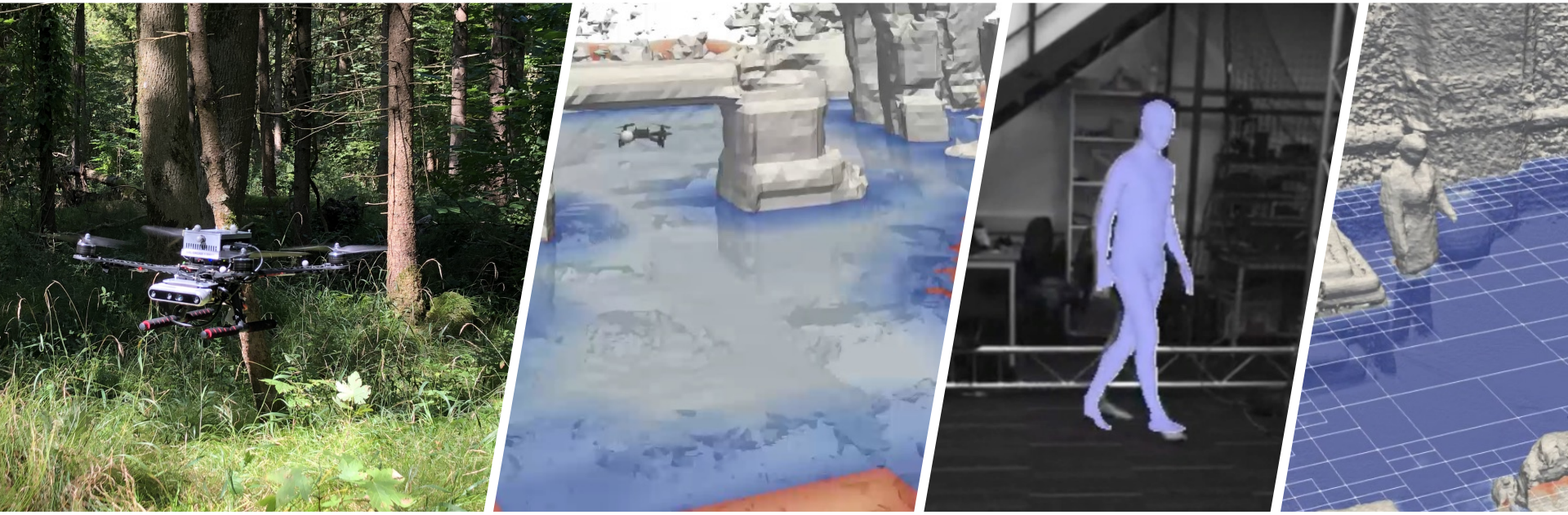
From Perception to Navigation and Control for Fully Autonomous Drones in Cluttered, Open-Ended Environments

Stefan Leutenegger, TU Munich (and Imperial College London)

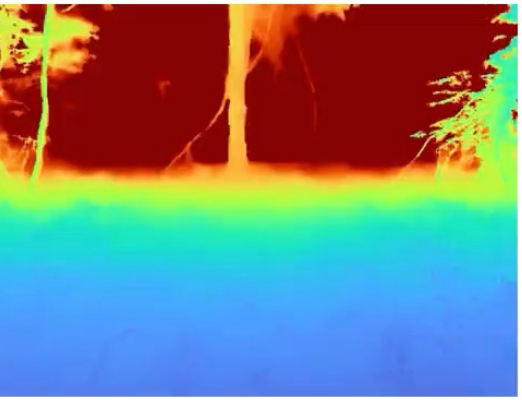
AUSROS'24 Deep Dive, 3rd July 2024



Example: Drone-based Mapping for Forestry [Subm.]

S Barbas Laina, S Boche, S Papatheodorou, D Tzoumanikas, S Schaefer, H Chen, S Leutenegger

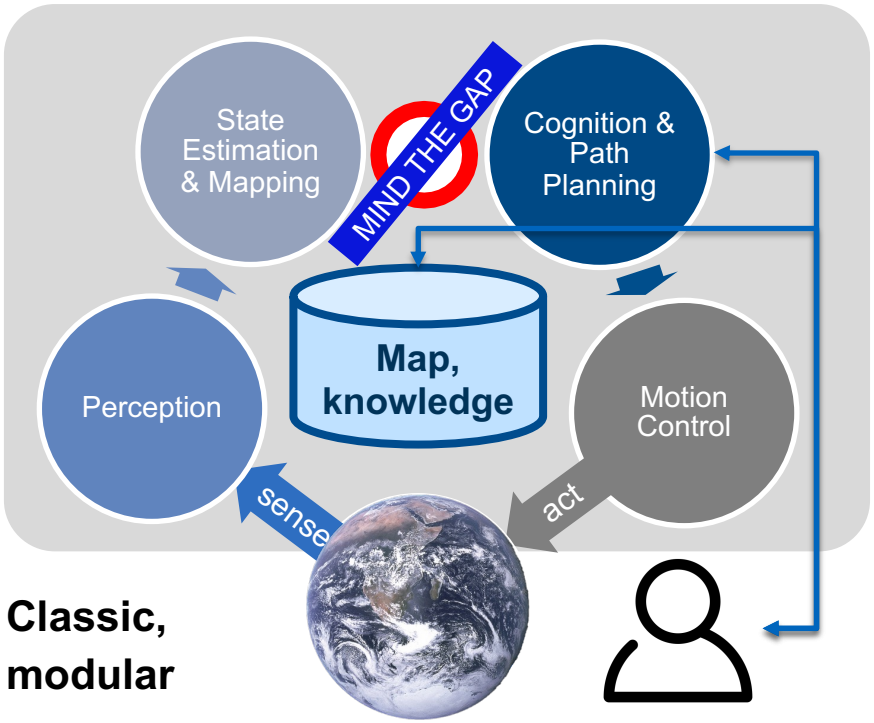
MAV IR (15 Hz)



MAV depth (5 Hz)

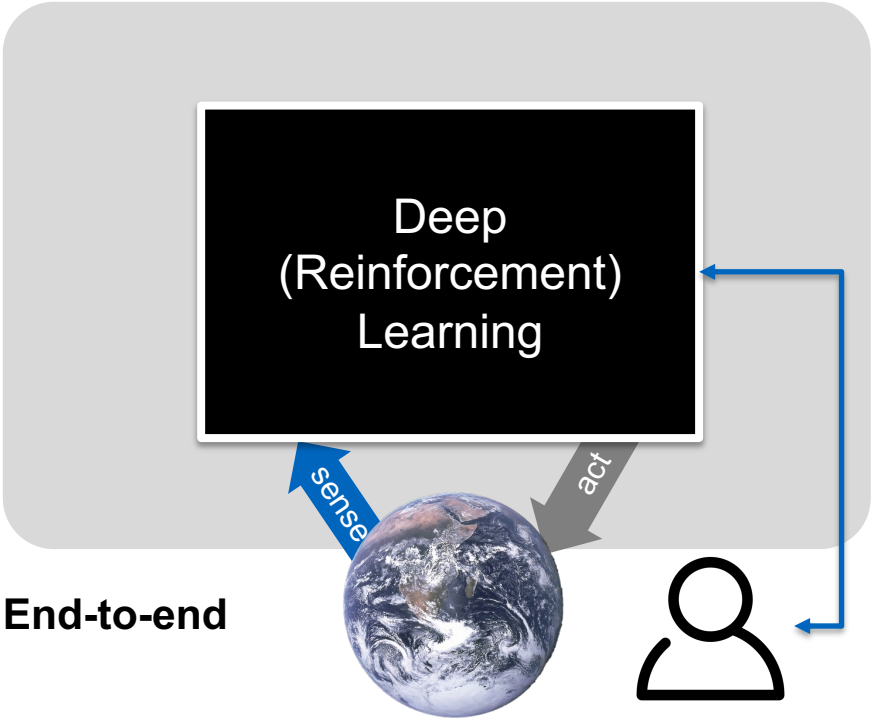
1x

How Mobile Robots Work and Improve



Classic, modular

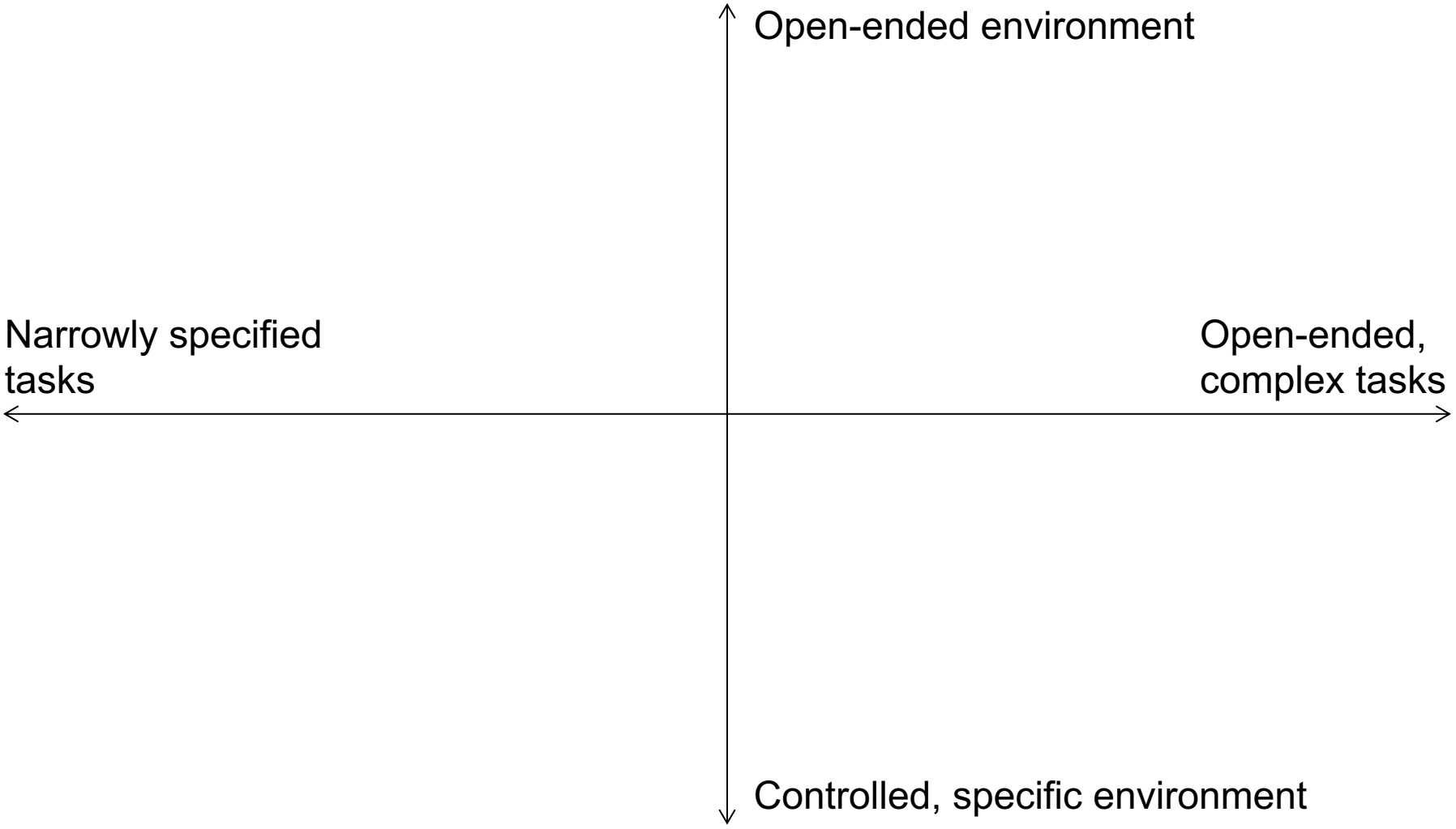
- **Provides introspection, shared representation of map/knowledge**
- **Bottlenecks at module boundaries**
- **Bridge the gap: tightly integrate Deep Learning in modules**



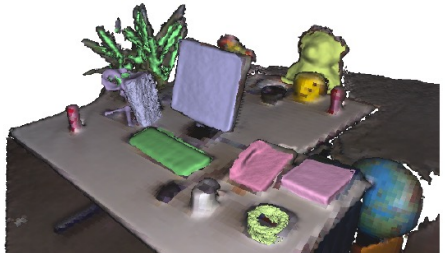
End-to-end

- **Self-emergent, not handcrafted**
- **Black-box: no guarantees, very difficult to introspect**
- **Not data efficient enough for training of complex tasks in the real world**

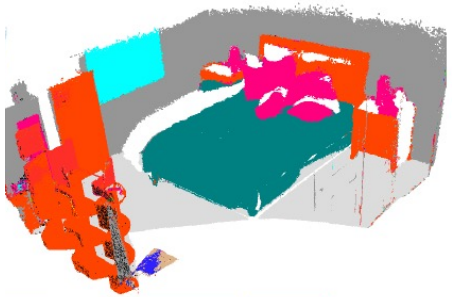
Readiness of (Mobile) Robots in 2024



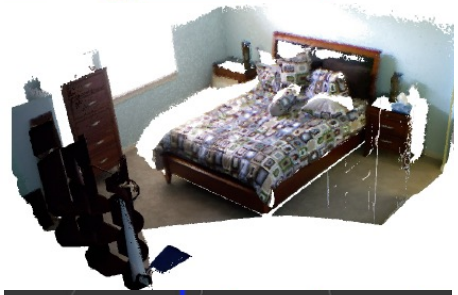
Levels from SLAM to Spatial AI



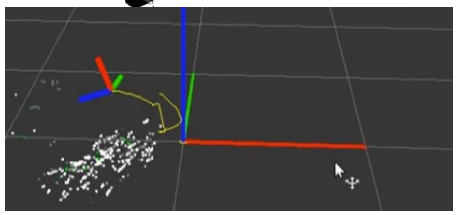
- **Object-level** and **dynamic maps** e.g. [MID-Fusion]
  - Advanced physical interaction robot-environment
  - More advanced human robot interaction



- **Semantic maps:** 3D scene understanding e.g. [SemanticFusion]
  - Advanced task planning
  - Advanced human robot interaction



- **Dense mapping** e.g. [ElasticFusion, supereight 2]
  - Motion planning
  - Collision avoidance



- Sparse **Visual-Inertial** Localisation and Mapping e.g. [OKVIS, OKVIS2]
  - Robust & accurate motion tracking for position control

Stefan Leutenegger

The Vision

Hand me the box!

