University of Zurich[UZH]
ETH zürich
Institute of Informatics – Institute of Neuroinformatics
ROBOTICS & PERCEPTION GROUP

Learning Autonomous, Vision-based, Agile Flight

Davide Scaramuzza & Elia Kaufmann

http://rpg.ifi.uzh.ch

Related Work on Learning Vision-based Flight

Learning monocular reactive UAV control in cluttered natural environments

Task: Collision free flight in a forest

Network input: Images from forward-facing camera

Network output: desired lateral speed

Training methodology: Supervised Learning with recorded data from human pilot. After an initial training on the expert data, the policy is refined using Dataset Aggregation.

Limitations:

- 2D *sideway motion only* (roll left or right),
- vehicle dynamic not taken into account
- constant linear speed (<1.5m/s)

Ross, Melik-Barkhudarov, Shankar, Wendel, Dey, Bagnell, Hebert,
Learning monocular reactive UAV control in cluttered natural environments, ICRA'13

CAD2RL - Real single-image flight without a single real image

Task: Follow the hallway, do obstacle avoidance

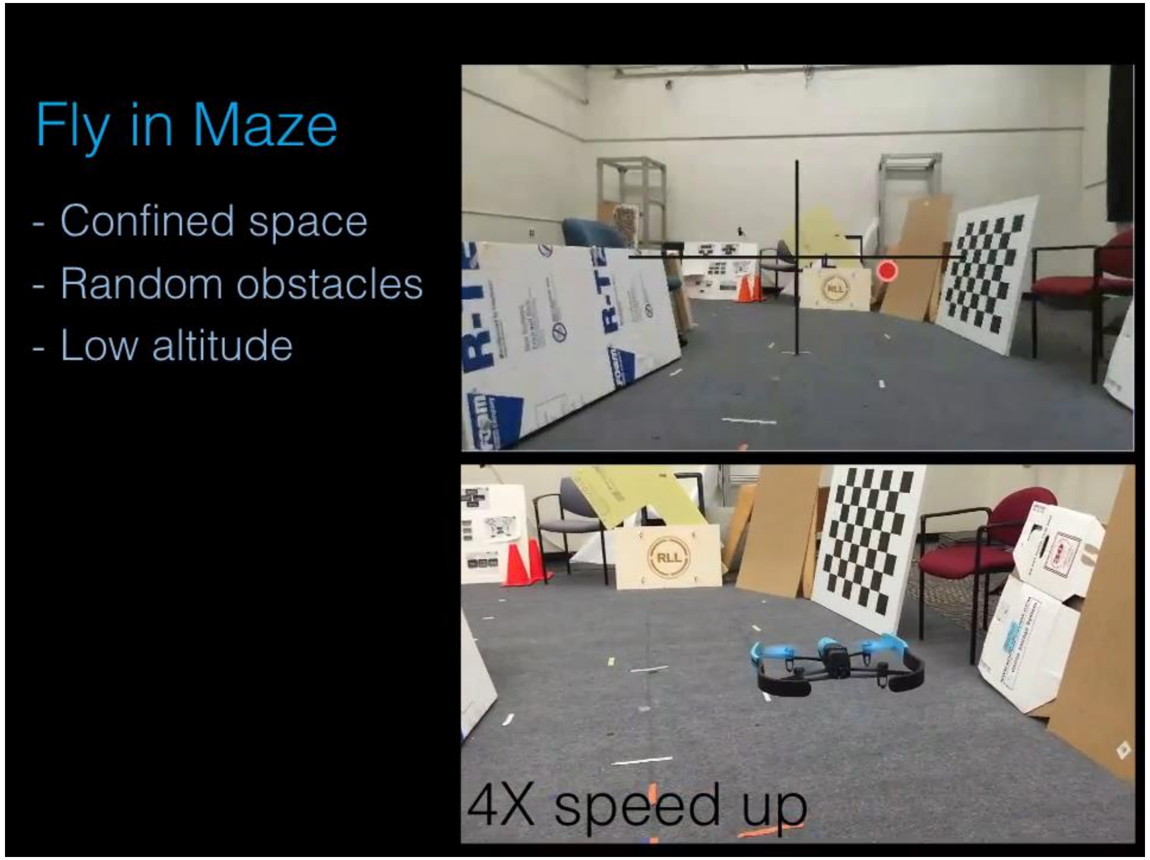
Network input: Images from forward-facing camera

Network output: desired heading angle

Training methodology: Reinforcement Learning in simulation, the network is ported to the real platform without any fine-tuning

Limitations:

- 2D *stop-rotate-go* motion,
- vehicle dynamic not taken into account
- constant linear speed (<1m/s)



Sadeghi, Levine,
CAD2RL - Real single-image flight without a single real image, RSS'17

Learning to fly by crashing

Task: Collision-free navigation in indoor scenes

Network input: 3 image from forward/left/right-facing camera

Network output: turn left, go straight, turn right

Training methodology: Supervised learning on hand-recorded data

Limitations:

- 2D *stop-rotate-go* motion,
- vehicle dynamic not taken into account
- constant linear speed (<0.5m/s)

Gandhi, Pinto, Gupta,
Learning to fly by crashing, IROS'17

A Machine Learning Approach to the Visual Perception of Forest Trails for Mobile Robots

Task: Follow the forest trail

Network input: Image from forward-facing camera

Network output: turn left, go straight, turn right

Training methodology: Supervised learning on hand-recorded data

Limitations:

- 2D motion,
- vehicle dynamic not taken into account
- constant linear speed (<2m/s)

Giusti, Guzzi, Ciresan, Lin He, Rodríguez, Fontana, Faessler, Forster, Schmidhuber, Di Caro, Scaramuzza, Gambardella,
A Machine Learning Approach to the Visual Perception of Forest Trails for Mobile Robots, RAL'16
PDF PPT Datasets YouTube

Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness

Task: Follow the forest trail

Network input: Image from forward-facing camera

Network output: heading angle & lateral offset to forest trail

Training methodology: Supervised learning on hand-recorded data

Limitations:

- 2D *unicycle motion*,
- vehicle dynamic not taken into account
- constant linear speed (<2m/s)

Smolyanskiy, Kamenev, Smith, Birchfield,
Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness, IROS'17

DroNet: Learning to Fly by Driving

Task: Follow an urban road and stop with obstacles

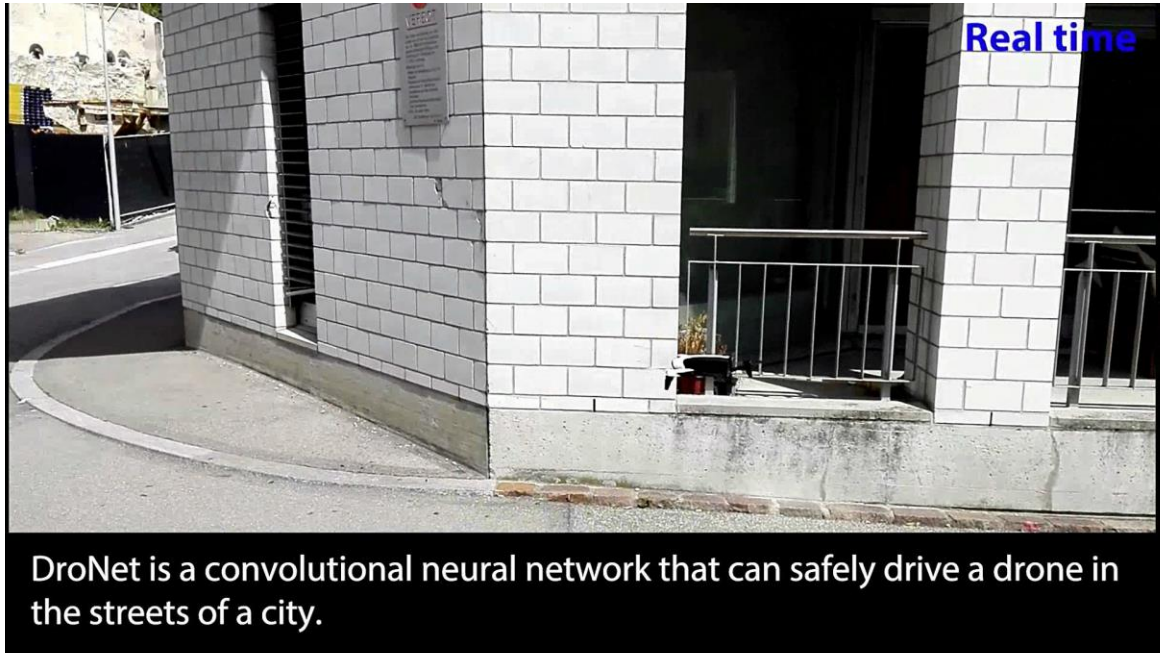
Network input: Image from forward-facing camera

