A classifier is often used to further filter the detected objects

\+ Can detect distant and small targets

\+ Agnostic to the target's visual appearance

– Unreliable distance estimation

Marker-less relative localization - visual detection



https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html

J. Li, D. H. Ye, … C. Bouman, "Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs)", *IROS, 2016.*

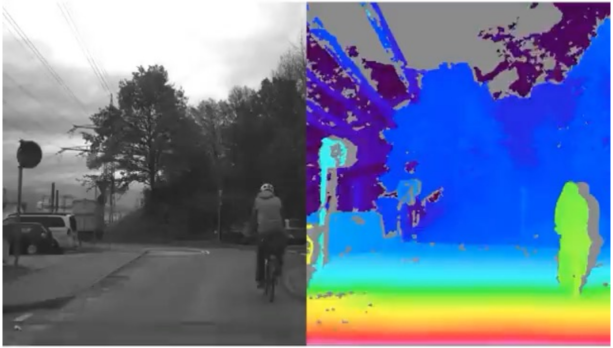
- Visual tracking steps:
  - ✓ Initialize with a bounding box of the target in an image
  - ✓ In the subsequent image, find a visually similar shape nearby
  - ✓ Estimate the target's motion to predict its position in the next frame
  - ✓ Repeat from the second step
- Related to multi-target tracking (later in this lecture)

\+ Can associate the targets and fill out frames when detection fails

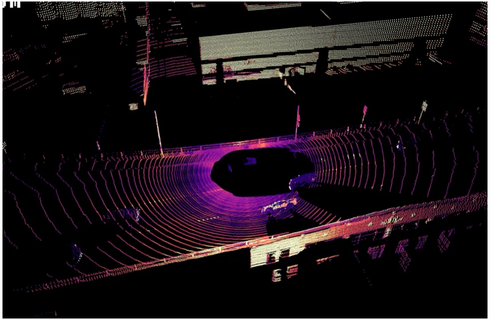
− Can latch onto a visually similar background shape

Marker-less relative localization - spatial detection

- Spatial information about the environment provided by stereo cameras and 3D LiDARs used for flying object detection
- Flying objects “stand out” in spatial sensor modalities as compared to visual

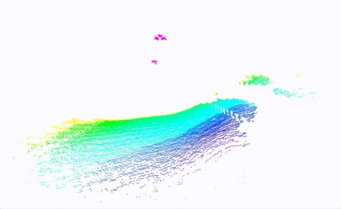


https://nerian.com

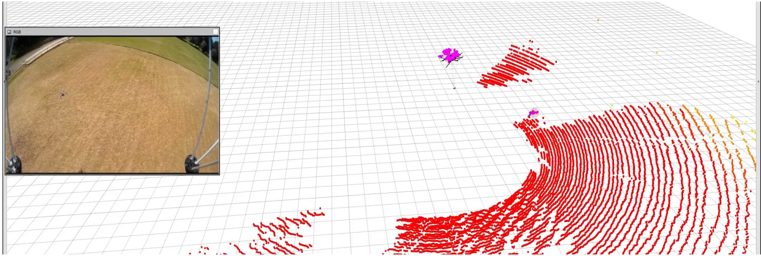


https://ouster.com/products/scanning-lidar/os1-sensor/

Marker-less relative localization - spatial detection



- \+ Appearance-agnostic
- \+ Implicit estimation of the target's relative 3D position
- − Ad-hoc methods that are not always reliable (ongoing research)



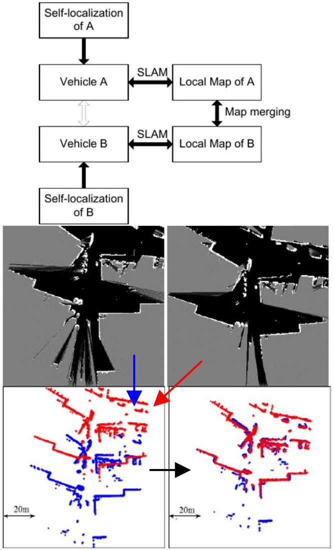
M Vrba, D Heřt and M Saska, "Onboard Marker-Less Detection and Localization of Non-Cooperating Drones for Their Safe Interception by an Autonomous Aerial System.", *IEEE RAL, 2019.*