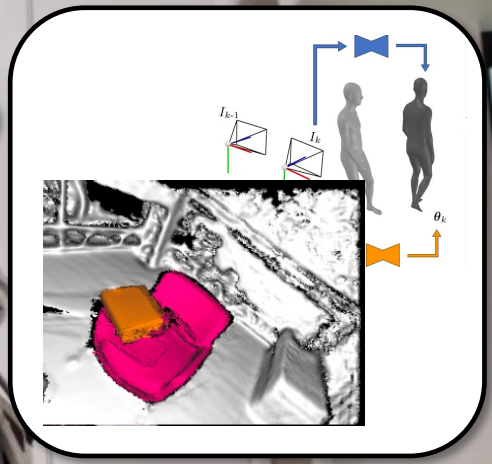
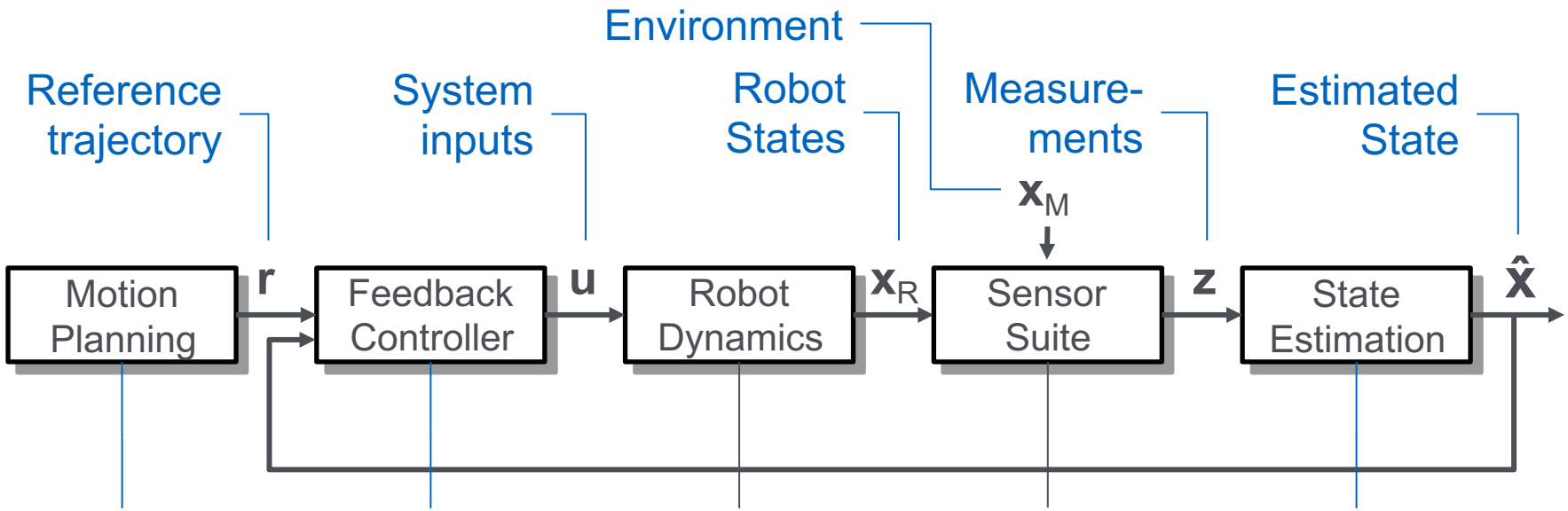


Image credit: Boston Dynamics

Problem Setting



What trajectory that is safe do we want the robot to follow?

What commands should we send to the robot so it follows the reference trajectory?

How does the robot react to inputs given initial states?

How are measure-ments are generated given the robot states?

What is the best explan-ation for the robot states given the measure-ments?

From Multi-Sensor SLAM to Spatial AI

SMART ROBOTICS LAB

Imperial College London

Technische Universität München

Probabilistic Estimation

Measurements $\mathbf{z}$ (and odometry $\mathbf{u}$) are modeled as samples from a **distribution** $p(\mathbf{z}|\mathbf{x})$, i.e. given the variables $\mathbf{x}$ (here: robot states $\mathbf{x}_\mathrm{R}$ plus possibly the map $\mathbf{x}_\mathrm{M}$).

What values for variables $\mathbf{x}$ best explain all the measurements $\mathbf{z}$ (and odometry $\mathbf{u}$): Maximum Likelihood (or Maximum a Posteriori with some prior)

Localisation:

$\mathbf{x}^*_{\mathrm{R},1:k} = \operatorname{argmax} p\left(\mathbf{x}_{\mathrm{R},1:k} \middle| \mathbf{x}_\mathrm{M}, \mathbf{z}_{1:k}, \mathbf{u}_{1:k}\right)$, (or only of $\mathbf{x}_{\mathrm{R},k}$ in recursive formulation)

Simultaneous Localisation and Mapping (SLAM):

$\{\mathbf{x}^*_{\mathrm{R},1:k}, \mathbf{x}^*_\mathrm{M}\} = \operatorname{argmax} p\left(\mathbf{x}_{\mathrm{R},1:k}, \mathbf{x}_\mathrm{M} \middle| \mathbf{z}_{1:k}, \mathbf{u}_{1:k}\right)$, (or only of $\mathbf{x}_{\mathrm{R},k}$ in recursive formul.)

Mapping:

$\mathbf{x}^*_\mathrm{M} = \operatorname{argmax} p\left(\mathbf{x}_\mathrm{M} \middle| \mathbf{x}_{\mathrm{R},1:k}, \mathbf{z}_{1:k}\right)$.

To Be Estimated: Robot States and Environment

Robot state

Typical components:

- Position
- Orientation
- Velocity
- Sensor-internal states
- …

Characteristics:

- Typically *time-varying*

Map – the environment

Popular representations:
3D points (sparse)
3D pointcloud/surfels (dense)
Occupancy grid / voxels
Triangulated mesh
…

Characteristics:
Typically, a large part is assumed to be *static*

Robot State

An example state vector:

$$\mathbf{x}_{\mathrm{R}}^T = \left[{}_W\mathbf{t}_S^T, \mathbf{q}_{WS}^T, {}_W\mathbf{v}^T, {}_S\mathbf{t}_{C_1}^T, \mathbf{q}_{SC_1}^T, {}_S\mathbf{t}_{C_2}^T, \mathbf{q}_{SC_2}^T, \mathbf{b}_g^T, \mathbf{b}_a^T \right]$$

Position and orientation of frame S origin w.r.t. in W

Velocity of frame S origin expressed in W

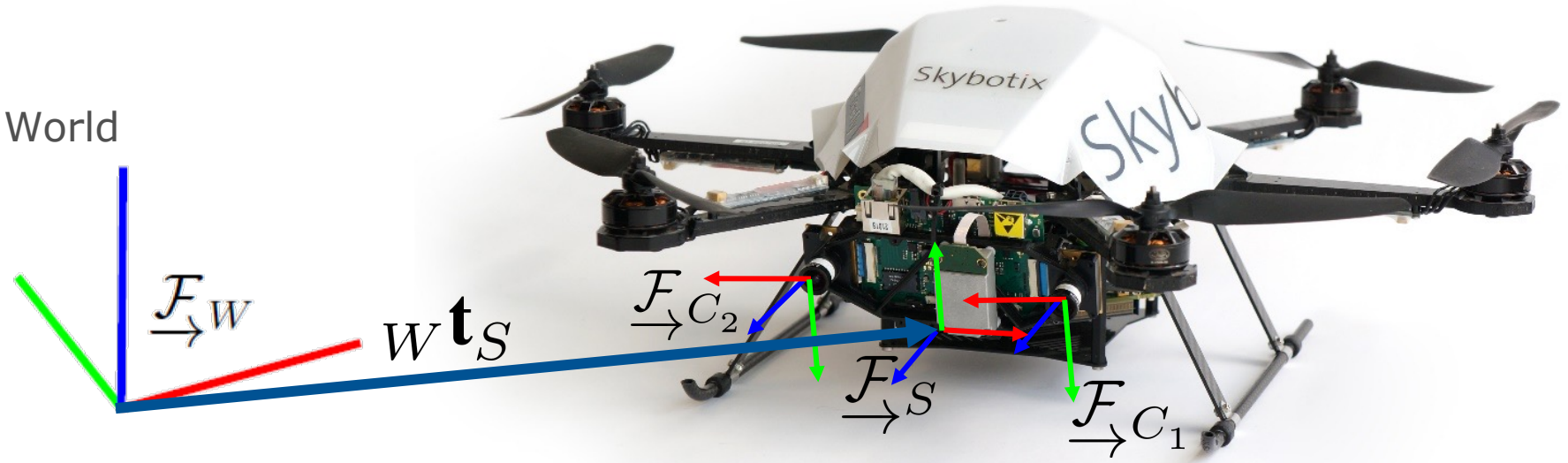
Physical robot states

Poses of cameras 1 and 2 relative to frame S

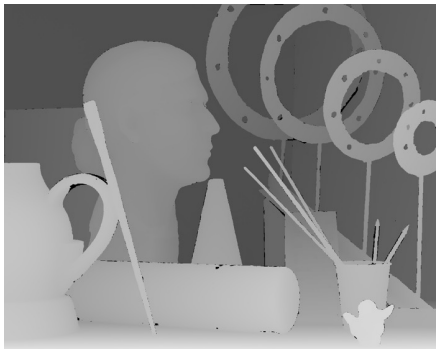
Sensor extrinsics (here: camera)

IMU gyro and accelerometer biases

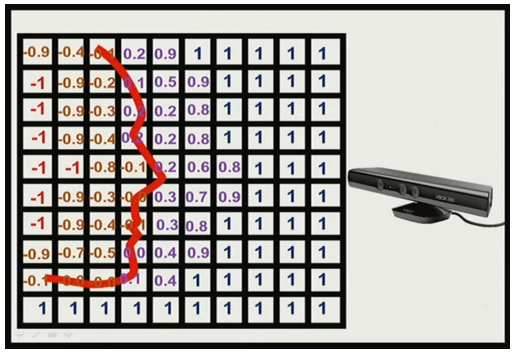
Sensor intrinsics (here: IMU)



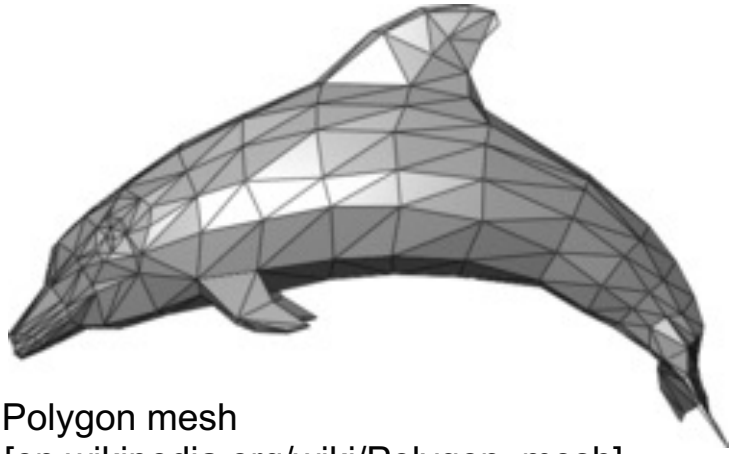
Map Representations



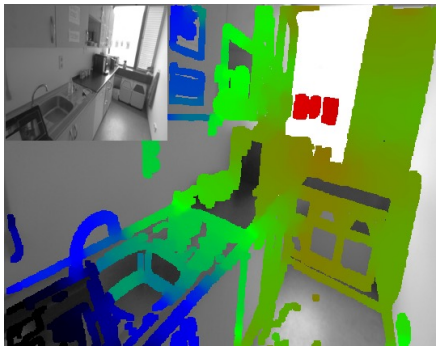
Depth maps
[vision.middlebury.edu]



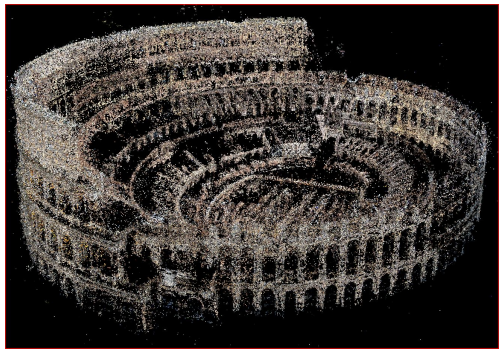
Truncated Signed Distance Function [pointclouds.org]



Polygon mesh
[en.wikipedia.org/wiki/Polygon_mesh]



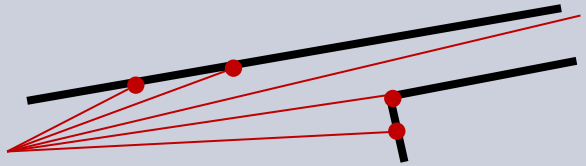
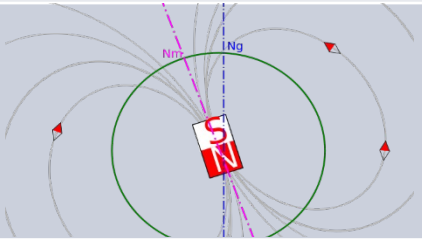
Semi-dense depth maps
[vision.in.tum.de]



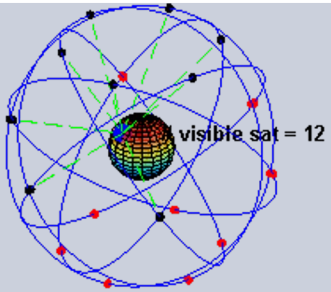
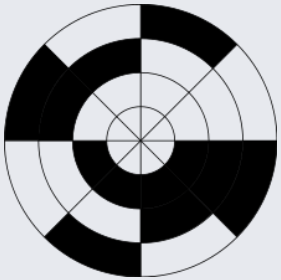
Point clouds (here: sparse)
[grail.cs.washington.edu]

Surfel maps
[wp.doc.ic.ac.uk/thefutureofslam]

Typical Sensors – Exteroceptive

Sensor	Measurement	
Laser Scanner	3D points	
Camera	(Colour) image (RGB-D: with depth!)	
Magnetometer	3D magnetic field	
Pressure sensor	Air pressure (altitude / airspeed)	

Typical Sensors – Proprioceptive

Sensor	Measurement	
GPS	pseudo-ranges (position)	
Encoders	Joint / wheel angles	
Inertial Measurement Unit (IMU)	Rotation rates and accelerations (with caution: orientation)	

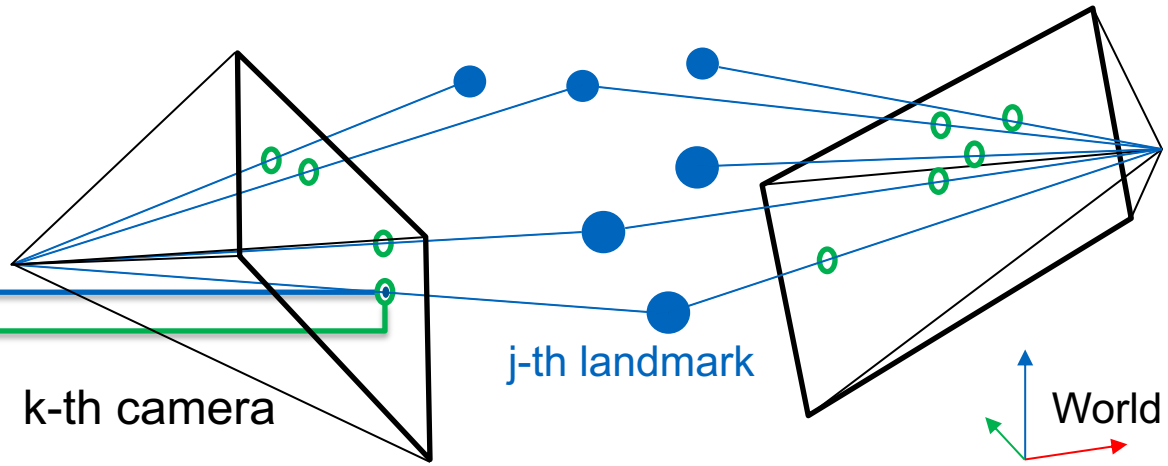
(Sparse) Probabilistic Visual-Inertial SLAM

Reprojection error:

$$\mathbf{e}_{\mathrm{r}}^{j,k}({}_W\mathbf{t}_{C_k}, \mathbf{q}_{WC_k}, {}_W\boldsymbol{l}_j) = \tilde{\mathbf{u}} - \pi(\boldsymbol{T}_{C_kW}\, {}_W\boldsymbol{l}_j),$$

… and assume:

$$\mathbf{e}_{j,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathrm{r}}^{j,k}).$$



Minimize $c(\mathbf{x}) = \frac{1}{2}\sum_{i,j,k} \left|\left|\mathbf{e}_{\mathrm{r}}^{i,j,k}\right|\right|^2_{\mathbf{W}_{\mathrm{r}}} + \frac{1}{2}\sum_{k} \left|\left|\mathbf{e}_{\mathrm{s}}^{k}\right|\right|^2_{\mathbf{W}_{\mathrm{s}}} + \frac{1}{2}\sum\sum \left|\left|\mathbf{e}_{\mathrm{p}}^{r,c}\right|\right|^2_{\mathbf{W}_{\mathrm{p}}} + \frac{1}{2}\sum_{l} \left|\left|\mathbf{e}_{\mathrm{g}}^{l}\right|\right|^2_{\mathbf{W}_{\mathrm{g}}}$

Reprojection err. | IMU (preintegrated) err. | Relative pose err. | (GNSS) position err.