Network output: steering angle & probability of collision

Training methodology: Supervised learning from car and bicycle data

Limitations:

- 2D *unicycle* motion,
- vehicle dynamic not taken into account
- speed <2m/s



Loquercio, Maqueda, Del Blanco, Scaramuzza, DroNet: Learning to Fly by Driving, RAL'18. PDF. Video. IEEE Spectrum
Code, datasets, and training models: http://rpg.ifi.uzh.ch/dronet.html

Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning

Task: Navigate through a set of gates

Network input: Image from forward-facing camera

Network output: segmentation of gate, velocity commands are computed to align optical center with gate center. Network inference is performed onboard.

Training methodology: supervised training

Limitations:

- 2D *unicycle* motion,
- vehicle dynamic not taken into account

Jung, Hwang, Shin, Shim,
Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning, RAL’18

Learning a Controller Fusion Network by Online Trajectory Filtering for Vision-based UAV Racing

Task: Navigate through a set of gates as fast as possible

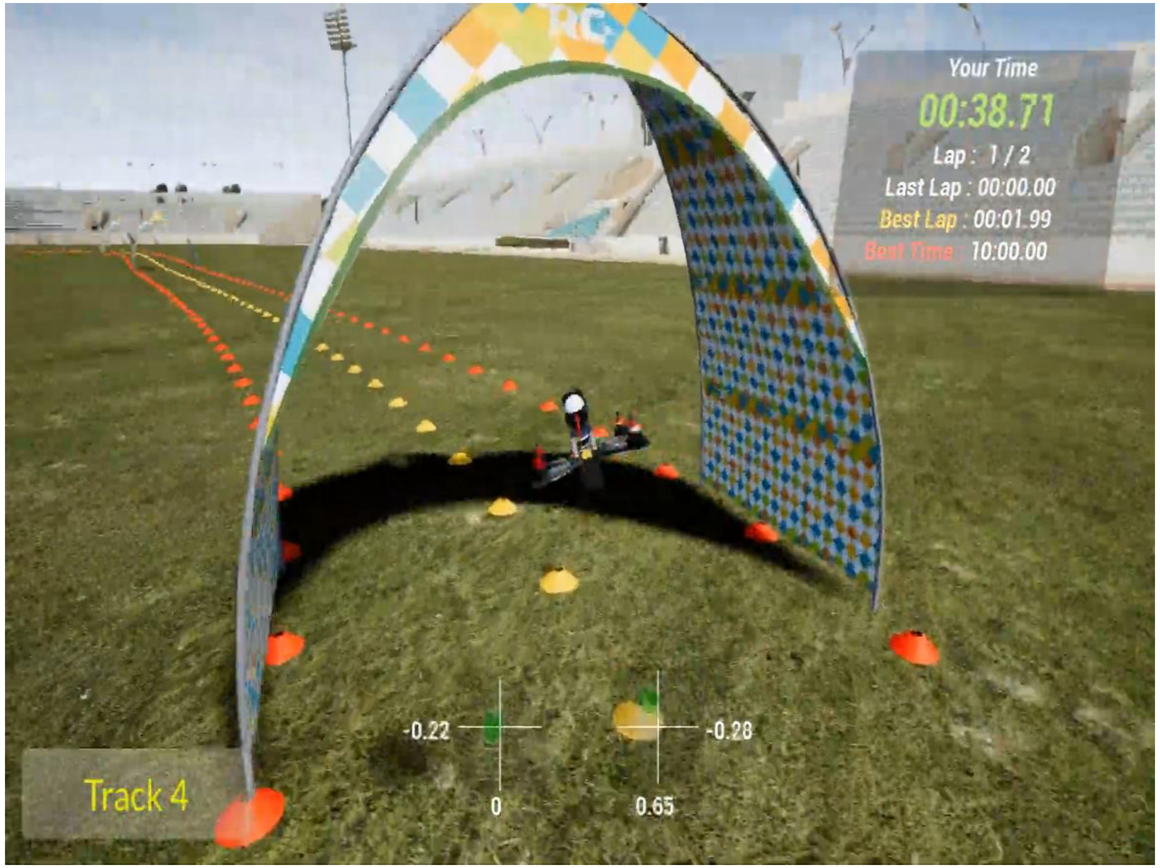
Network input: Image from forward-facing camera & platform state

Network output: low-level control commands (body rates & thrust)

Training methodology: The network is trained using an ensemble of classical controllers. Each classical controller is evaluated, the best one is chosen to imitate.

Limitations:

- Assumes **perfect knowledge** of system **state**,
- **Only works in simulation**



Müller, Li, Casser, Smith, Michels,
Learning a Controller Fusion Network by Online Trajectory Filtering for Vision-based UAV Racing, CVPRW’19

Taxonomy of Related Work on Learning Vision-based Flight

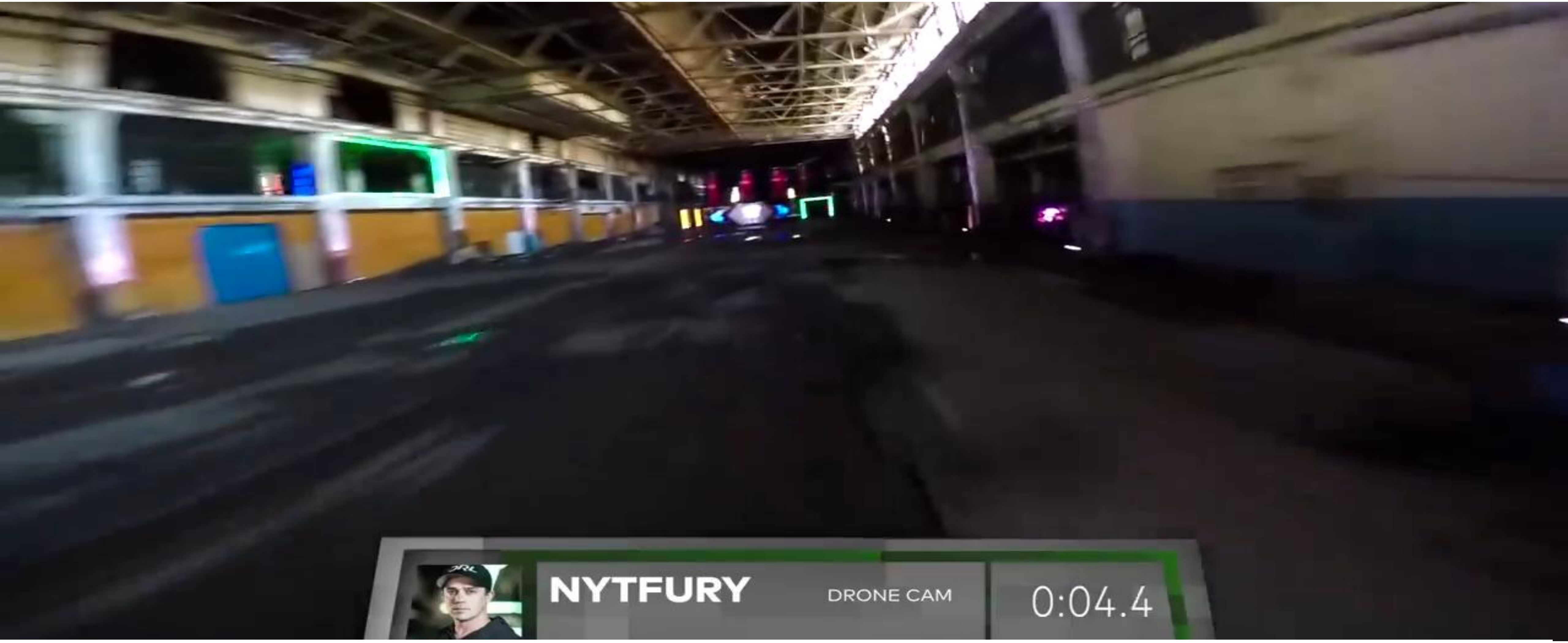
- **Images to intermediate-level commands** [Ross, ICRA'13], [Giusti, RAL'16], [Smolyanskiy, IROS'17], [Sadeghi, RSS'17], [Gandhi, IROS'17], [Loquercio, RAL'18], [Jung, RAL'18]
  - **Output of the network:** linear speed commands / desired steering angle)
  - **Pros**: stable
  - **Cons**:
    - **2D motion only**
    - vehicle **dynamic not taken into account**
    - **known state estimate**