Ultra-Wideband (UWB) localization

- UWB beacons placed on the UAVs and optionally in the environment
- ToF-based range measurements between the beacons provided by the UWB system
- Relative positions of beacons multilaterated – similar approach as in task 02
- UWB provides:
  - ✓ Good accuracy and robustness to interference and reflections provided by UWB - compared to e.g. Bluetooth
  - ✓ Precise and reliable localization especially when combined with other methods

Map merging

- Neighboring agents sharing their own maps
- Only salient features/landmarks can be transmitted to reduce bandwidth
- Each agent tries to localize itself in the map of the other robots (similarly to normal SLAM algorithms)
- Requires communication
- Susceptible to local minima when merging
- A global map provided for free



H. Li and F. Nashashibi, "Multi-vehicle cooperative localization using indirect vehicle-to-vehicle relative pose estimation", *CVES, 2012.*

Multi-target tracking and filtration for localization - motivation

Lecture 9: Mutual localization
Martin Saska
Motivation
Taxonomy
Marker-based
Marker-less
Other sensors
Tracking & filtration
Distance estimation
Conclusion

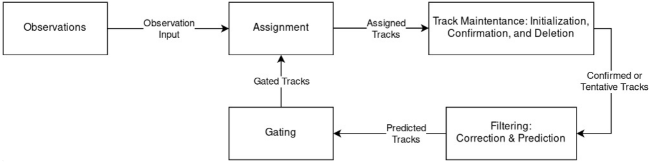
- Relative localization methods can produce noisy measurements and false positives
- Not all methods provide unique identification of the detected targets
- Required estimating more states than the localization methods provide
- Fusing multiple different localization methods together to improve accuracy, robustness, and reduce delay

Multi-target tracking and filtration algorithms with a motion model of the robot can help!

Multi-target tracking and filtration for localization

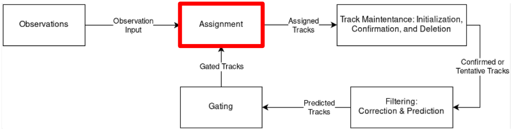
- Issues to be solved in Multi-Target Tracking (MTT):
  - ✓ An unknown number of relevant objects (UAVs in our case) in the observed area
  - ✓ Objects appearing and disappearing (e.g. when entering/leaving the area)
  - ✓ Anonymous inaccurate position estimates of the objects
  - ✓ Tracking the objects – identification of which detection belongs to which object
  - ✓ Filtering the position estimates to obtain a better state estimation of each object

- Input of an MTT algorithm in one iteration: a set of detections and their corresponding position estimates = a single observation
- Desired output of an MTT algorithm: filtered state estimates of the identified objects in the observed area = confirmed tracks



https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

Multi-target tracking and filtration for localization

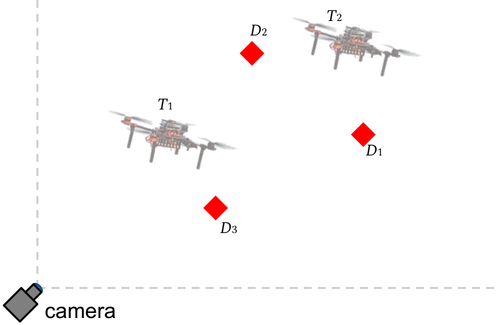


https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

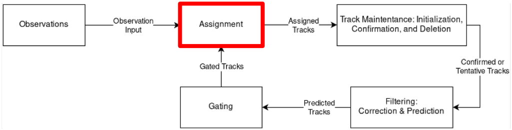
- How to assign the detections to the tracks? – **Global Nearest Neighbor** (GNN):

✓ An example of a **greedy method** using track-detection distance:

1. Start with any valid assignment of detections to tracks
2. Let $i := 1$ be the index of the first assigned pair
3. For an assigned pair $\{T_i, D_i\}$, find $j \geq i$ so that after swapping D_i with D_j the sum of distances between all assigned pairs is minimized. Swap D_i with D_j
4. Increment i and repeat step 3 for all pairs $\{1, \ldots, n\}$