[OKVIS2: Stefan Leutenegger, arXiv'22] **HILTI Challenge '22 VI winner**

[OKVIS2-GNSS: Simon Boche & Stefan Leutenegger IROS'22]

Simon B.

Imperial College London

Technische Universität München

Occupancy Mapping with Distance Sensor

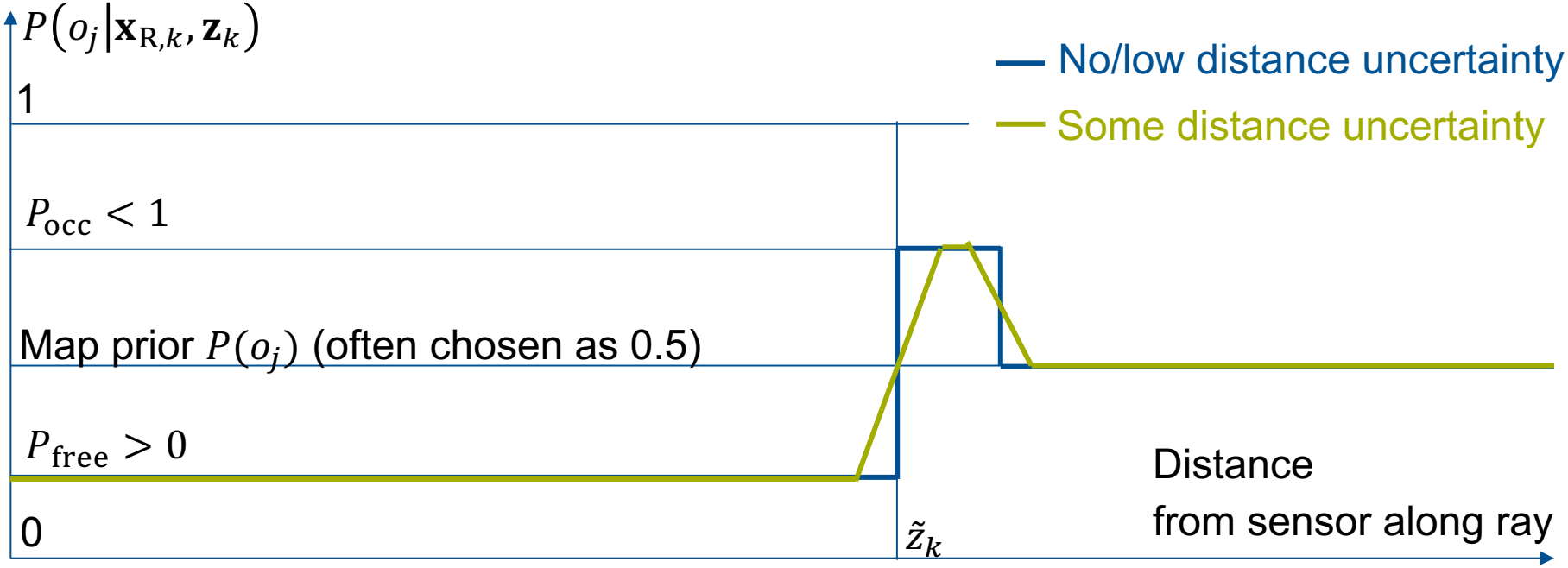
To make a practical implementation even more trivial, we can use **Log-Odds**:

$$l(a) \coloneqq \log(Odds(a))$$

So now we get for the recursive update a simple addition:

$$l\left(o_j \middle| \mathbf{x}_{\mathrm{R},1:k}, \mathbf{z}_{1:k}\right) = l\left(o_j \middle| \mathbf{x}_{\mathrm{R},k}, \mathbf{z}_k\right) + l\left(o_j \middle| \mathbf{x}_{\mathrm{R},1:k-1}, \mathbf{z}_{1:k-1}\right) - l(o_j)$$

What is left to define is the **inverse sensor model** in Log-Odds form, $l\left(o_j \middle| \mathbf{x}_{\mathrm{R},k}, \mathbf{z}_k\right)$:

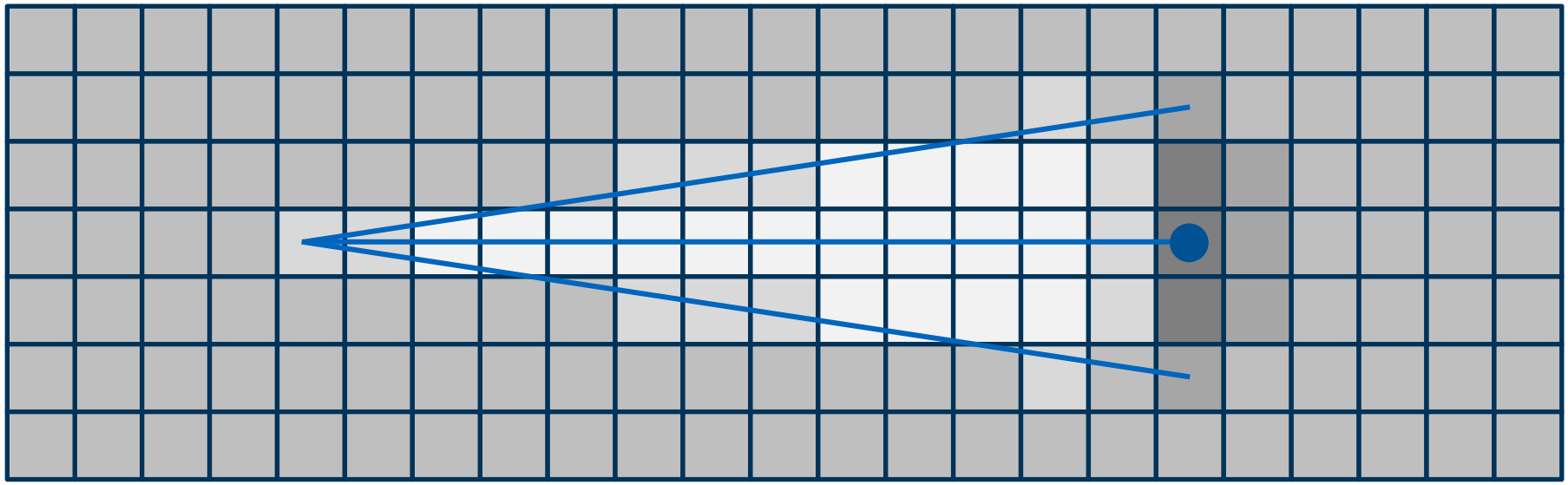


Occupancy Mapping with Distance Sensor

In the **2D inverse sensor model**, depending on sensor/resolution considerations, we may want to consider an opening angle (larger for sonar, very small for LiDAR).

Notes:

- In practice, we will **only have to update the cells within this sensor field** (i.e. anything deviating from the map prior)
- We will want to **bound the occupancy values** to a minimum/maximum to avoid overconfidence (and account for deviations from the static world assumption)



Memory and Access Efficiency in 3D Volumes

Using a dense 3D grid, e.g. as array, memory will run out soon.

Efficient alternatives:

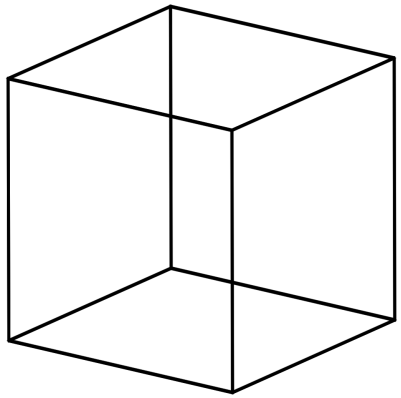
- Use a **hashtable** (e.g. [1])
  - 3D position keys
  - Values are **contiguous voxel blocks**
- Use an **octree** (e.g. [2, 3]):
  - Voxels stored in **pointer-based** tree with 8 children per node
  - Leaves as **contiguous voxel blocks**
  - Tree **dynamically allocated** as observed map grows

root level

node level 1

node level 2

voxel block level



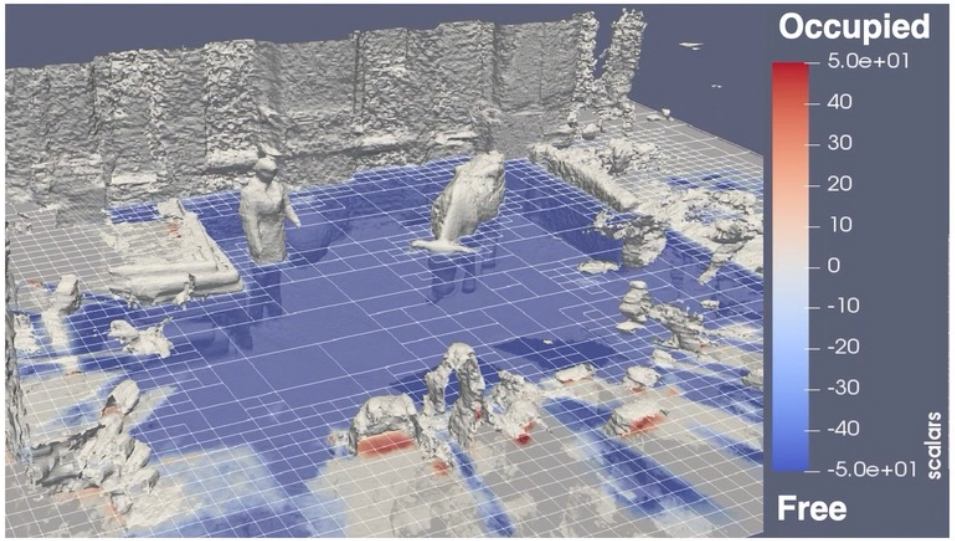
[1] Oleynikova, Helen, et al. "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning." *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.
[2] Hornung, Armin, et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees." *Autonomous robots* 34.3 (2013): 189-206.
[3] Funk, Nils, et al. "Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning." *IEEE Robotics and Automation Letters* 6.2 (2021): 3553-3560.

3D Occupancy Mapping [ICRA'21]

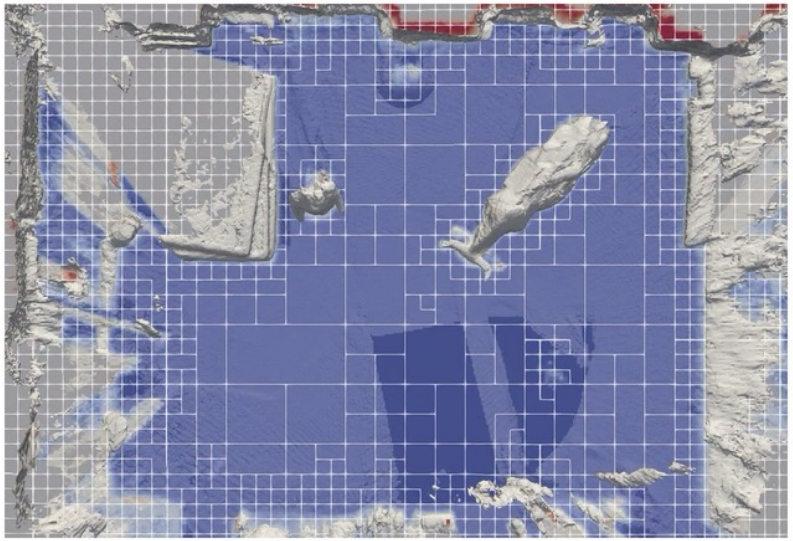
Nils Funk, Juan Tarrio, Sotiris Papatheodorou, Masha Popovic, Pablo Alcantarilla, Stefan Leutenegger

Map Visualisation

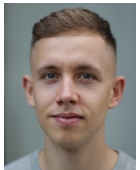
Side view



Top view



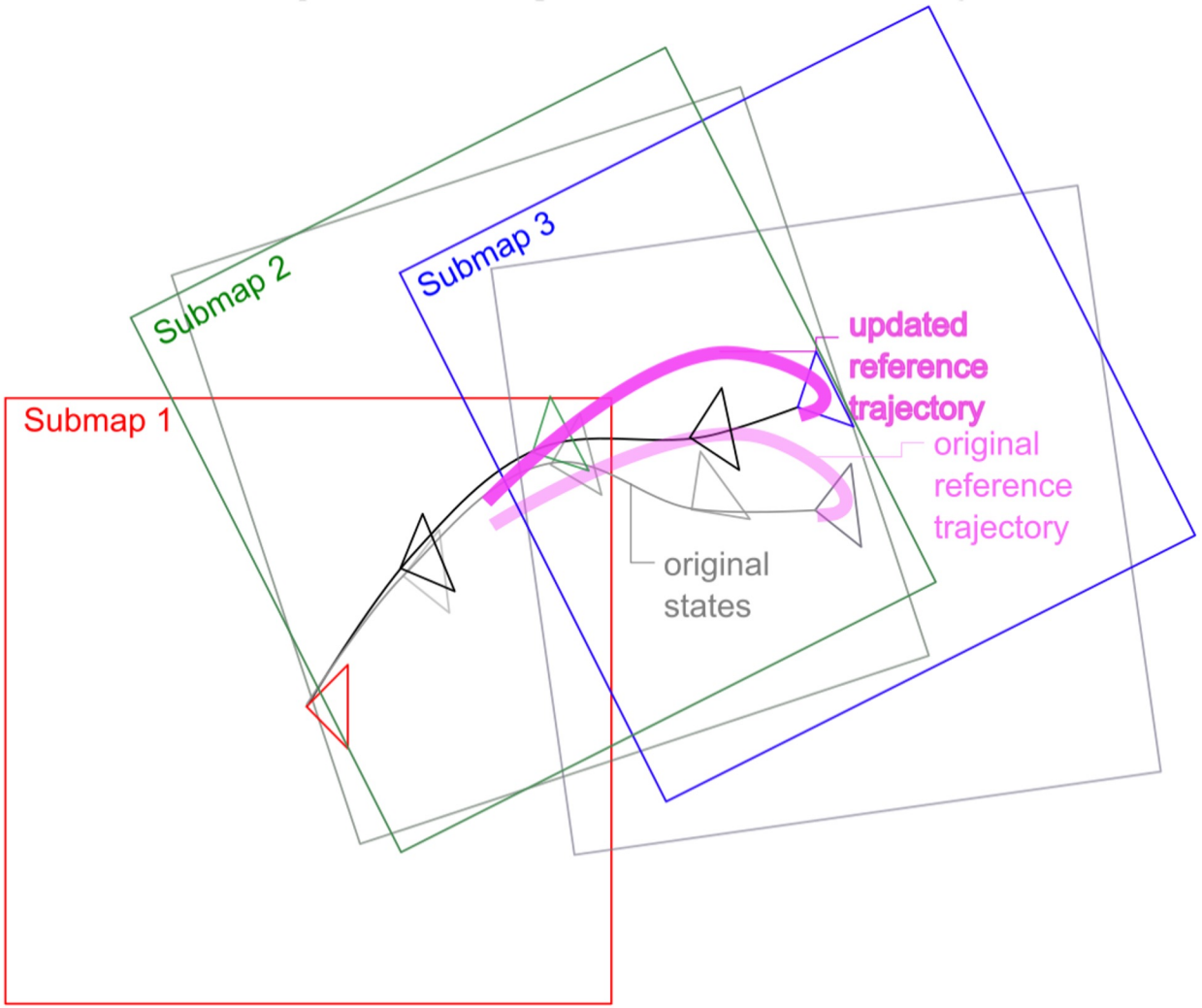
Explicit free-space representation encoding in a hierarchical way, supporting fast collision checking as crucial in robotic path planning and collision avoidance.