- **Images to low-level control commands** [Müller, CVPRW'19]
  - **Output of the network:** body rates & thrust
  - **Pros**: vehicle dynamic taken into account
  - **Cons**:
    - **unstable**, may crash at any time
    - works **only in simulation**
    - **known state estimate**

Can we learn **high-level commands** (e.g., waypoints) in order to take advantage of plethora of existing optimal-control algorithms for UAVs?

Loquercio, Scaramuzza, Learning to Control Drones in Natural Environments: A Survey, ICRA18 Workshop on Perception, Inference, and Learning

What does it take to fly as **good as or better** than a human pilot?



WARNING! This drone flown is NOT autonomous; it is operated by a human pilot.
Human pilots take years to acquire the skills shown in this video.
Can we use drone racing as a proxy to learn agile flight?

Why is Drone Racing important?

It raises **fundamental challenges** for robotics research:

- **Real-time coupling of perception and action**
- Coping with **inaccurate models** of **sensors, actuators**, **environment**
- Coping with **dynamically changing** environments
- Coping with **unreliable perception and state estimation**:
  - **low texture**
  - **HDR scenes**
  - **motion blur**

Why is Drone Racing important?

Knowledge transfer to other domains!

Delivery

Search and Rescue

Surveillance

Inspection

Outline

- Imitating expert trajectories
- Generalization to unseen environments
- Simulation to real-world transfer

Deep Drone Racing:
Learning Agile Flight in Dynamic Environments

CORL 2018

Best System Paper Award

PDF YouTube


Elia Kaufmann


Antonio Loquercio


René Ranftl


Alexey Dosovitskiy


Vladlen Koltun


Davide Scaramuzza

Related Work

Drews CoRL'17 (E. Theodorou's lab)	Aggressive Deep Driving: Model Predictive Control with a CNN Cost Model	
Pan RSS'18 (E. Theodorou's lab)	Agile Autonomous Driving using End-to-End Deep Imitation Learning	

Approach

Approach

Camera

30Hz

Images

Neural Network

10Hz

Waypoint & Speed

Trajectory Generation

?

Position / Velocity

Motors

1kHz

Motor Commands

Low Level Control

100Hz

Thrust + Bodyrates

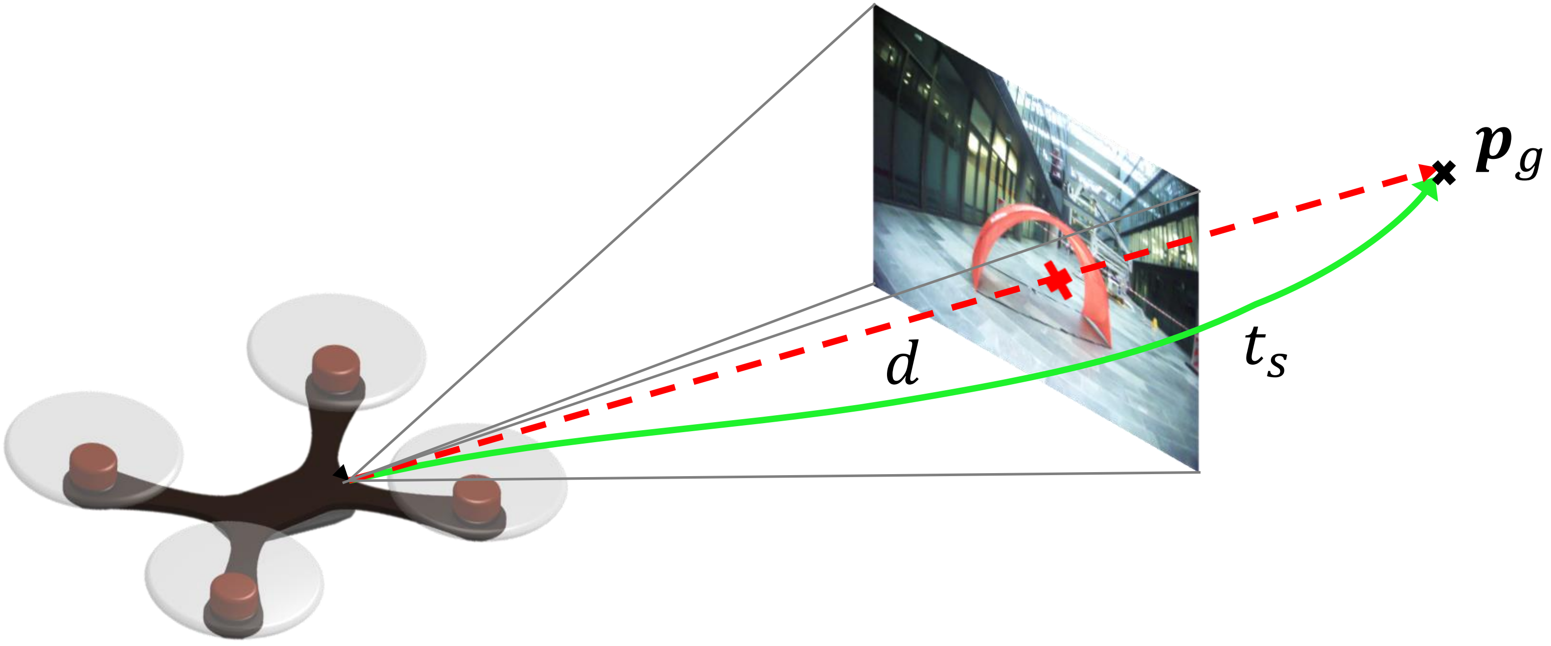
High Level Control

Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Approach

Image coordinates Speed

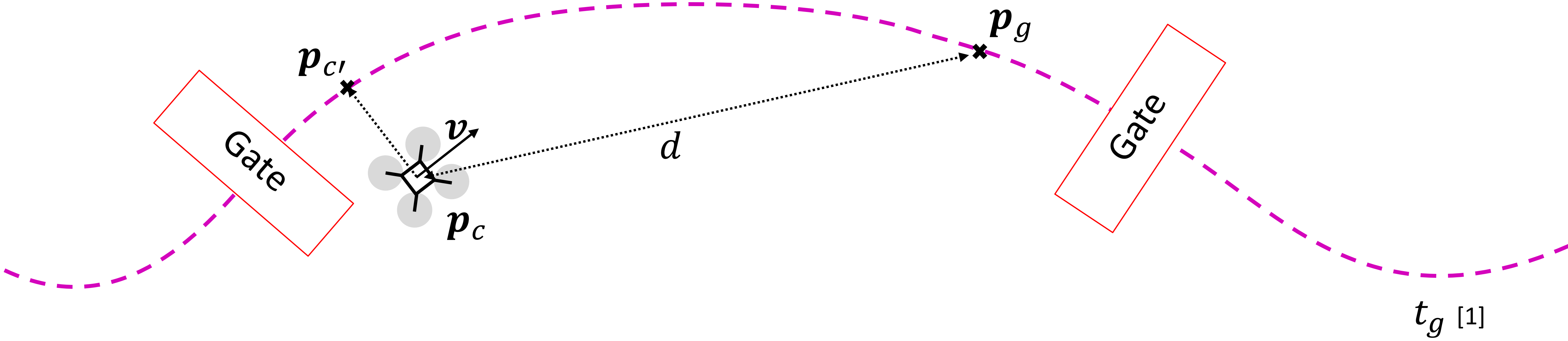
Image → Neural Network → $\{x, y, v\} \in \mathbb{R}^3$ → $\boldsymbol{p}_g \in \mathbb{R}^3$ → Trajectory t_s [1]