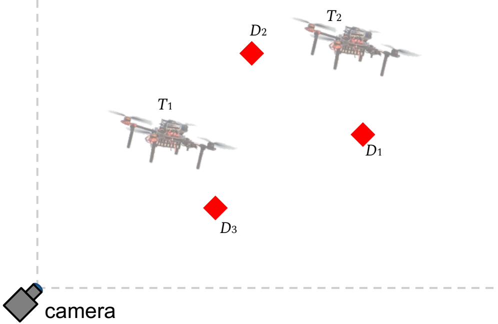


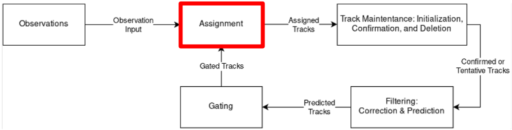
Multi-target tracking and filtration for localization



https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

- How to assign the detections to the tracks? – **Global Nearest Neighbor** (GNN):
  - ✓ The greedy method provides a reasonable feasible solution, although not necessarily the optimal solution
  - ✓ But in most of the cases good enough
  - ✓ Asymptotic complexity: $O(N^2)$
  - ✓ An optimal solution - using the Hungarian algorithm with $O(N^3)$ asymptotic complexity





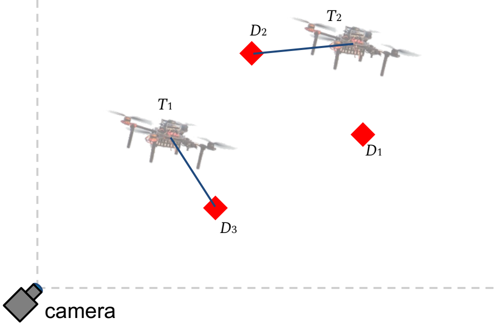
https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

- How to assign the detections to the tracks? – **Global Nearest Neighbor** (GNN):
  - ✓ Using Euclidean distance between the track and detection:

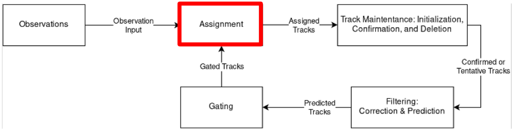
$$D1 \leftrightarrow \emptyset$$

$$D2 \leftrightarrow T2$$

$$D3 \leftrightarrow T1$$



Multi-target tracking and filtration for localization



https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

- How to assign the detections to the tracks? – **Global Nearest Neighbor** (GNN):

✓ Using track-detection distance scaled by track and detection uncertainties:

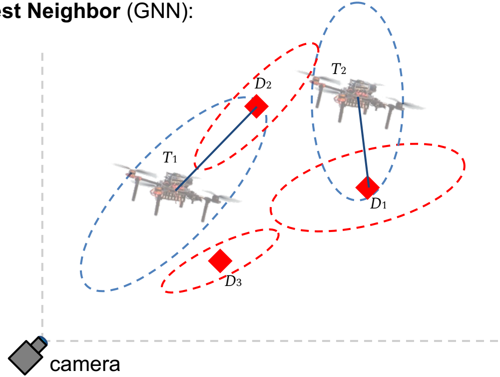
$$D1 \leftrightarrow T2$$

$$D2 \leftrightarrow T1$$

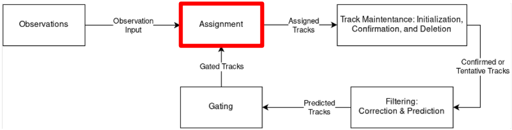
$$D3 \leftrightarrow \emptyset$$

✓ Taking known uncertainties into account - statistical improving the algorithm

✓ Applying only track uncertainties or even detection uncertainties



Multi-target tracking and filtration for localization



https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

- How to assign the detections to the tracks? – **Joint Probabilistic Data Association** (JPDA):

  ✓ Multiple detections used to update a single track (no hard associations):