Nils

https://youtu.be/XlnO3GBNPvc

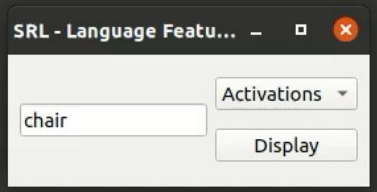
Occupancy Submaps Anchored to VI-SLAM

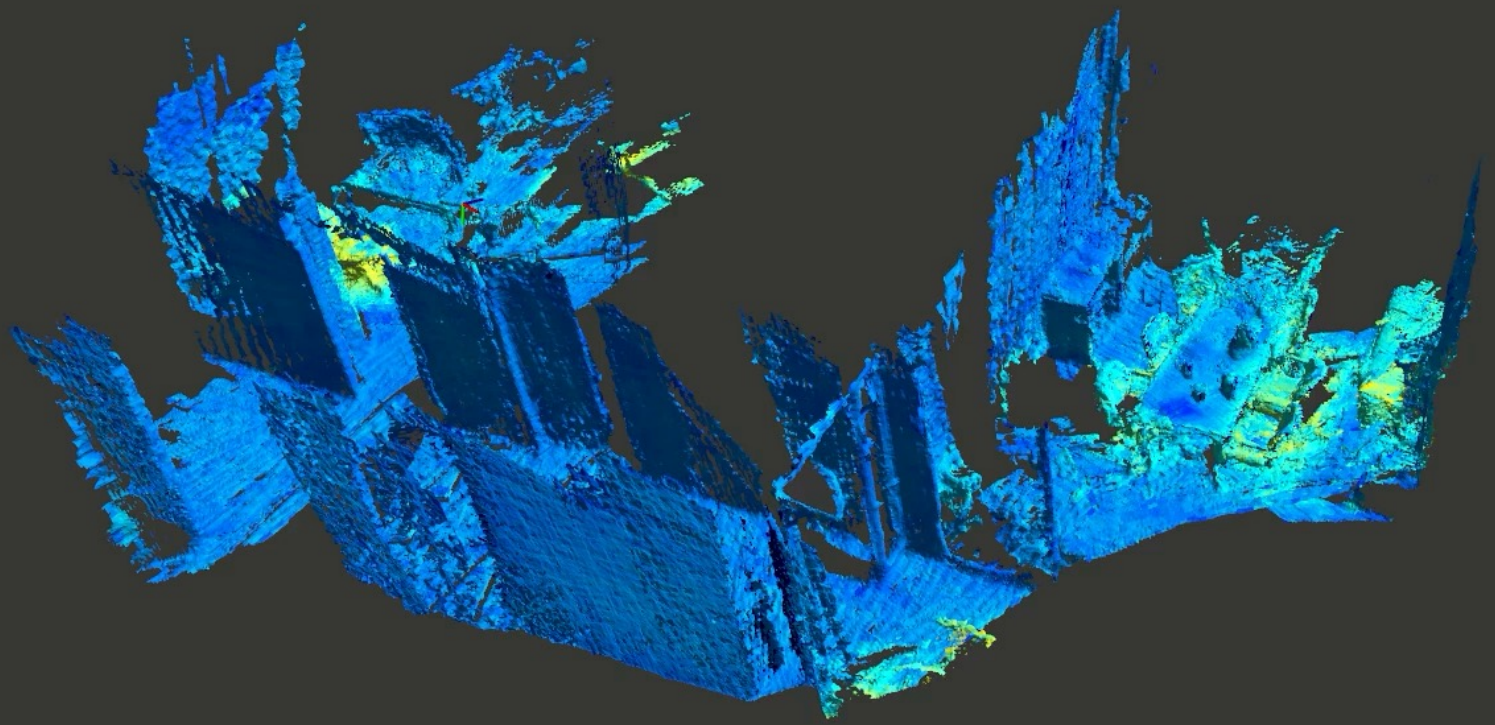


Stefan Leutenegger

Submaps Anchored to VI-SLAM

S Barbas Laina, S Boche, S Papatheodorou, S Schaefer, H Chen, S Leutenegger



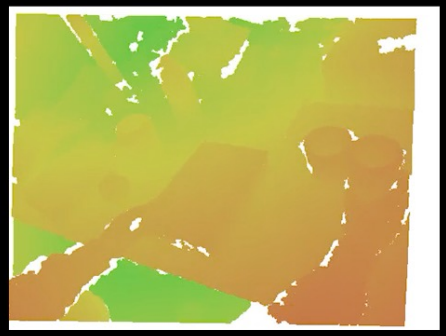


MID-Fusion [ICRA'19]

Binbin Xu, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, Stefan Leutenegger

Given RGB-D inputs:

RGB

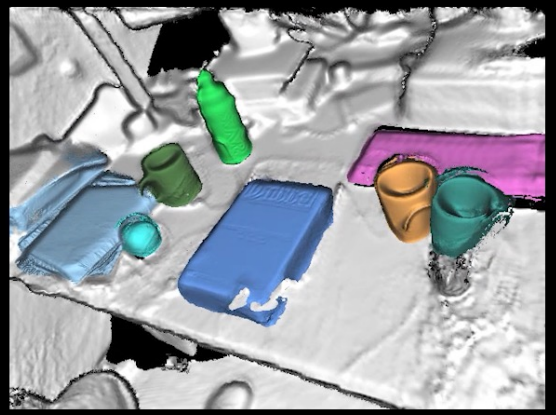


Depth

Output:Object-level dynamic volumetric map

-> track camera and each object pose
-> refine geometric, semantic, motion, and foreground property for each object

3D reconstruction

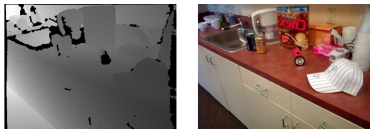


3D label map

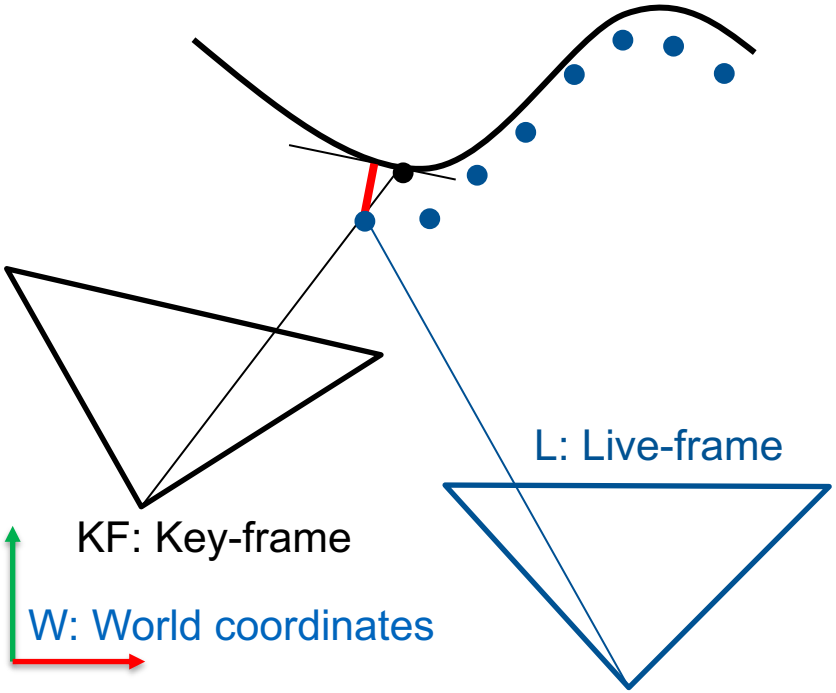
Binbin

https://youtu.be/gturboNl9gg

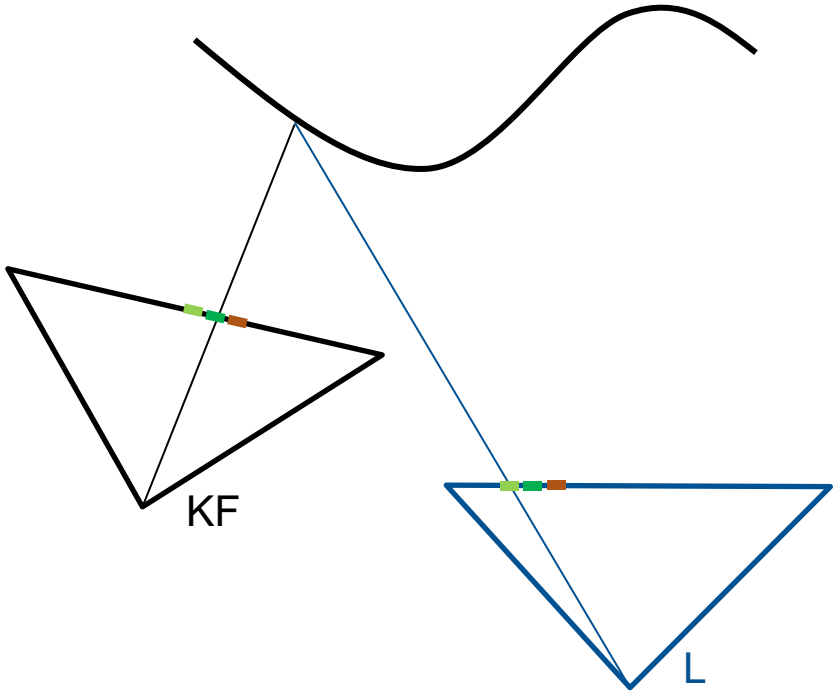
Dense SLAM: Tracking



ICP error (geometric distance):



Photometric error (colour difference):



$$\mathbf{u}_{\mathrm{KF}} = \pi(\boldsymbol{T}_{WC_{\mathrm{KF}}}^{-1}\, \boldsymbol{T}_{WC_{\mathrm{L}}}\, (\pi^{-1}(\mathbf{u}_{\mathrm{L}}, D_{\mathrm{L}}[\mathbf{u}_{\mathrm{L}}])))$$

$$e_{\mathrm{ICP}} = {}_{W}\mathbf{n}_{\mathrm{KF}}[\mathbf{u}_{\mathrm{KF}}] \cdot (\boldsymbol{T}_{WC_{\mathrm{L}}}\, {}_{C_{\mathrm{L}}}\mathbf{v}_{\mathrm{L}}[\mathbf{u}_{\mathrm{L}}] - {}_{W}\mathbf{v}_{\mathrm{KF}}[\mathbf{u}_{\mathrm{KF}}])$$

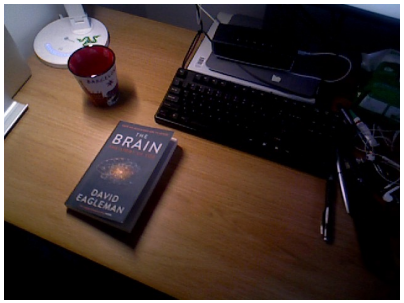
$$\mathbf{u}_{\mathrm{L}} = \pi(\boldsymbol{T}_{WC_{\mathrm{L}}}^{-1}\, \boldsymbol{T}_{WC_{\mathrm{KF}}}\, (\pi^{-1}(\mathbf{u}_{\mathrm{KF}}, D_{\mathrm{KF}}[\mathbf{u}_{\mathrm{KF}}])))$$

$$e_{\mathrm{p}} = I_{\mathrm{KF}}[\mathbf{u}_{\mathrm{KF}}] - I_{\mathrm{L}}[\mathbf{u}_{\mathrm{L}}]$$

Deep Probabilistic Feature-metric Tracking [RA-L'21]

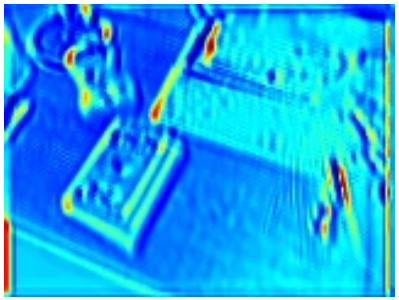
Binbin Xu, Andrew J. Davison, Stefan Leutenegger

Coupled **two-view feature and uncertainty encoders** predict pixel-wise feature and uncertainty maps

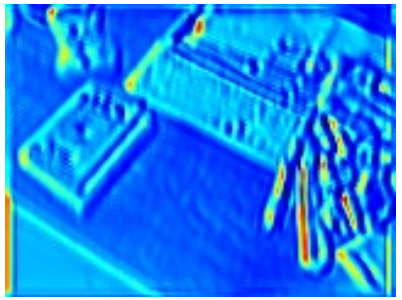


$\mathbf{I}_A$

$\mathbf{I}_B$



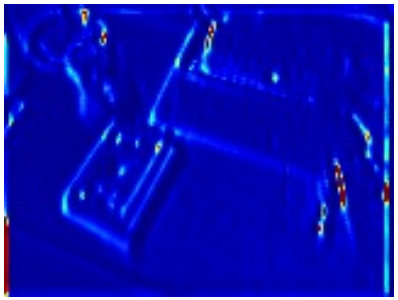
$\mathbf{F}_A$



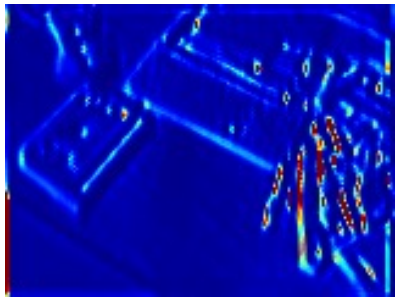
$\mathbf{F}_B$

Proposed probabilistic **feature-metric** residual:

$$\mathbf{e}_{\text{pfm}} = \frac{\mathbf{F}_A[\mathbf{u}_A] - \mathbf{F}_B[\mathbf{u}_B]}{\sqrt{\sigma_A^2[\mathbf{u}_A] + \sigma_B^2[\mathbf{u}_B]}}$$



σ_A



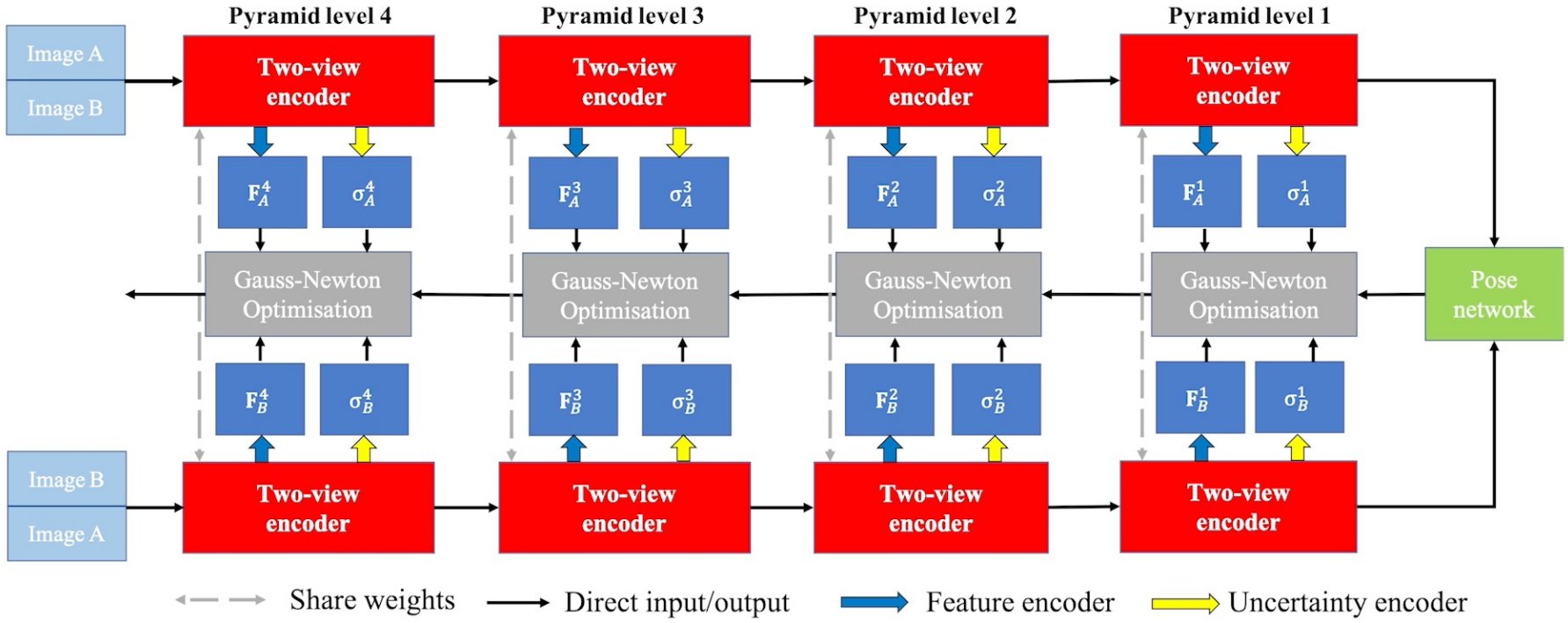
σ_B

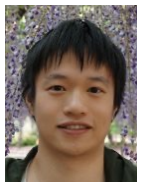
Binbin

Deep Probabilistic Feature-metric Tracking [RA-L'21]