[1] Mueller, Hehn, D'Andrea: A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification

Approach: Data Collection

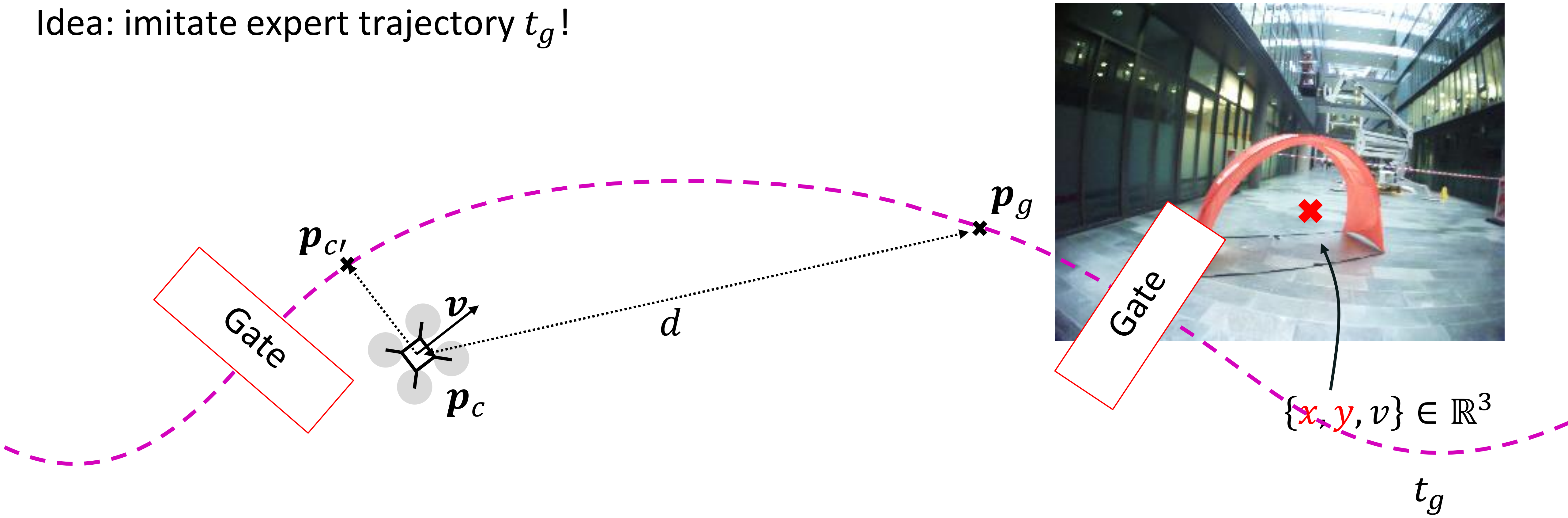
Idea: imitate expert trajectory t_g



Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Approach: Data Collection

Idea: imitate expert trajectory t_g!