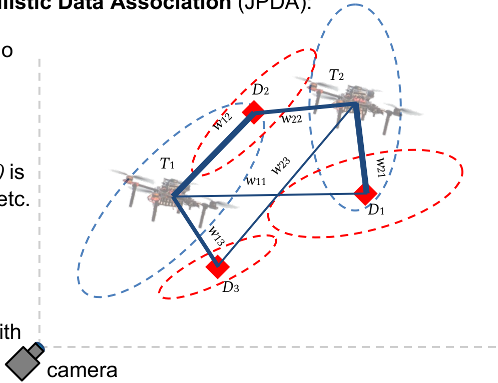
$$w_{11}D_1 + w_{12}D_2 + w_{13}D_3 \leftrightarrow T_1$$
$$w_{21}D_1 + w_{22}D_2 + w_{23}D_3 \leftrightarrow T_2$$

  with $w_{ij} = d(T_i, D_j)$, where the distance function $d(\cdot)$ is Euclidean, Mahalanobis, measurement likelihood, etc.

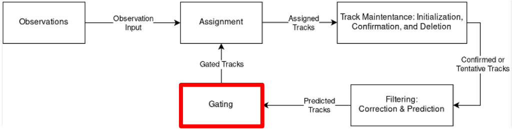
  ✓ Used to simplify the Assignment problem

  ✓ Can deal with multiple detections corresponding to the same object, track overlapping, etc.

  ✓ Number of track-detection combinations growing with the number of tracks and detections

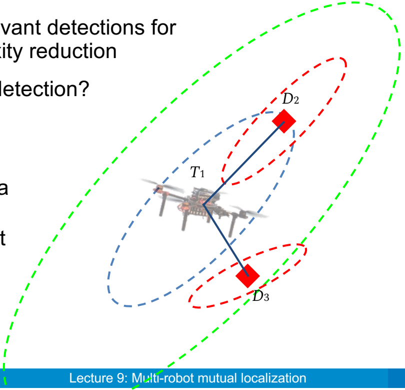


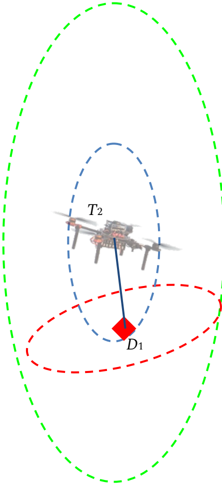
Multi-target tracking and filtration for localization



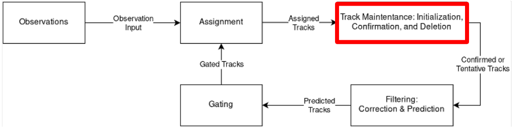
https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

- Complexity of the Assignment problem increasing with many tracks and detections!
  - ✓ Considering only relevant detections for each track – complexity reduction
  - ✓ What is a “relevant” detection?
  - ✓ A possible approach: Using the track’s uncertainty to select a gating volume and ignore detections that lie outside of it



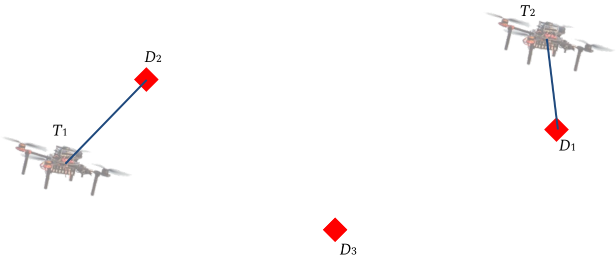


Multi-target tracking and filtration for localization

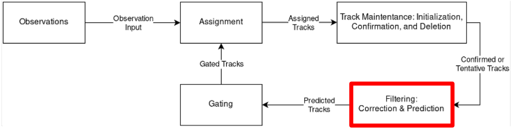


https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

- How to deal with new objects entering the observed area and old objects leaving the area?
  - ✓ Some detections not assigned to any track and vice-versa
  - ✓ Unassigned detections used to initialize new tracks (but only tentative – to avoid false positives)
  - ✓ Tracks not updated for a long time are deleted
  - ✓ Tracks with enough assigned detections are confirmed
  - ✓ Considering uncertainty etc.