Binbin Xu, Andrew J. Davison, Stefan Leutenegger

Network architecture:

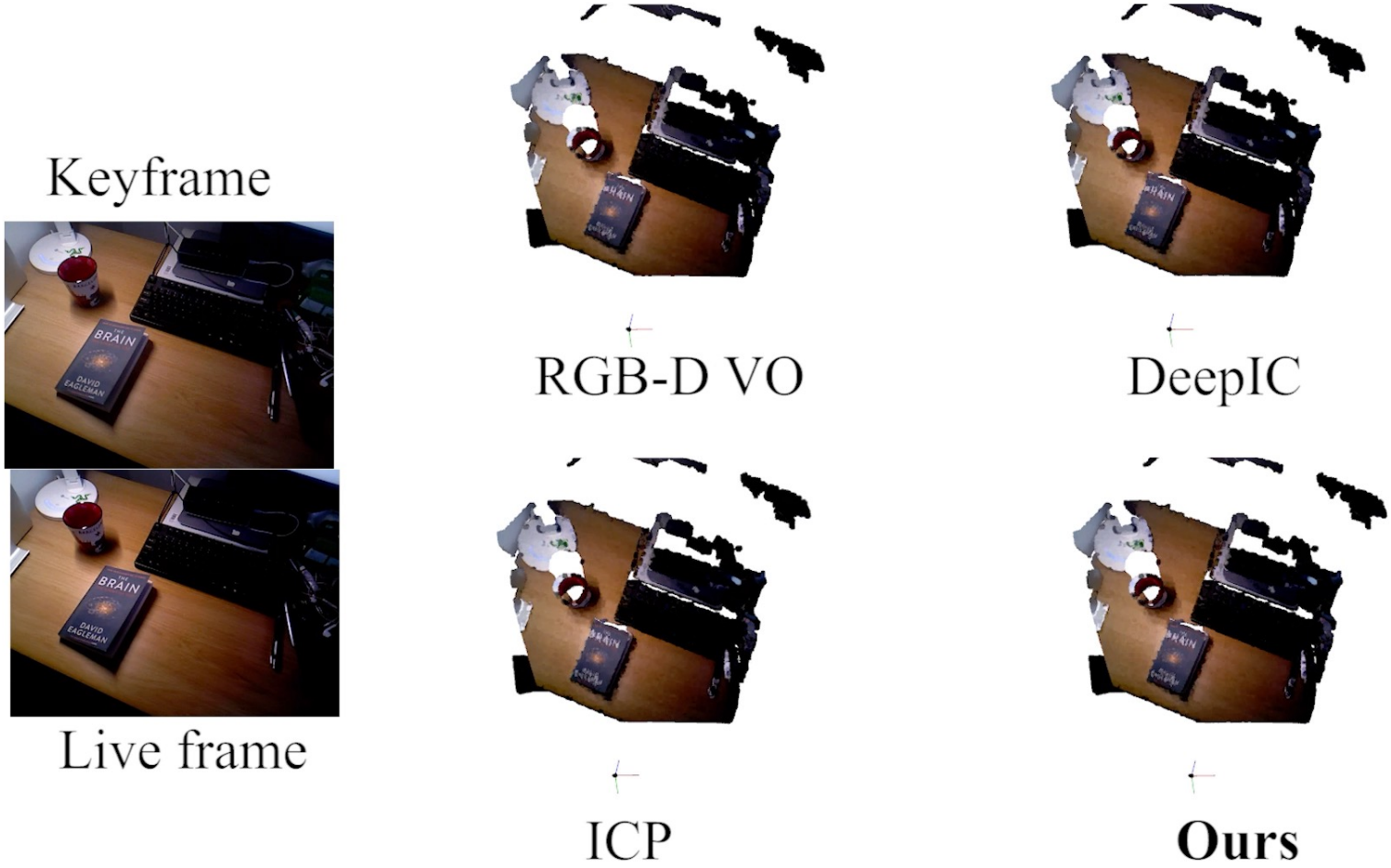


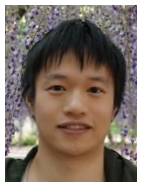


Binbin https://youtu.be/6pMosl6ZAPE

Deep Probabilistic Feature-metric Tracking [RA-L'21]

Binbin Xu, Andrew J. Davison, Stefan Leutenegger

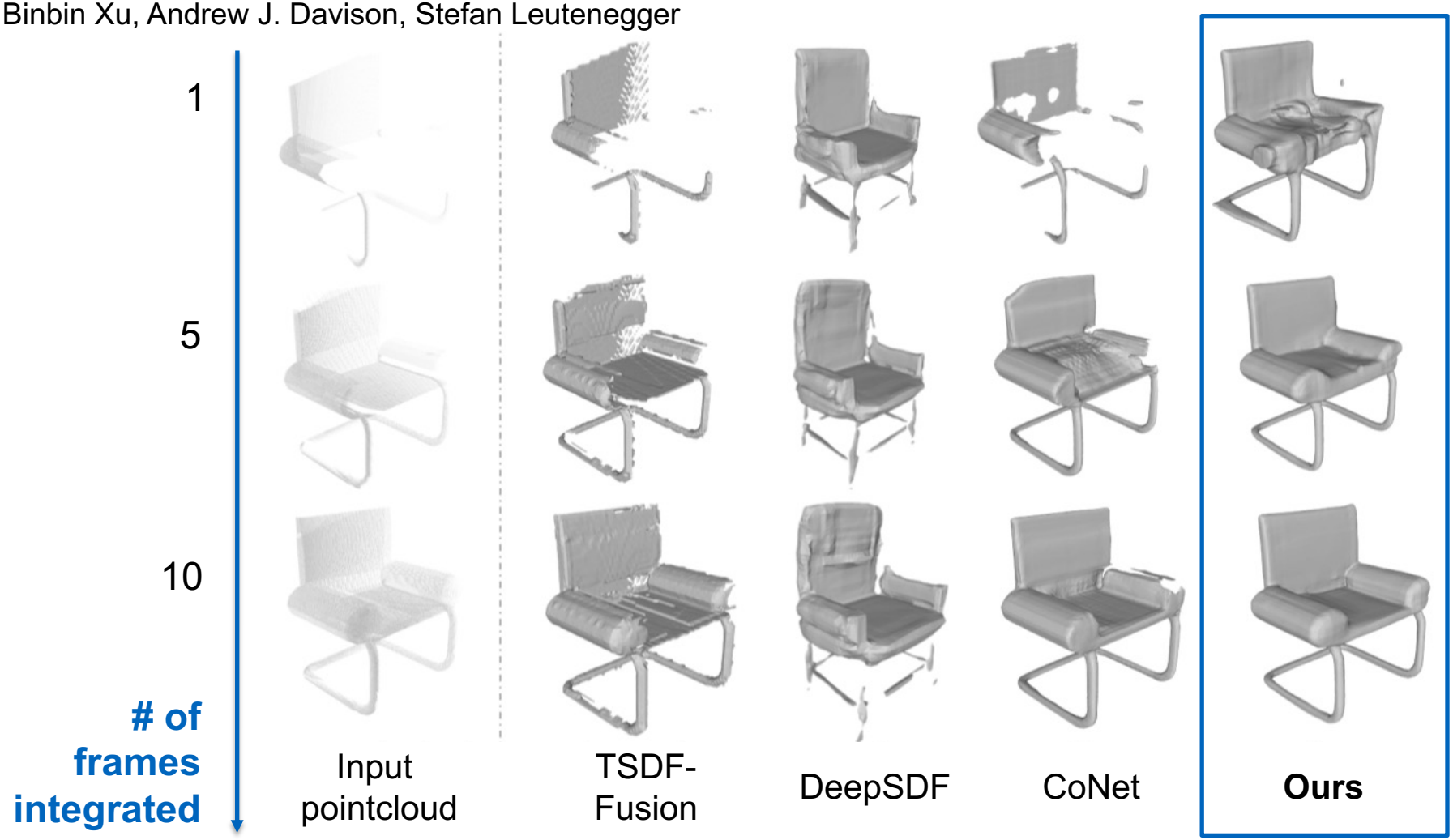


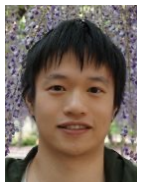


Binbin https://youtu.be/6pMosl6ZAPE

Completion of Dynamic Objects [IROS'22]

Binbin Xu, Andrew J. Davison, Stefan Leutenegger

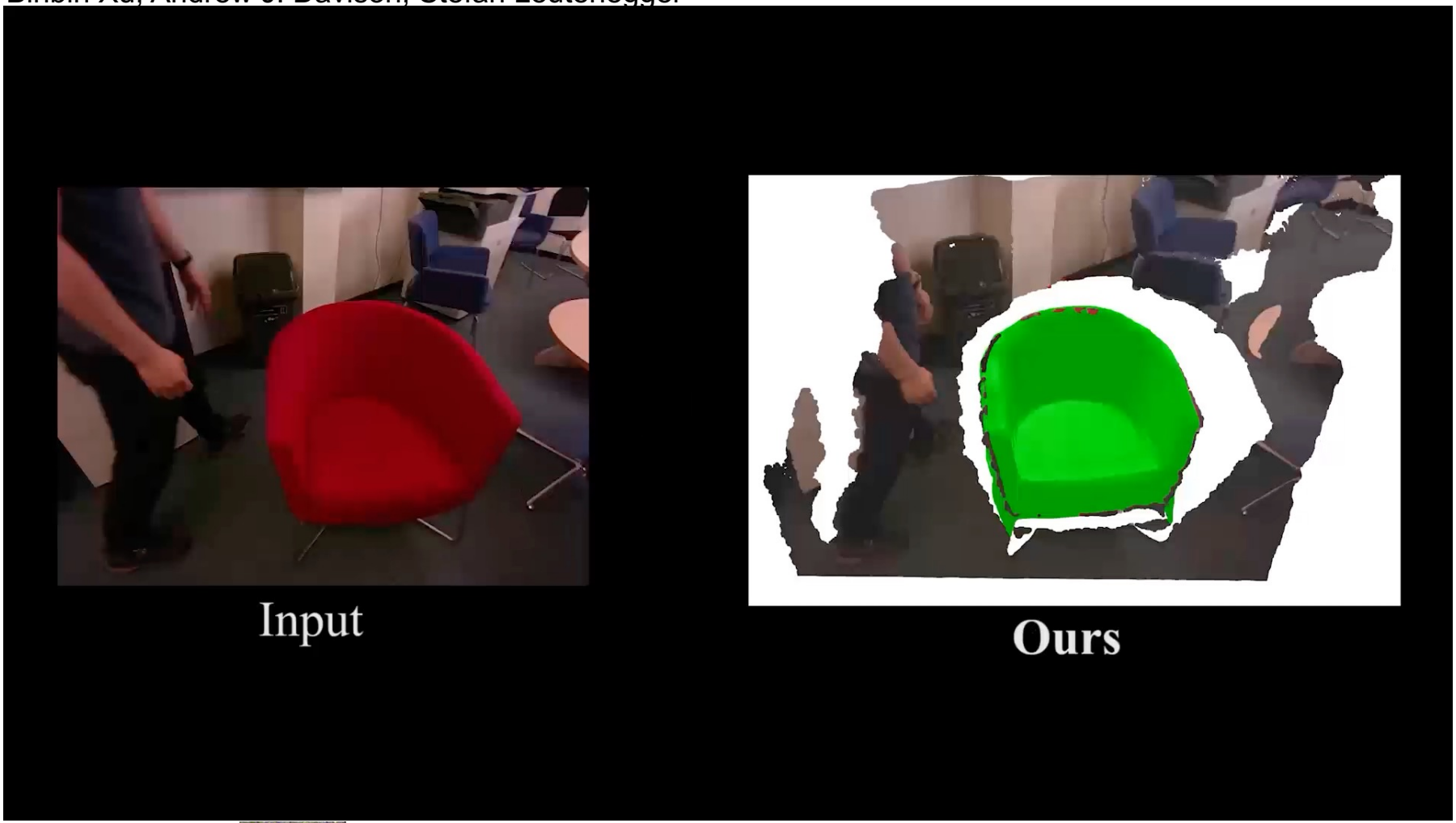




Binbin

Completion of Dynamic Objects [IROS'22]

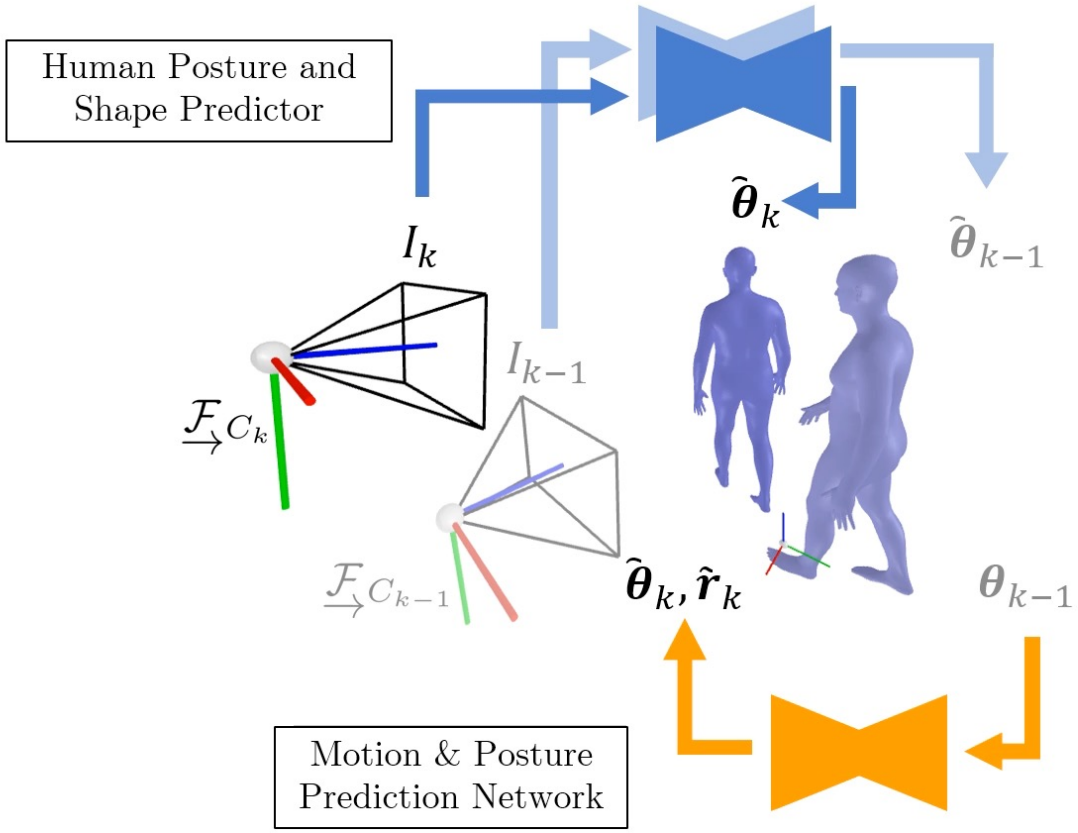
Binbin Xu, Andrew J. Davison, Stefan Leutenegger



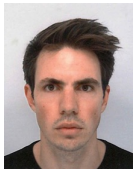
Binbin https://youtu.be/mH22H7jp1D8

BodySLAM [ECCV'22]

Dorian Henning, Tristan Laidlow, Stefan Leutenegger



We introduce a human motion model, that uses past human postures and shapes to predict the next postures and scaled human centre pose changes.