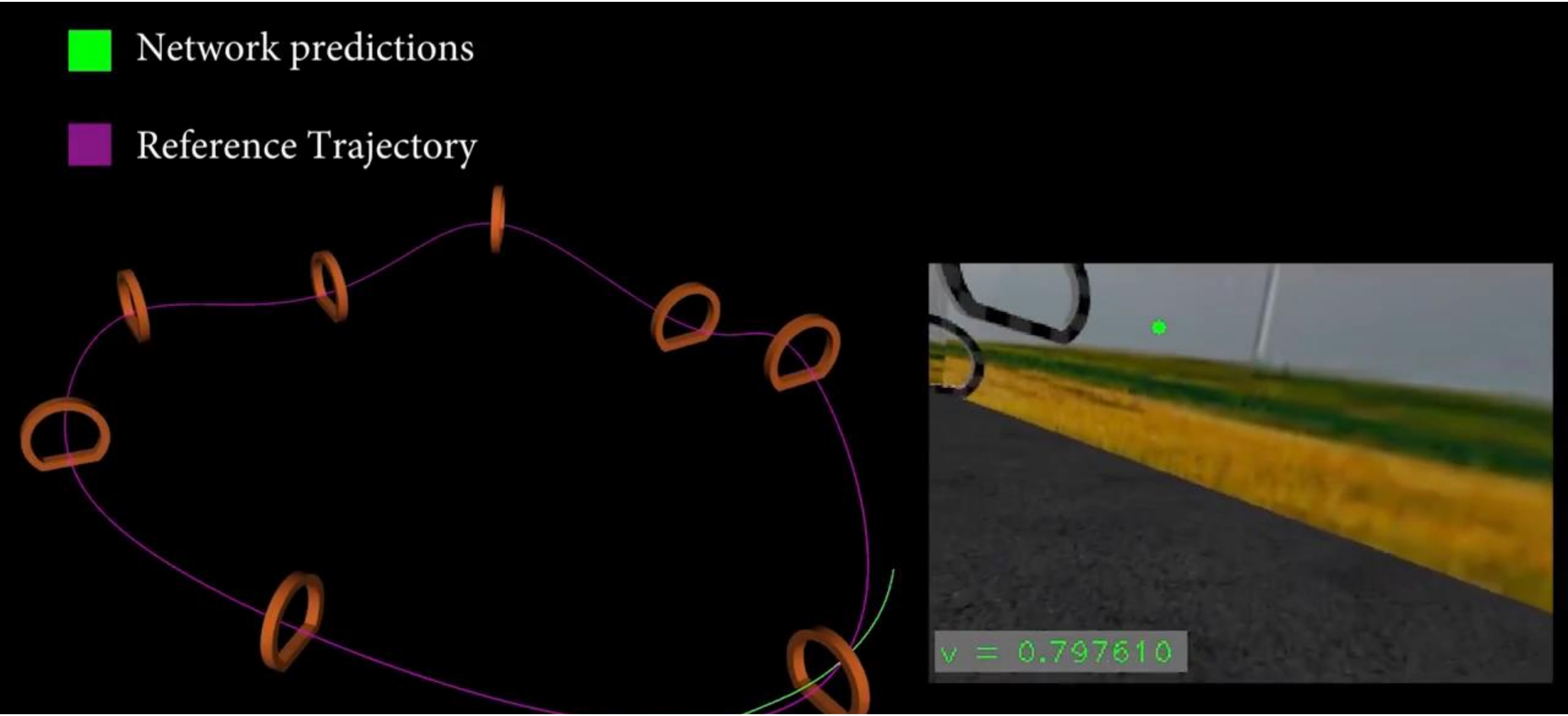


Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Experiments

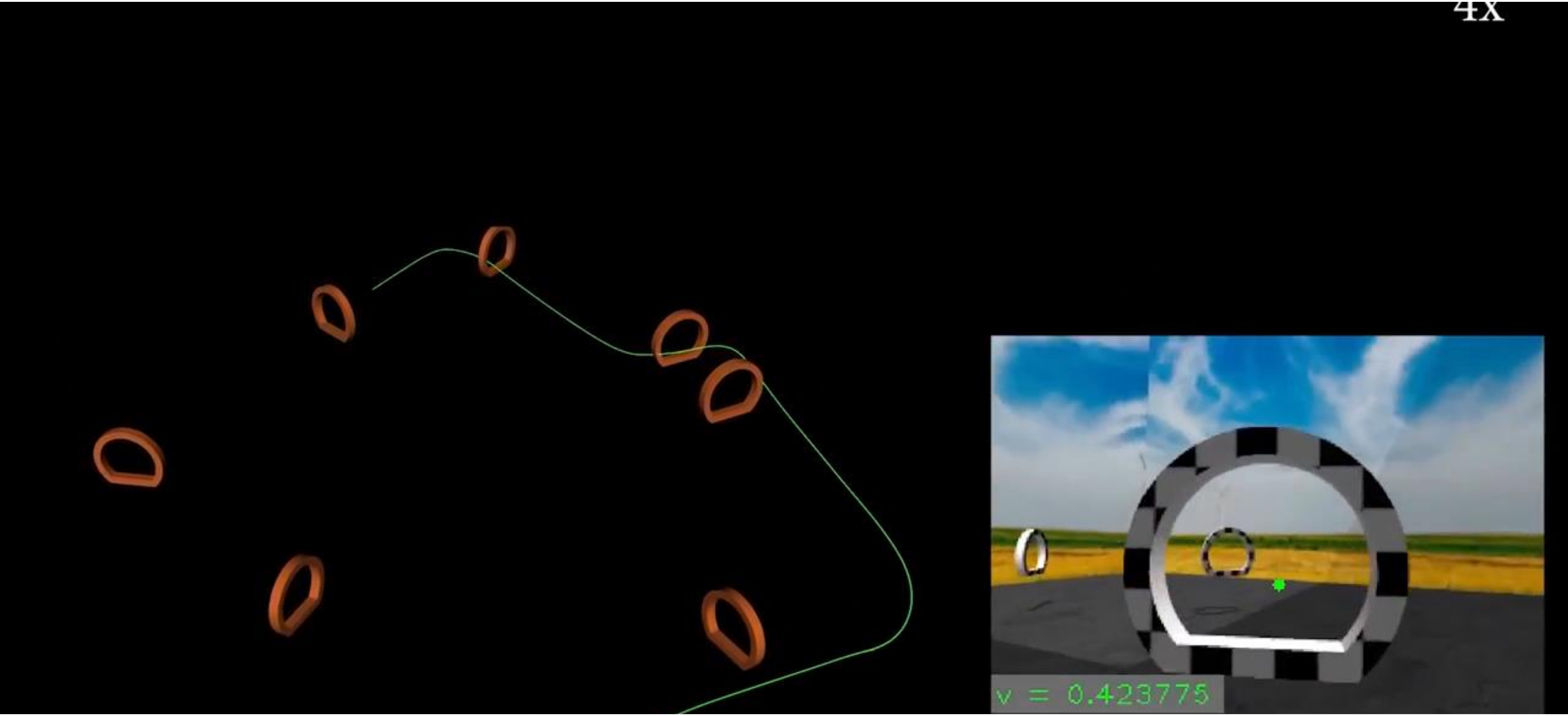
Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Simulation Experiments with Static Gates



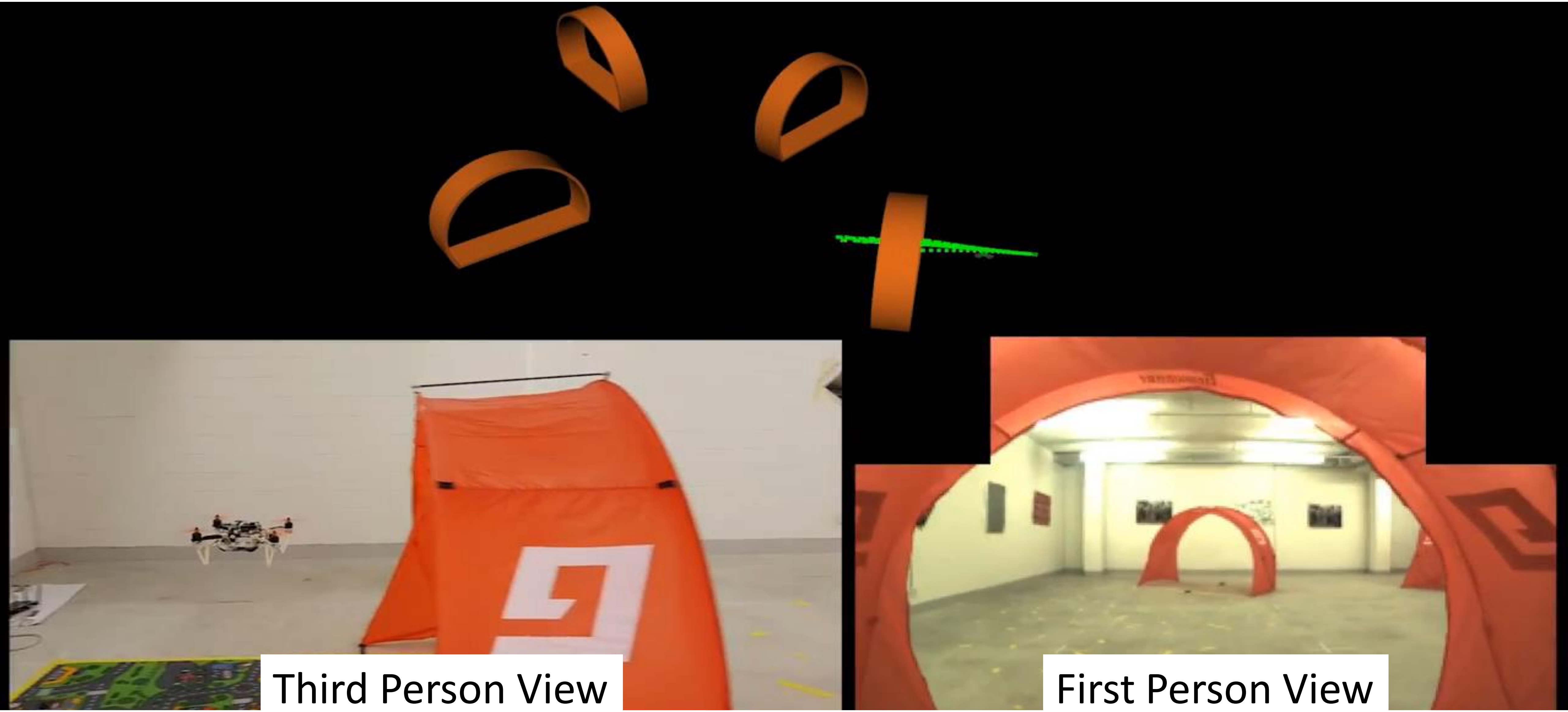
Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Simulation Experiments with Moving Gates



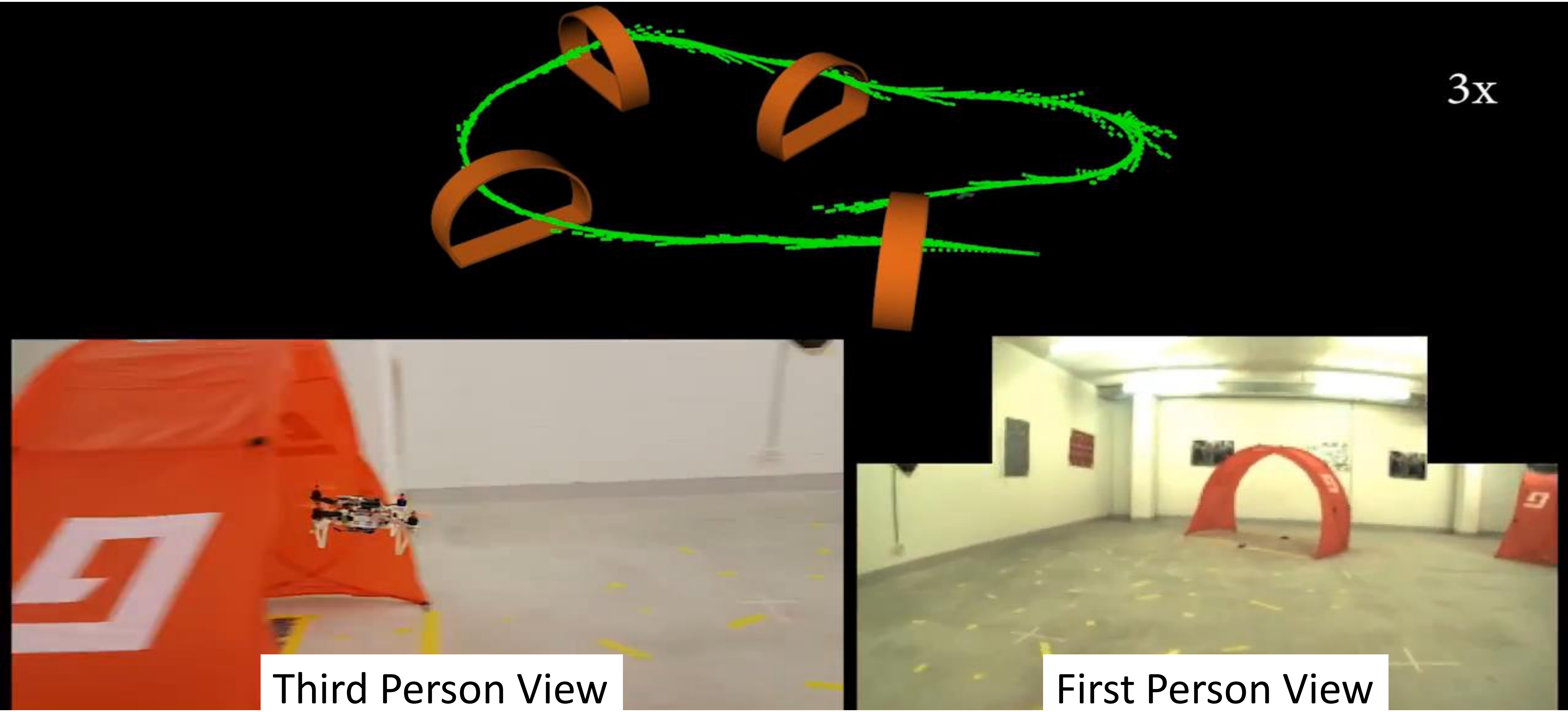
Kaufmann et al., Deep Drone Racing, CORL’18, **Best System Paper Award**. PDF. Video.