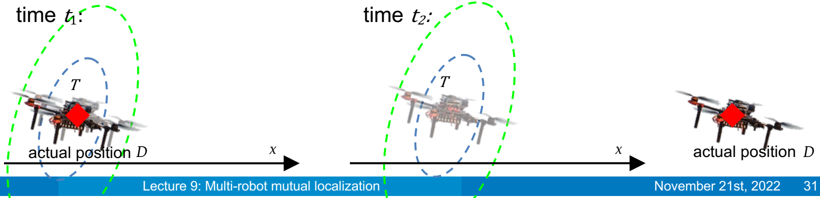


Multi-target tracking and filtration for localization



https://www.mathworks.com/help/fusion/ug/introduction-to-multiple-target-tracking.html

- What to do with noisy measurements?
- Estimating and filtering states of the targets: e.g. a Kalman Filter:
  - ✓ To improve accuracy, to provide estimation of the track uncertainties (covariance matrices)
  - ✓ To enable estimation of non-measured states of the targets (e.g. velocity, acceleration) and prediction of future movement, e.g. using a state-space motion model of a mass point in 3D

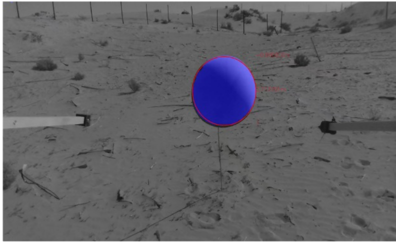


Monocular distance estimate

- Relative 3D position task divided into relative bearing and range (distance) estimation sub-tasks
  - ✓ Bearing calculated from pixel position of an image of the object using a calibrated projection model
  - ✓ More difficult task is **estimating the range** if assuming a monocular system
- **Using perspective** – object image size depends on its distance from the camera
- The distance can be retrieved from

1) a segment area (pixel count) vs. a known object size
2) a dimension of a bounding box vs. a known object size

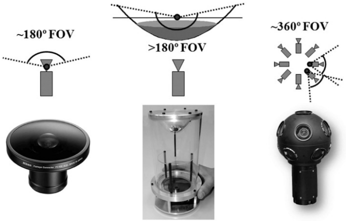
- Both approaches require a priori knowledge of object size and shape
- Uncertainty caused by
  - ✓ inaccurate object segmentation
  - ✓ inaccurate assumptions on the object's properties
  - ✓ inaccurate camera calibration



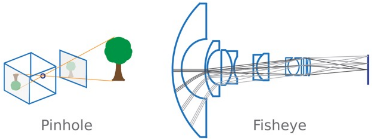
Source: MRS archive

Monocular distance estimate

- Common camera models:
  - ✓ Pinhole
  - ✓ Fisheye
  - ✓ Omnidirectional

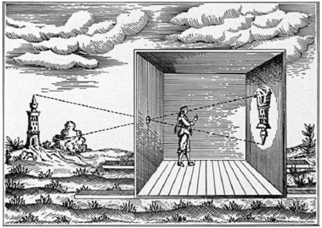


Source: Immersive Media



Source: Mathworks

- The pinhole camera model:
  - ✓ The classical principle of Camera Obscura
  - ✓ Computationally easy
  - ✓ Assumes no lens distortion (modelled separately)



Source: blackcreek.ca

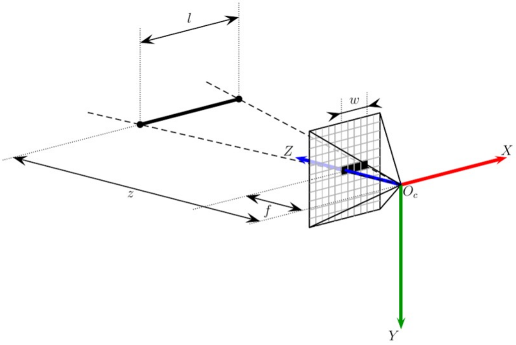
Monocular distance estimate

- The pinhole camera model assumes given:

  1) Focal distance f (in pixels): $f = f_x = f_y$
  2) Dimension of the object's image $w := x_{IMG2} - x_{IMG1}$ *[pixels]*
  3) Object is perpendicular to the optical axis Z with a horizontal dimension $l := x_2 - x_1$ *[m]*