Dorian https://youtu.be/0-SL3VeWEvU

Drone Planning and Control

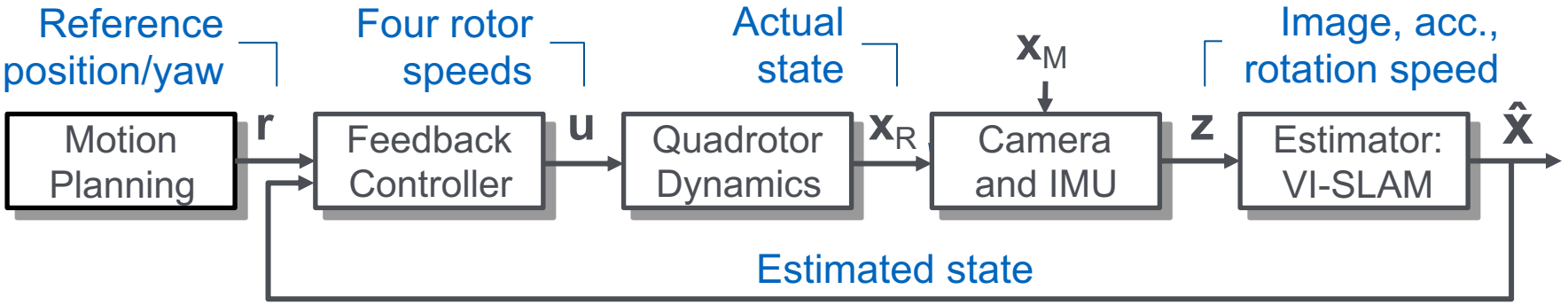
SMART ROBOTICS LAB

Imperial College London

Technische Universität München

PID Drone Control

Can we still use PID, even if this is a complicated MIMO system…?



Interestingly, this is done a lot in practice. But we need to apply some tricks.

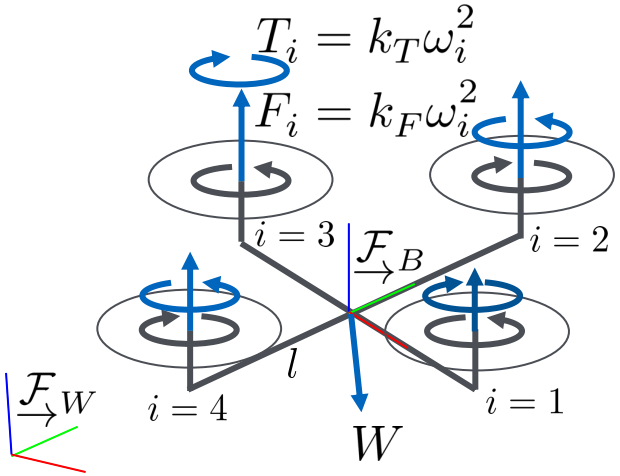
Re-map the input.

From the dynamic model we know:

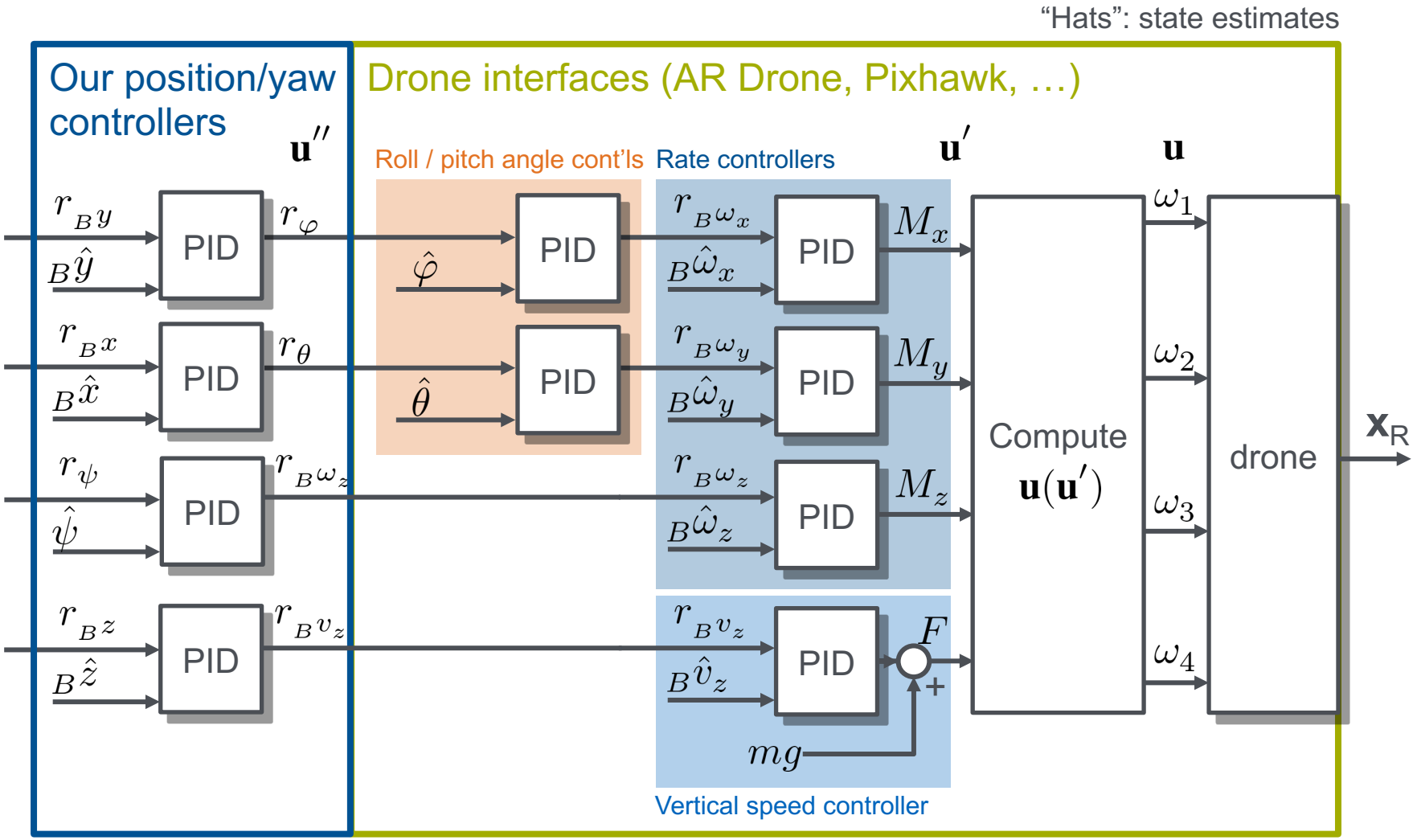
$$ {}_B\mathbf{F} = [0, 0, \underbrace{F_1 + F_2 + F_3 + F_4}_{=:F}]^T - \mathbf{R}_{BW}[0, 0, mg]^T, $$

$$ {}_B\mathbf{M} = [\underbrace{F_2 l - F_4 l}_{=:M_x}, \underbrace{-F_1 l + F_3 l}_{=:M_y}, \underbrace{-T_1 + T_2 - T_3 + T_4}_{=:M_z}]^T. $$

So, instead of $\mathbf{u} = [\omega_1, \omega_2, \omega_3, \omega_4]^T$, we use $\mathbf{u}' = [M_x, M_y, M_z, F]^T$.

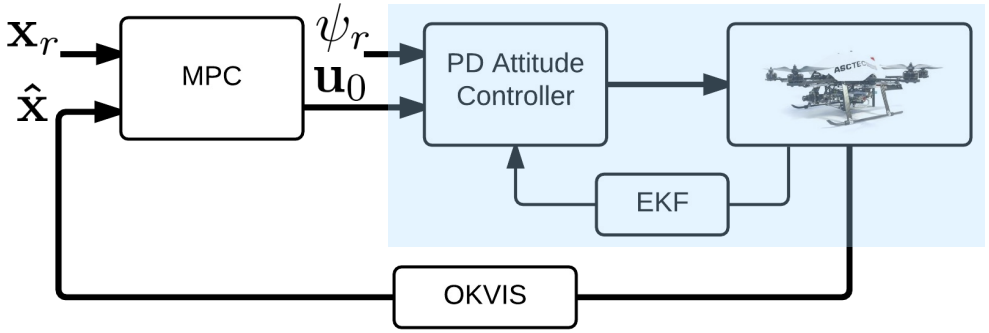


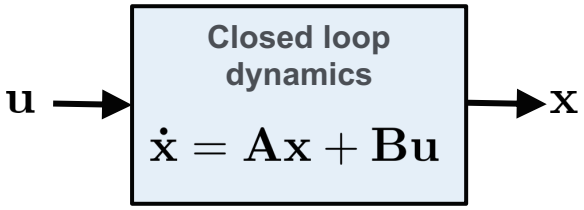
PID Drone Control: Cascaded and Parallel SISO Loops



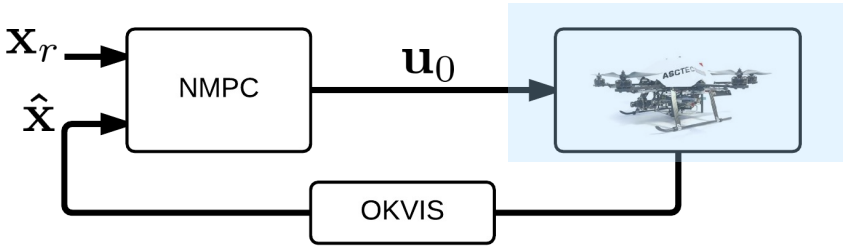
Fully Autonomous MAV Operation

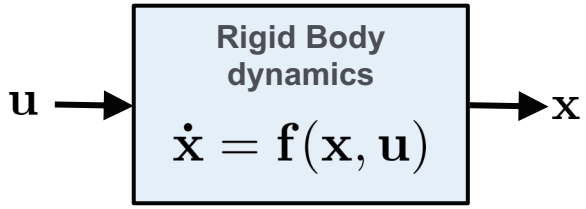
Cascaded **linear MPC** Position-Attitude Control Scheme