Robustness against drift in state estimation



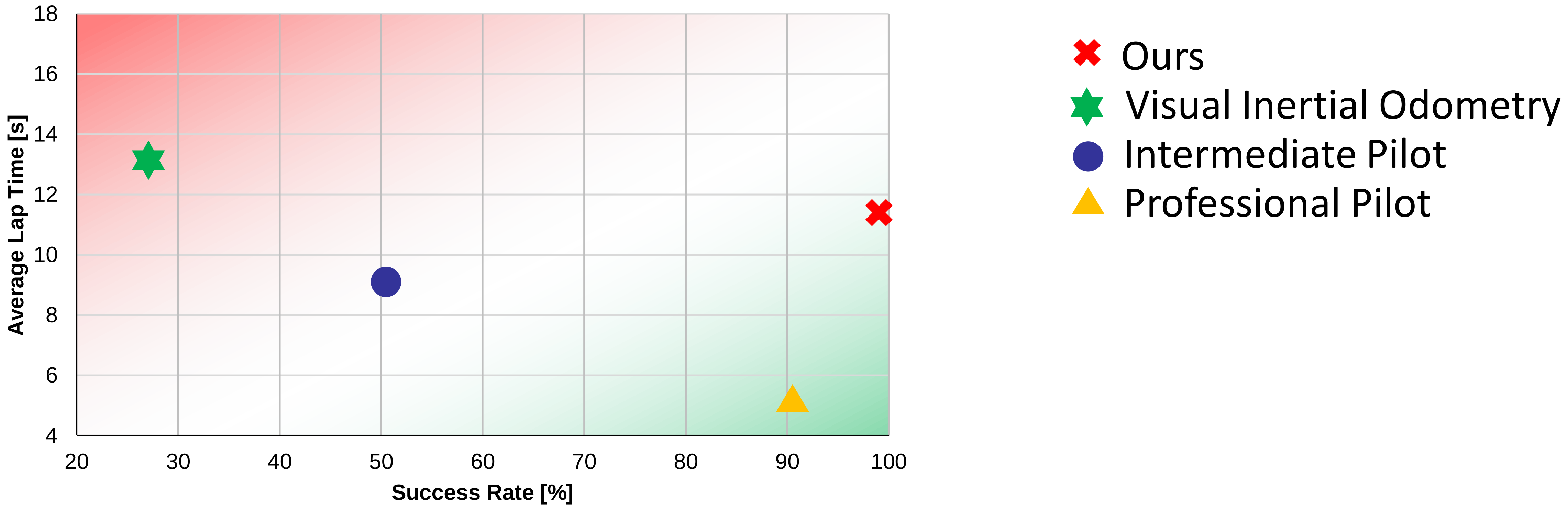
Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Robustness against drift in state estimation



Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Real World Experiments



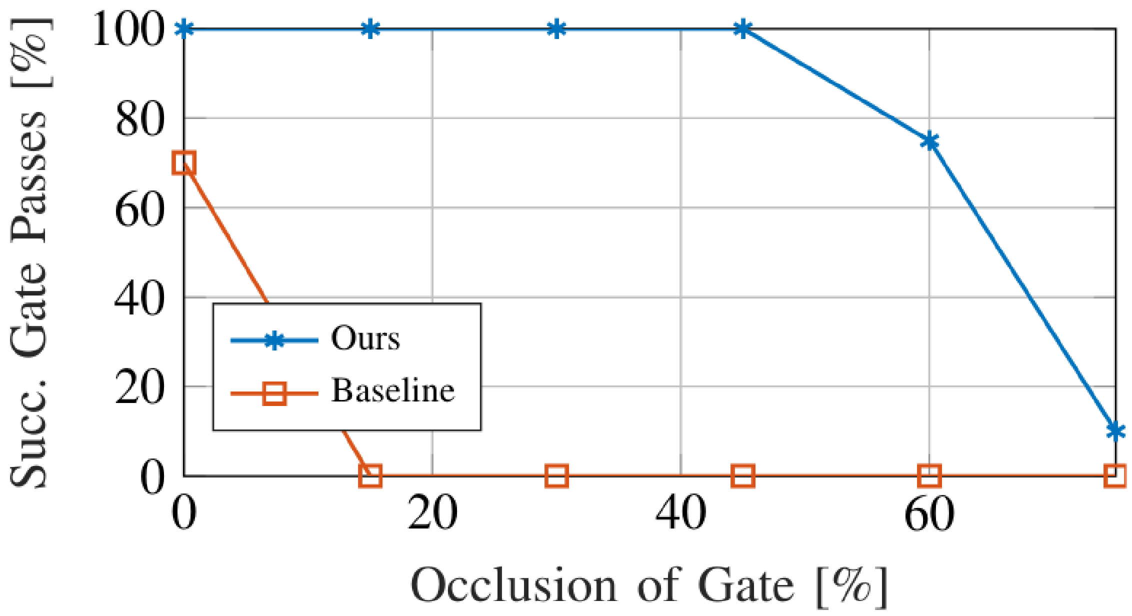
- Outperform Visual Inertial Odometry baseline
- Experienced pilots still better

Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. [PDF](). [Video]().

Quantitative Results: Robustness to Occlusions

We compared the performance of our method against an handcrafted gate detector.

Our approach is significantly more robust to occlusions of the gate!



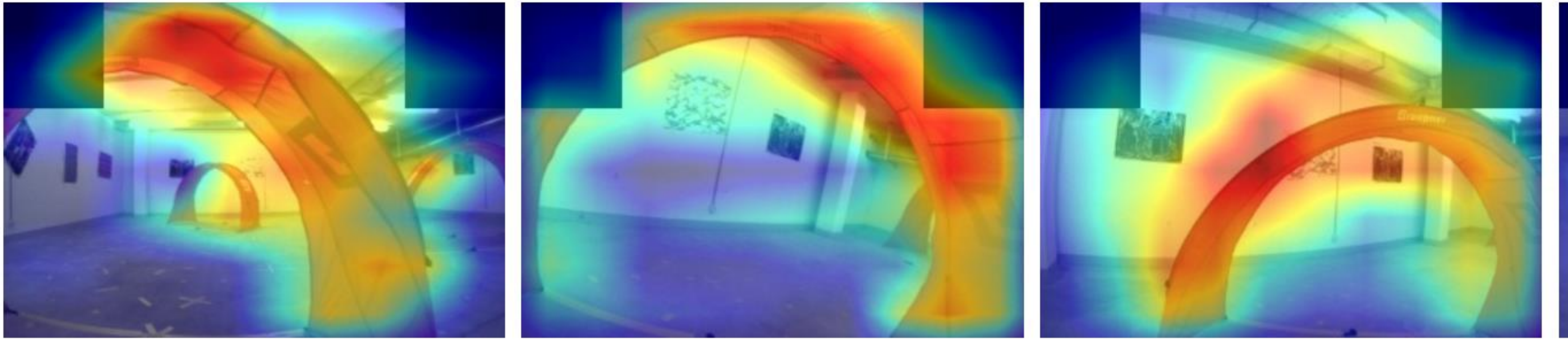
Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Moving gates

Kaufmann et al., Deep Drone Racing, CORL'18, **Best System Paper Award**. PDF. Video.