- Estimated distance to the object is obtained as:

$$\frac{l}{z} = \frac{w}{f} \rightarrow z = \frac{lf}{w}$$



Monocular distance estimate

- For an approximate evaluation of the depth error, we can linearize the formula
- Error caused by an incorrect object segmentation or bounding box estimation:

$$\frac{dz}{dw} = -\frac{lf}{w^2} \quad \rightarrow \quad e_w = -\frac{lf}{w^2}\Delta w$$

  - ✓ Blur, blending with background, badly trained CNN, etc.
  - ✓ A pixel averages light intensity over its surface, cameras have a limited resolution
- Error caused by an incorrect assumption about the object's size:

$$\frac{dz}{dl} = \frac{f}{w} \quad \rightarrow \quad e_l = \frac{f}{w}\Delta l$$

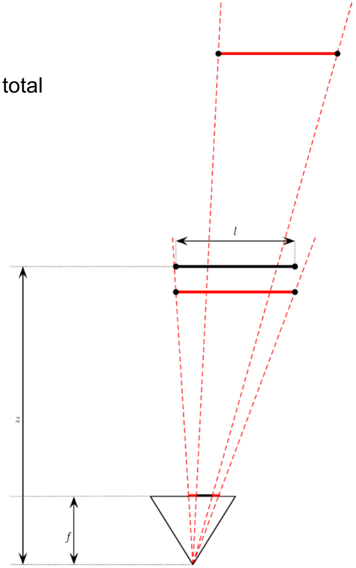
  - ✓ Inaccurate size measurement, imprecise assumptions about the object's shape, etc.
  - ✓ It is not always possible to measure physical dimensions of the object accurately
- Error caused by an incorrect focal length:

$$\frac{dz}{df} = \frac{l}{w} \quad \rightarrow \quad e_f = \frac{l}{w}\Delta f$$

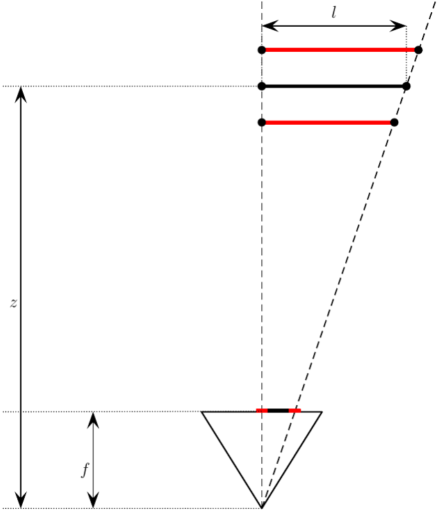
  - ✓ Inaccurate calibration of the projection model, wrong projection model, lens misalignment, mechanical damage of the lens assembly, etc.

Error from object image size

- Example 1 – **incorrect size estimation of the object's image:**
  - ✓ Error is $\pm 0.5px$ on each side of the bounding box → $\pm 1\mathrm{px}$ in total
  - ✓ Assuming an object at $z = 10m$ and camera parameters:
  - ✓ $f = 200px$ (for image resolution $W = 640 \rightarrow 116°$ HFoV)
  - ✓ $l = 1m$ (a relatively large micro UAV)
  - ✓ $w = 20px$
  - ✓ $\Delta w = \pm 1px$
  - ✓ $z = \frac{lf}{w} = 10m$
  - ✓ $w' = w \pm 2 \cdot 0.5px = w \pm 1px$
  - ✓ $e_w = \mp \frac{fl}{w^2} \Delta w = \mp \frac{200 \cdot 1}{20^2} \Delta w = \mp 0.5 \Delta w$
  - ✓ $\pm 1px \rightarrow e_{w1} = \pm 0.5\mathrm{m}$
  - ✓ The error is not symmetrical (we just linearized it)!