Combined Position-Attitude **Nonlinear MPC**

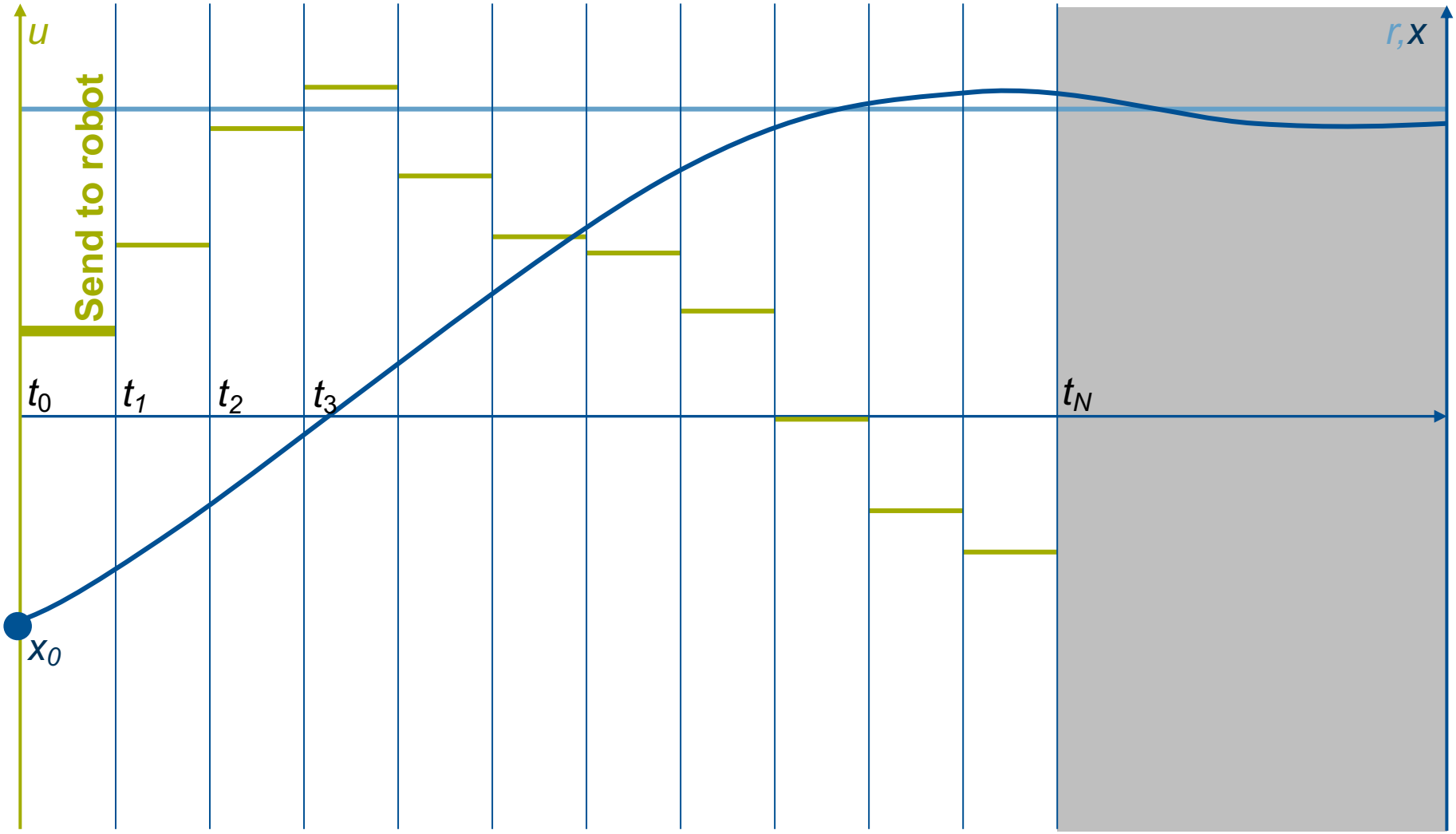




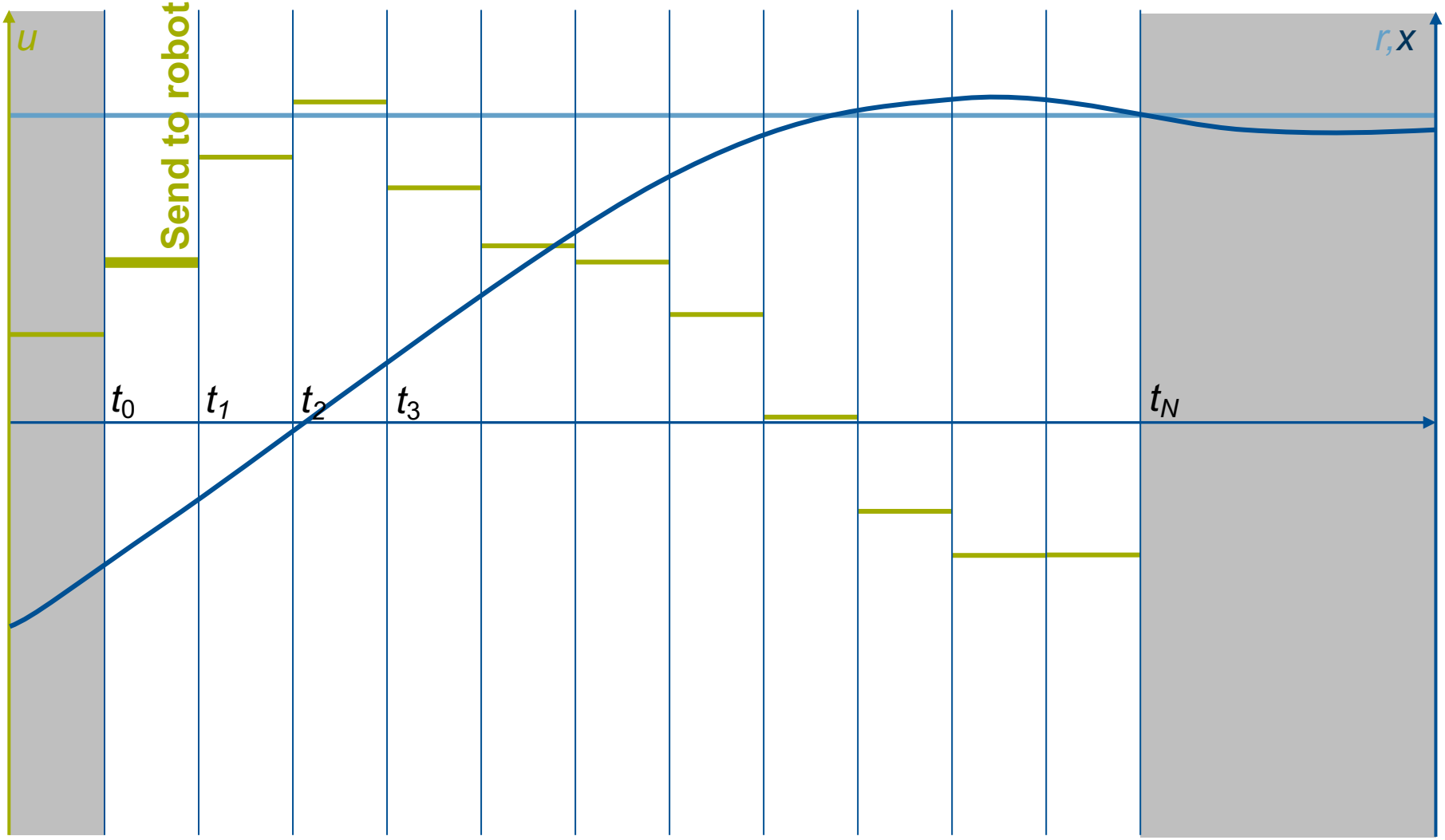
$$\mathbf{x}^T = \left[{}_W\mathbf{r}_B^T, \mathbf{q}_{WB}^T, {}_B\mathbf{v}^T, {}_B\boldsymbol{\omega}^T\right]$$

$$\mathbf{u}^T = \left[\omega_1, \ldots \omega_6\right]$$

MPC: Application as Receding Horizon Controller



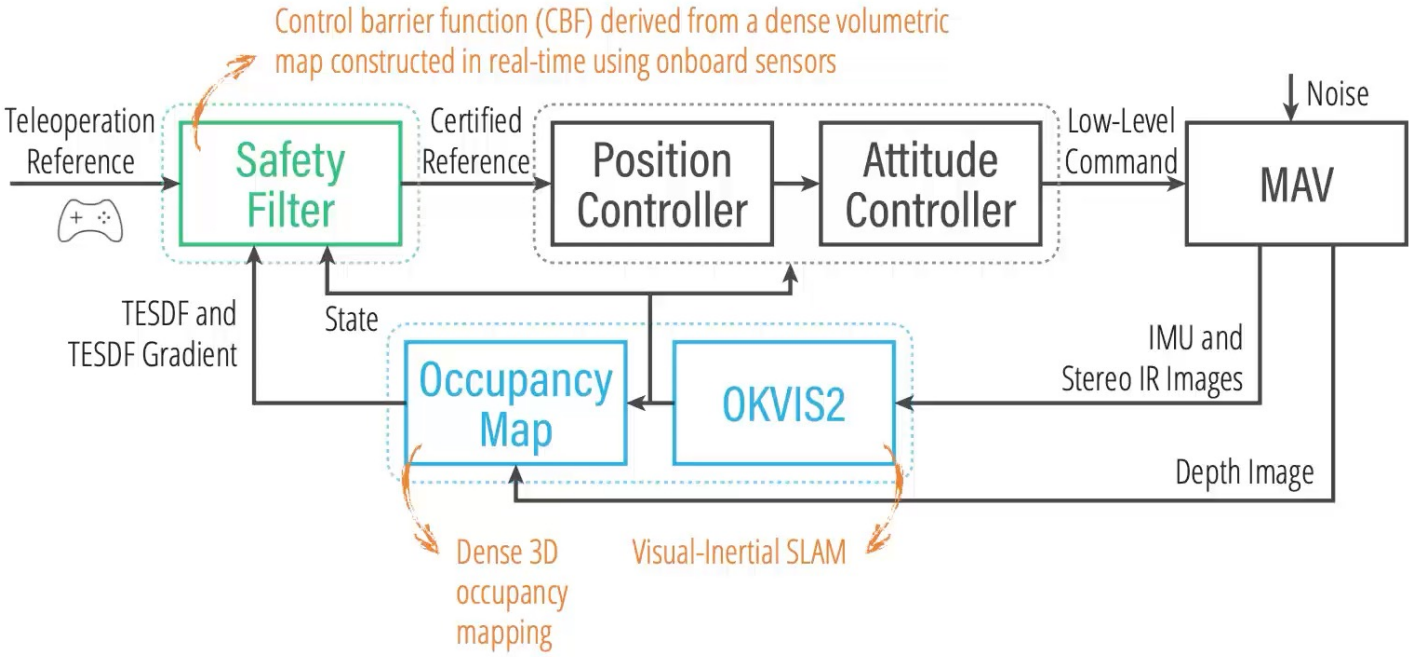
MPC: Application as Receding Horizon Controller



Safe and Easy Teleoperation [ICRA'24]

SiQi Zhou, Sotiris Papatheodorou, Stefan Leutenegger, Angela P. Schoellig

Overview



Sotiris

Nonlinear Model-Predictive Control (NMPC)

System Dynamics

Newton Euler rigid body dynamics

$$ {}_B\mathbf{F} = m\,{}_B\dot{\mathbf{v}} + {}_B\boldsymbol{\omega} \times (m\,{}_B\mathbf{v}) $$

$$ {}_B\mathbf{M} = \mathbf{I}_B\,{}_B\dot{\boldsymbol{\omega}} + {}_B\boldsymbol{\omega} \times (\mathbf{I}_B\,{}_B\boldsymbol{\omega}) $$

Non-linear least squares Identification

$$ \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left(\sum_{i=1}^{N} \|\boldsymbol{\psi}_i - \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \boldsymbol{\theta})\|_2^2 \right) $$

Identified Parameters	
k_T	$6.8110^{-6} N(\frac{rad}{s})^{-2}$
k_M	$0.0395\ m$
I_{xx}	$0.0328\ kg\ m^2$
I_{yy}	$0.0492\ kg\ m^2$
I_{zz}	$0.0972\ kg\ m^2$
C_x	$0.15\ N\ (\frac{m}{s})^{-2}$
C_y	$0.12\ N\ (\frac{m}{s})^{-2}$
C_z	$0.17\ N\ (\frac{m}{s})^{-2}$

NMPC Optimisation problem

Position and yaw penalisation

$$ \mathbf{u}^* = \arg\min_{\mathbf{u}} \int_t^{t+T_h} \left(h_r + h_\psi + h_v + h_u \right) dt $$

Velocity and Input penalization

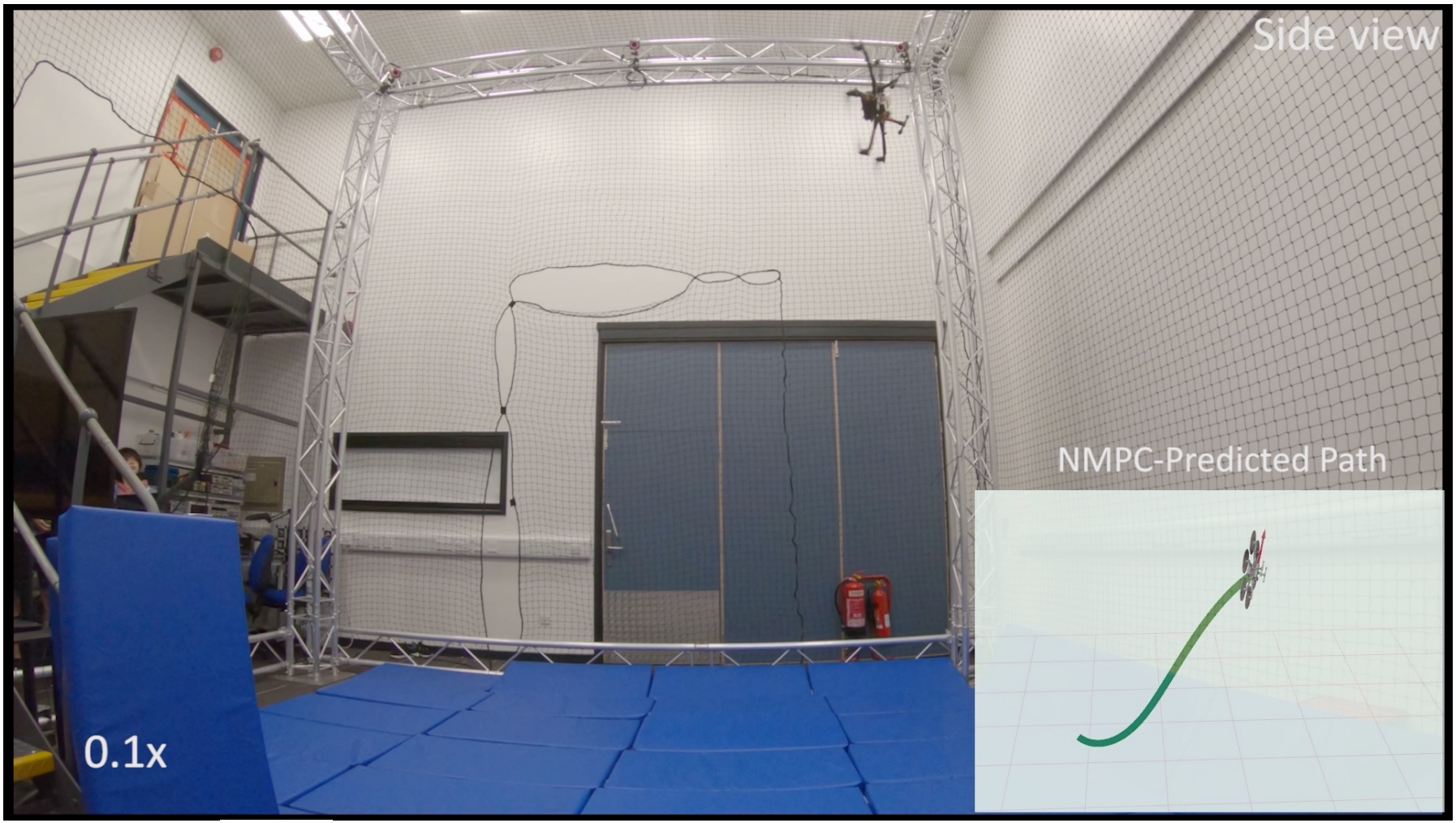
$$ \text{subject to} : \mathbf{x}_0 = \hat{\mathbf{x}}_0 $$

$$ \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) $$

$$ \mathbf{0} \geq \mathbf{g}(\mathbf{x}, \mathbf{u}) $$

Aggressive Flight / Failure Recovery [ICRA'20]

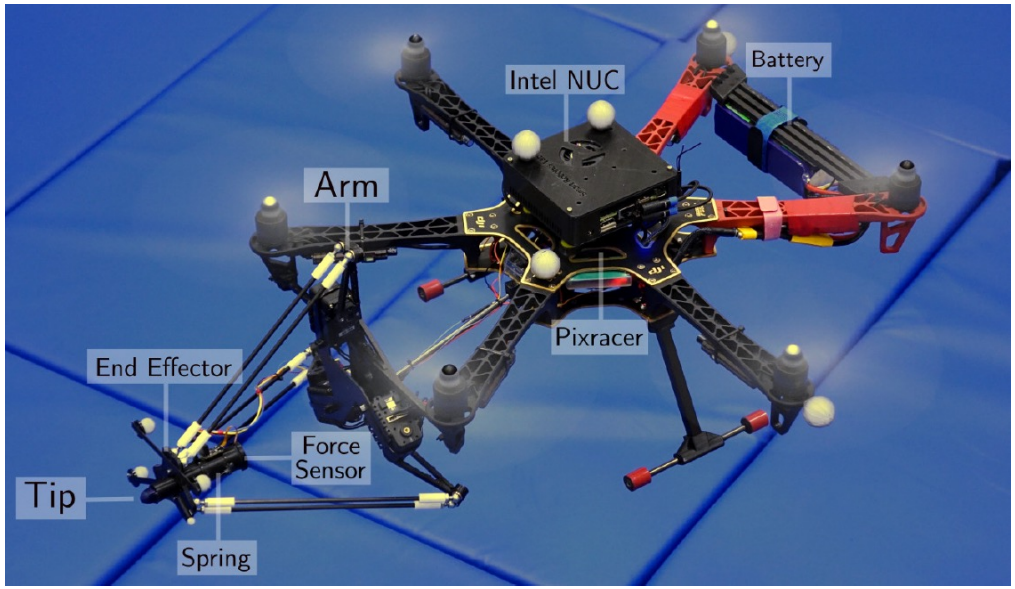
Dimos Tzoumanikas, Qingyue Yan, Stefan Leutenegger



Dimos https://youtu.be/cAQeSZ3tIqY

Integrated Arm and Drone Control

Considering centre of mass shift (quasi-static)
Formulating cost for the end effector (E)



$$\mathbf{u}^\star = \underset{\mathbf{u}}{\operatorname{argmin}} \int_t^{t+T_h} \underbrace{(h_\mathbf{p} + h_\mathbf{v} + h_\mathbf{q} + h_{\boldsymbol{\omega}})}_{\text{Drone state penalisation}} + \underbrace{h_\mathbf{e} + h_c}_{\text{End effector lateral position and contact normal force penalisation}} \, dt$$