Qualitative Results: What is the network looking at?

Using GradCam [Selvaraju et al. 2017], we investigated which features a network uses to make control decisions.



[Selvaraju et al. 2017] Grad-CAM: Visual explanations from deep network via gradient-based localization, ICCV, 2017

What about New Environments?

- Previous system was trained for a **specific environment**

What about training a policy that is only gate-specific?

Beauty and the Beast:
Optimal Methods Meet Learning for Drone Racing

ICRA 2019

PDF Video


Elia Kaufmann


Mathias Gehrig


Philipp Foehn

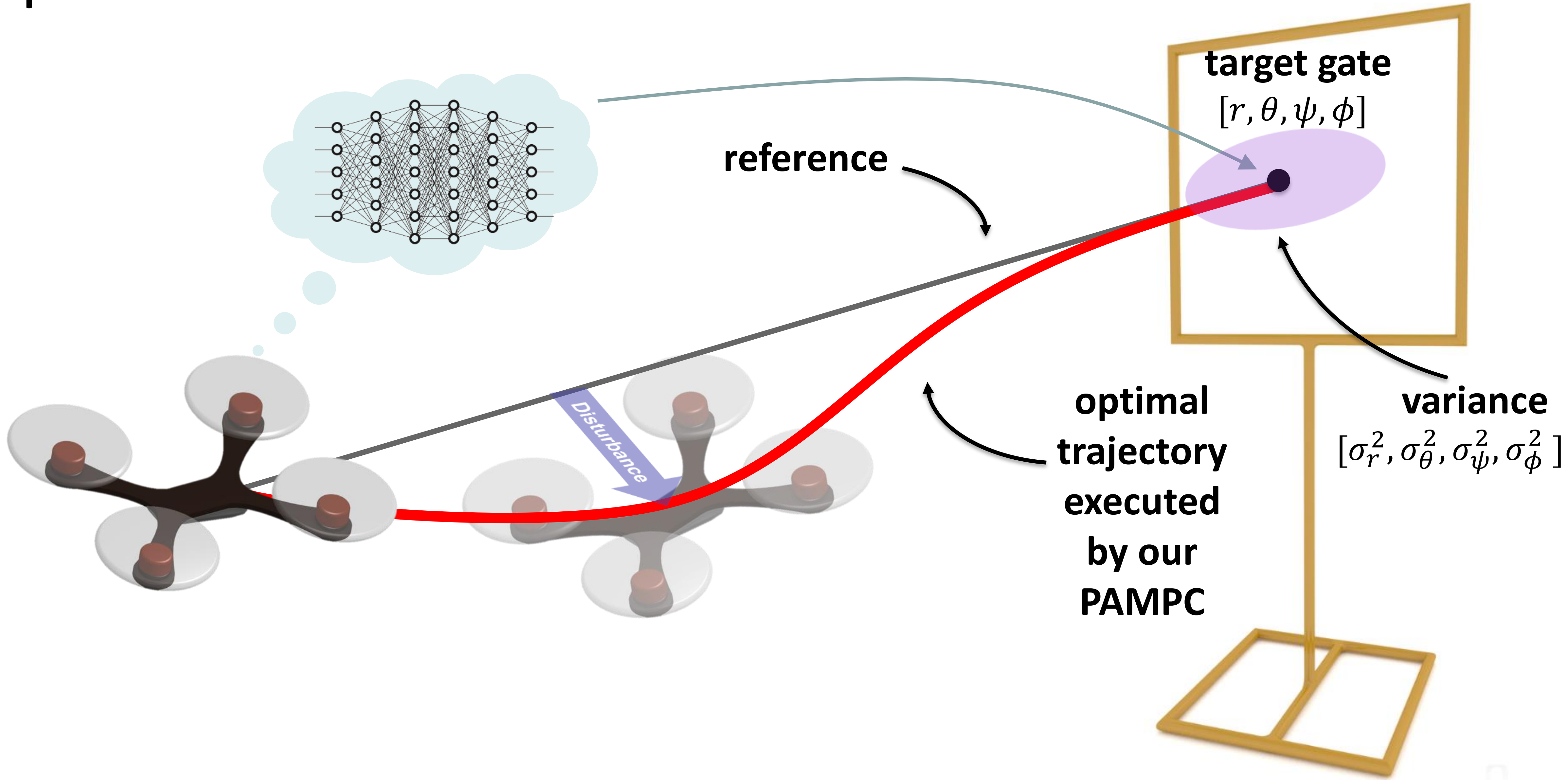

René Ranftl


Alexey Dosovitskiy


Vladlen Koltun

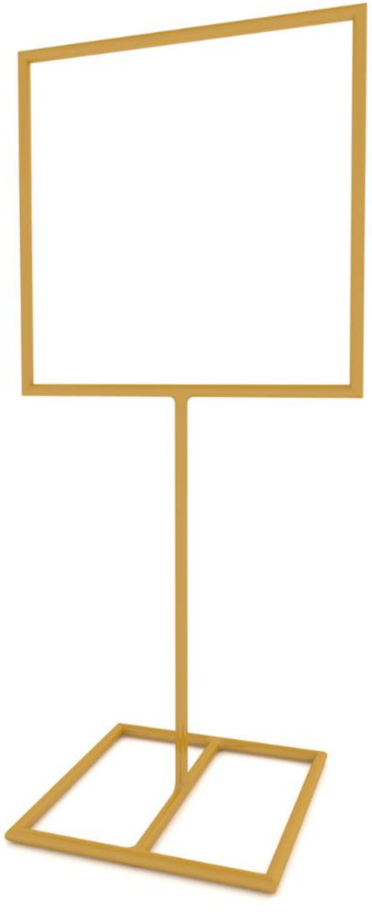

Davide Scaramuzza

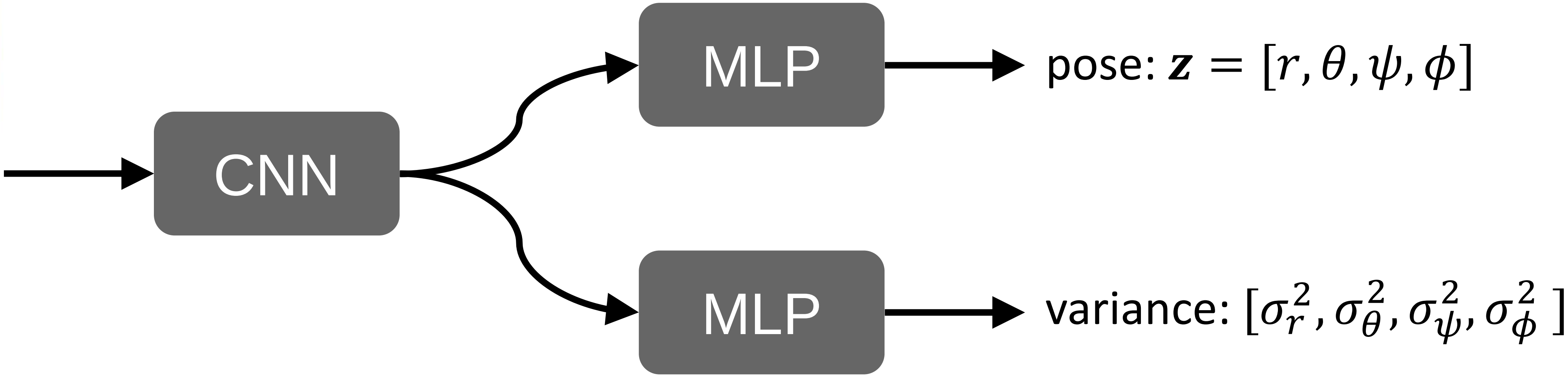
Approach Overview



Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video
Deployed to win the IROS Autonomous Drone Racing Competition, *IROS'18*. Video.