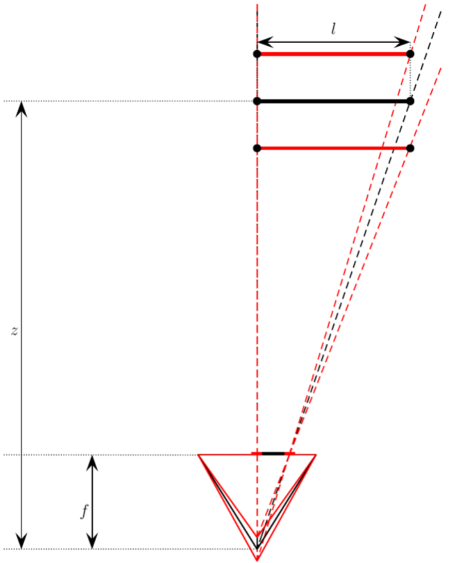


- Example 2 – **incorrect assumption about the object's physical size:**
  - ✓ Error is 5% of the object's true dimension
  - ✓ Same assumptions as in Example 1:
  - ✓ $f = 200px$
  - ✓ $l = 1m$
  - ✓ $w = 20px$
  - ✓ $\Delta l = 0.05l$
  - ✓ $z = \frac{lf}{w} = 10m$
  - ✓ $l' = 1.05l; \Delta l = 0.05l = 0.05m$
  - ✓ $e_l = \frac{f}{w}\Delta l = \frac{200}{20}\Delta l = 10\Delta l$
  - ✓ $\pm 0.05m \rightarrow e_{l5}= \pm 0.5\text{m}$
  - ✓ Note: It is a coincidence that the errors are the same, **not a rule!**



Error from camera calibration

- Example 3 – **incorrect focal length calibration:**
  - ✓ Error is 5% of the true f
  - ✓ Same assumptions as in Example 1 and 2:
  - ✓ $f = 200px$
  - ✓ $l = 1m$
  - ✓ $w = 20px$
  - ✓ $\Delta f = 0.05f$
  - ✓ $z = \frac{lf}{w} = 10m$
  - ✓ $f' = 1.05f; \Delta f = 0.05f = 10px$
  - ✓ $e_f = \frac{l}{w}\Delta f = \frac{1}{20}\Delta f = 0.05\Delta f$
  - ✓ $\pm 10px \rightarrow e_{f5} = \pm 0.5$m
  - ✓ Note: It is a coincidence that the errors are the same, **not a rule!**



Error from object orientation or shape

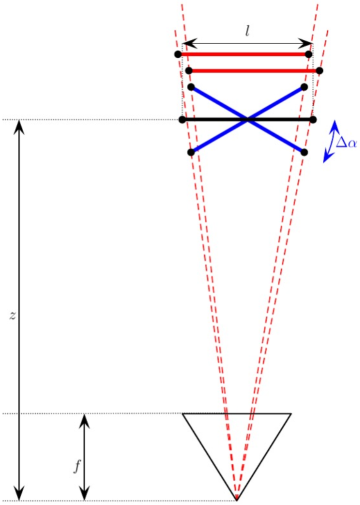
- Example 4 – **incorrect assumption about object's perpendicularity to Z**:
  - ✓ Objects are not perpendicular in real situations
  - ✓ Let assume (similarly as in the previous examples):
  - ✓ $f = 200px$
  - ✓ $l = 1m$
  - ✓ $w = 20px$
  - ✓ Object's center lies on the optical axis Z and $z \gg l$
  - ✓ $\Delta\alpha = 10°, 20°, 30°$

$$l'' \approx \cos(\Delta\alpha)\, l\, ; \quad z_l'' \approx \frac{l'' f}{w} = \cos(\Delta\alpha)\frac{lf}{w}$$

$$e_\alpha \approx (1 - \cos(\Delta\alpha))\frac{lf}{w} = (1 - \cos(\Delta\alpha))10m$$

$\Delta\alpha$	e_α
$\pm10°$	-0.15m
$\pm20°$	-0.60m
$\pm30°$	-1.34m

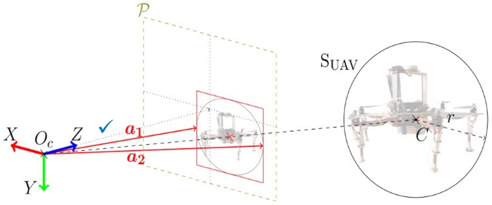
- ✓ $\cos(\Delta\alpha) \leq 1$ → the object is **never closer than estimated**