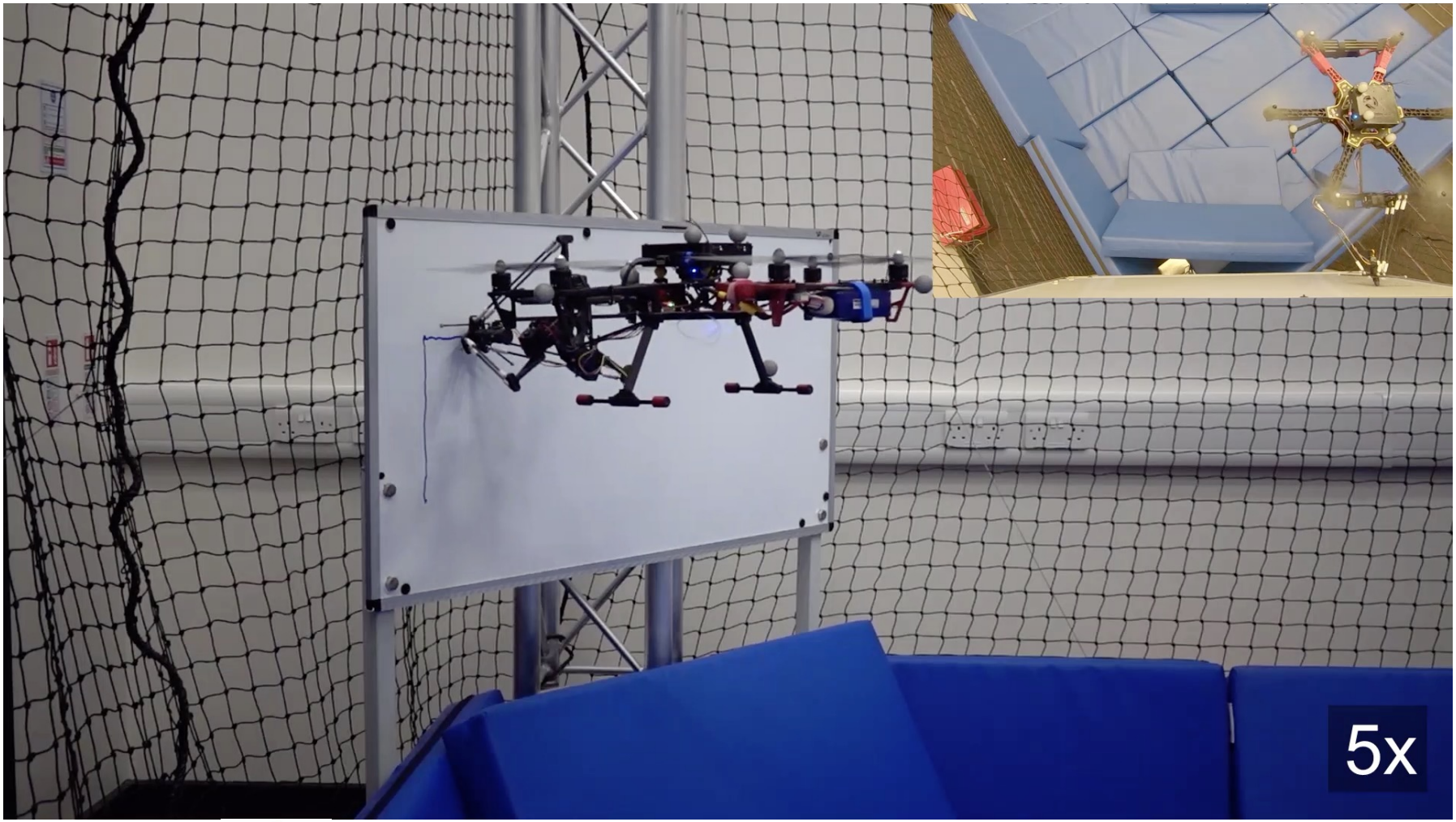
$$\text{s.t.} : \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}),$$

$$\mathbf{u} = [\mathbf{M}, T, {}_B\mathbf{r}_E]^\top,$$

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}.$$

Towards Mobile Manipulation [RSS'20]

Dimos Tzoumanikas, Felix Graule, Qingyue Yan, Dhruv Shah, Masha Popovic, Stefan Leutenegger



Dimos https://youtu.be/iE--MO0YF0o

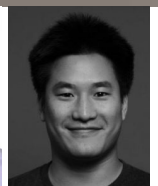
Aerial Additive Manufacturing [Nature'22]

K Zhang, …., D Tzoumanikas, C Choi, …, VM Pawar, RJ Ball, C Williams, P Shepherd, S Leutenegger, R Stuart-Smith & M Kovac

https://youtu.be/pDKNEO0gDuE

Aerial Additive Manufacturing [Nature'22]

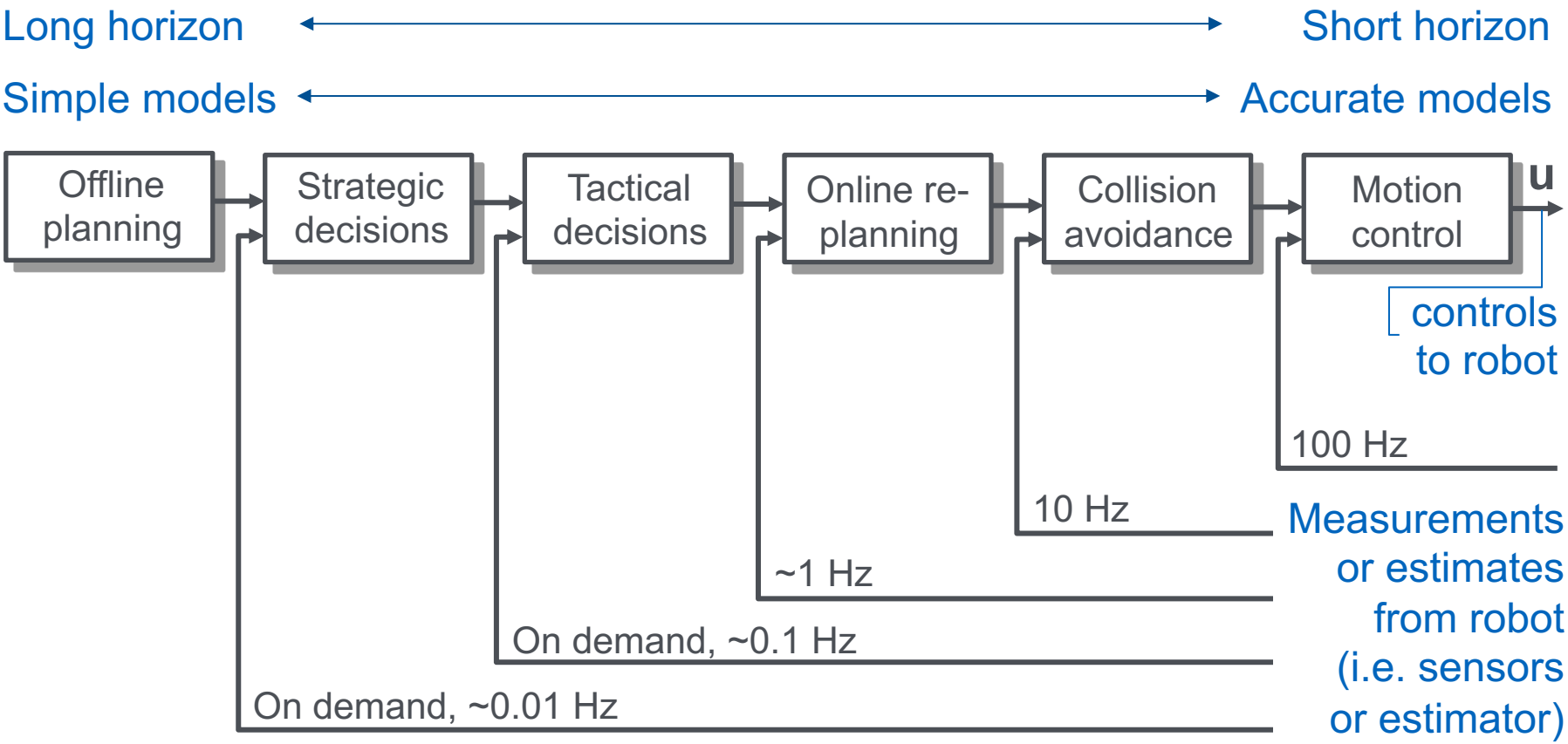
K Zhang, …., D Tzoumanikas, C Choi, …, VM Pawar, RJ Ball, C Williams, P Shepherd, S Leutenegger, R Stuart-Smith & M Kovac



https://youtu.be/pDKNEO0gDuE

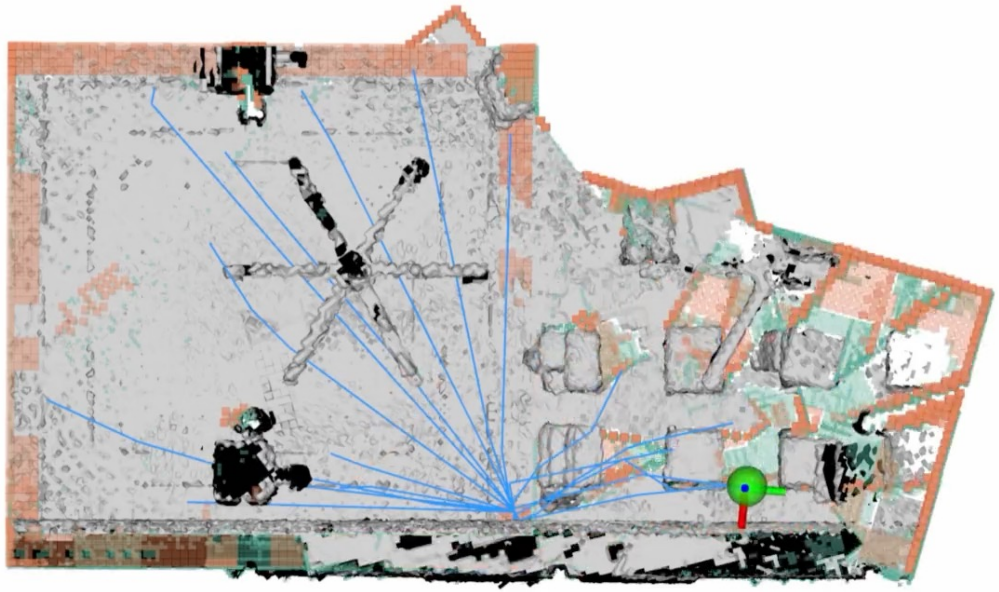
Autonomous Robot Navigation Architectures

Real-world robot navigation might use complex, cascaded feedback loops, e.g.



Exploration with LiDAR/Stereo-Inertial SLAM [Subm.]

Sotiris Papatheodorou*, Simon Boche*, Sebastián Barbas Laina, Stefan Leutenegger



5x

The candidate with the highest utility is selected as the next goal.

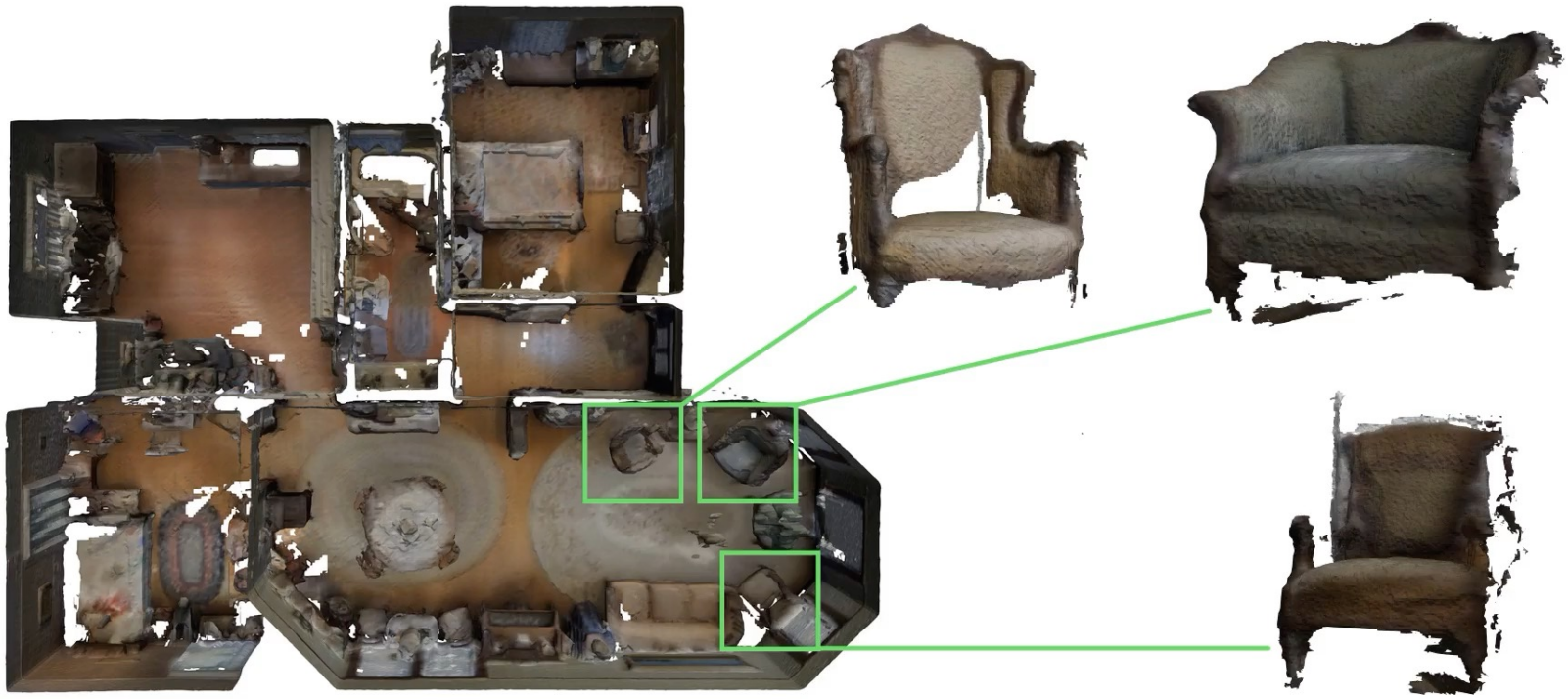
Sotiris

Simon B.

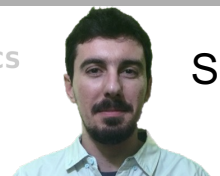
Finding Things in the Unknown [ICRA'23]

Sotiris Papatheodorou, Nils Funk, Dimos Tzoumanikas, Christopher Choi, Binbin Xu, Stefan Leutenegger

Object-centric autonomous exploration



We extend it to detect objects of interest and create detailed reconstructions



Sotiris

https://youtu.be/z0LVe_8SATU

Conclusions

- **Contributions SLAM and Spatial AI**
  - From robot **motion** to **dense** geometry, **semantics**, **objects**, and **human motion**
- **(Deep) Learning proves helpful for**
  - Inference of **semantics** / **objects** / **people** (including 6D pose and posture)
  - **Data association** over time
  - **Completion** of geometry beyond the visible
- **The modular architecture allows for integration with robot planning and control.**

Open Challenges:

- What are the roles of different kinds of learning robot tasks? E.g. imitation vs. reinforcement?
- How do we solve long-horizon tasks?
- How can LLMs be best leveraged and connected to Spatial AI representations to execute robot action?

Funders

Bayerisches Staatsministerium für Wissenschaft und Kunst

Imperial College London

Engineering and Physical Sciences Research Council

Innovate UK

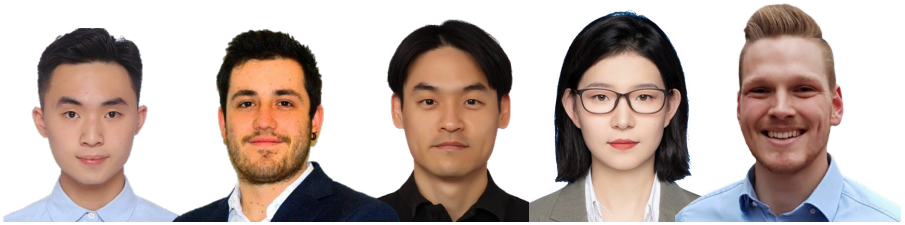
slamcore

intel. Meta Google

HEXAGON

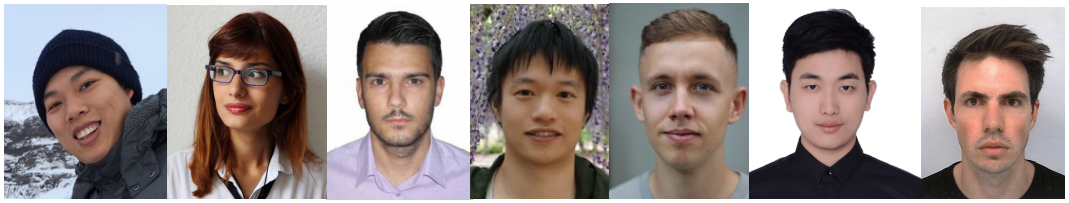
Thanks to the SRL Members!

Chris Sotiris Simon B. Simon S.



Hanzhi Sebastián Jaehyung Jiaxin Yannick

SRL Alumni:



Wenbin Masha Dimos Binbin Nils Xingxing Dorian

… and many other collaborators

SMART ROBOTICS LAB www.srl.cit.tum.de

Funders

Bayerisches Staatsministerium für Wissenschaft und Kunst

Imperial College London

TUM

UKRI Engineering and Physical Sciences Research Council

UKRI Innovate UK

slamcore

intel. Meta Google

HEXAGON

Imperial College London

Technische Universität München