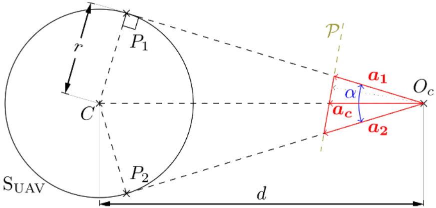


Distance estimation for a spherical object

- Assuming spherical objects - to tackle the perpendicularity issue
  - ✓ Complicated distance estimation formula
- Assumptions:
  - ✓ Spherical object (not a flat line) with a known radius r
  - ✓ Known bearing vectors a_1 and a_2 corresponding to the edges of the object's bounding box
  - ✓ Removing assumptions about the projection model – to simplify the math
  - ✓ Estimating the distance d instead of the depth z - again to simplify the math
  - ✓ Using the distance to estimate the 3D position given the bearing vector a_c

    Applying the triangle ⊿ $O_C C P_1$: $\alpha = \mathrm{acos}\left(\frac{a_1 \cdot a_2}{\|a_1\|\|a_2\|}\right)$; $d = \frac{r}{\sin\left(\frac{\alpha}{2}\right)}$

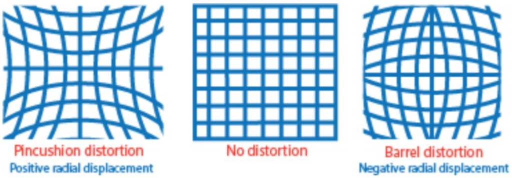


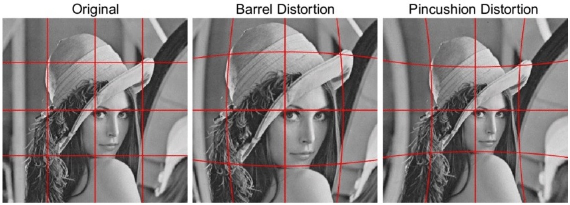


Other camera models

- Real cameras typically don't conform to the pinhole model due to distortion
  - ✓ Barrel
  - ✓ Pincushion

- Present in omnidirectional and fisheye cameras - useful for MRS with UAVs: scene overview without an extra movement





Source: Mathworks

Conclusion

- Motivation
  - ✓ Mutual/relative localization is a key technique enabling safe close coordination and interaction among the UAVs sharing the same workspace
- Marker-based visual relative localization
  - ✓ Reliable approach (mainly active markers – e.g. LEDs)
  - ✓ Enables difficult-to-jam communication (secured systems of large MRS – e.g. urban mobility)
- Marker-less visual relative localization
  - ✓ General usage – also none-cooperating UAVs
  - ✓ Sense and avoid mechanisms
- Multi-target tracking and filtration
  - ✓ Important technique for real-world systems with sensory uncertainty
  - ✓ Enables estimating states not observable by the localization method or even future behavior of the observed UAVs
  - ✓ Important for large teams of anonymous UAVs – distinguishing between particles/UAVs

References

[1] Xu, Hao, et al., “Decentralized visual-inertial-UWB fusion for relative state estimation of aerial swarm”, *IEEE ICRA* 2020.

[2] L. Teixeira, F. Maffra, M. Moos and M. Chli, “VI-RPE: Visual-Inertial Relative Pose Estimation for Aerial Vehicles”, *IEEE RAL, vol. 3, no. 4, pp. 2770-2777, 2018.*

[3] Faessler, M., Mueggler, E., Schwabe, K., & Scaramuzza, D., “A monocular pose estimation system based on infrared leds”, *IEEE ICRA, 2014.*

[4] J. Li, D. H. Ye, … C. Bouman, “Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs)”, *IROS, 2016.*

[5] H. Li and F. Nashashibi, “Multi-vehicle cooperative localization using indirect vehicle-to-vehicle relative pose estimation”, *CVES, 2012.*

[6] M Vrba, D Heřt and M Saska, “Onboard Marker-Less Detection and Localization of Non-Cooperating Drones for Their Safe Interception by an Autonomous Aerial System.”, *IEEE RAL, 2019.*