DNN to Predict Gate Pose and Uncertainty





Pose Mean Loss: $\{\theta^*_{CNN}, \theta^*_{\boldsymbol{z}}\} = \arg\min \sum_{i=1}^{N} \|\boldsymbol{y}_i - \boldsymbol{z}_i\|_2^2$

Covariance Loss[1]: $-\log p(\boldsymbol{y}\,|\boldsymbol{z}_i, \boldsymbol{\sigma}^2) \propto \sum_{j=1}^{M} \log \sigma_j^2 + \frac{(y_j - z_j)^2}{\sigma_j^2}$

[1] Nix and Weigend, Estimating the mean and variance of the target probability distribution. In IEEE International Conference On Neural Networks, 1994.

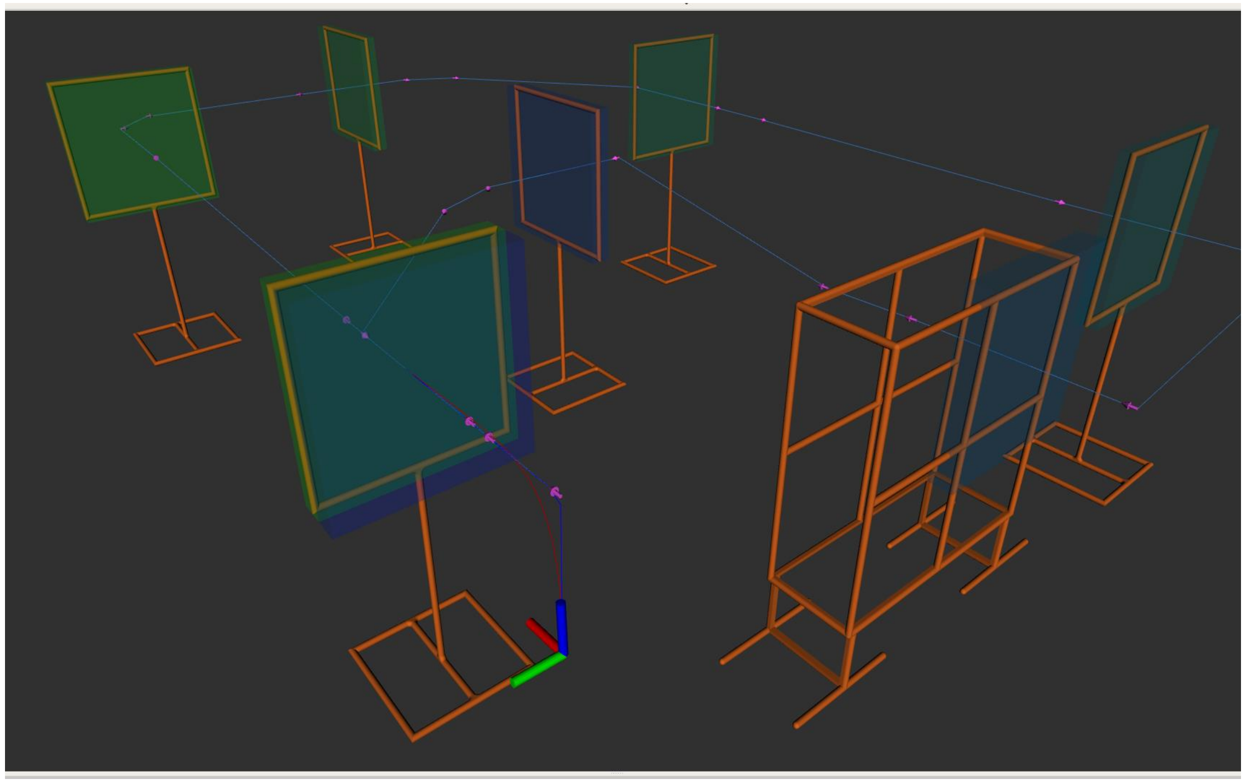
Training Data

- We collect a set of images from the forward-facing camera on the drone and associate each image with the **relative pose** of the gate with respect to the body frame of the quadrotor.
- We leverage the onboard state estimation of the quadrotor to **automatically** generate **labelled data**.

Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video
Deployed to win the IROS Autonomous Drone Racing Competition, *IROS'18*. Video.

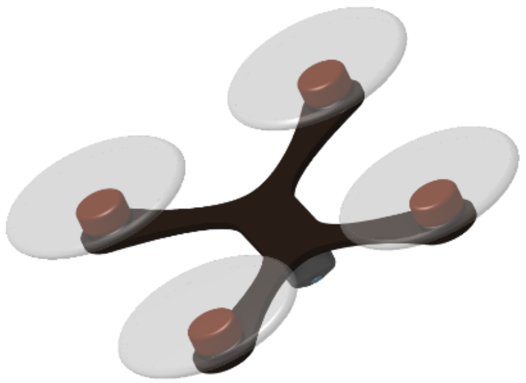
Kalman Filter for Gate Pose

- Each gate is represented by a **separate** Kalman filter that fuses prior pose and new measurements **over time.**
- The **initial prior** is the approximate gate position.
- The **process model** is an identity.
- A **reference trajectory** is computed through all the **filtered gate poses** (e.g., minimum snap, minimum time, etc.)



Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video
Deployed to win the IROS Autonomous Drone Racing Competition, *IROS'18*. Video.

Model Predictive Control



Model Dynamics

$$\boldsymbol{x} = [\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{v}]$$
$$\boldsymbol{u} = [\boldsymbol{f}_T, \boldsymbol{\omega}]$$

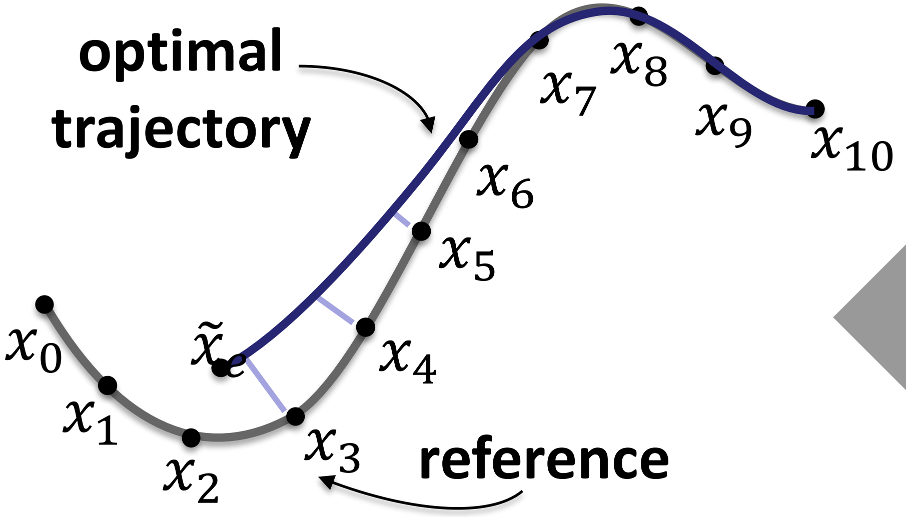
Costs

$$L_a = \boldsymbol{x}^T \boldsymbol{Q}_x \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{Q}_u \boldsymbol{u}$$

Optimization Problem

$$\min \int_{t_0}^{t_f} L_a(\boldsymbol{x}, \boldsymbol{u}) + L_p(\boldsymbol{z})\, dt$$
$$subject\ to \quad \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{z}) = \boldsymbol{0}$$
$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{z}) \leq \boldsymbol{0}$$

Sequential Quadratic Programming iterating at 100Hz

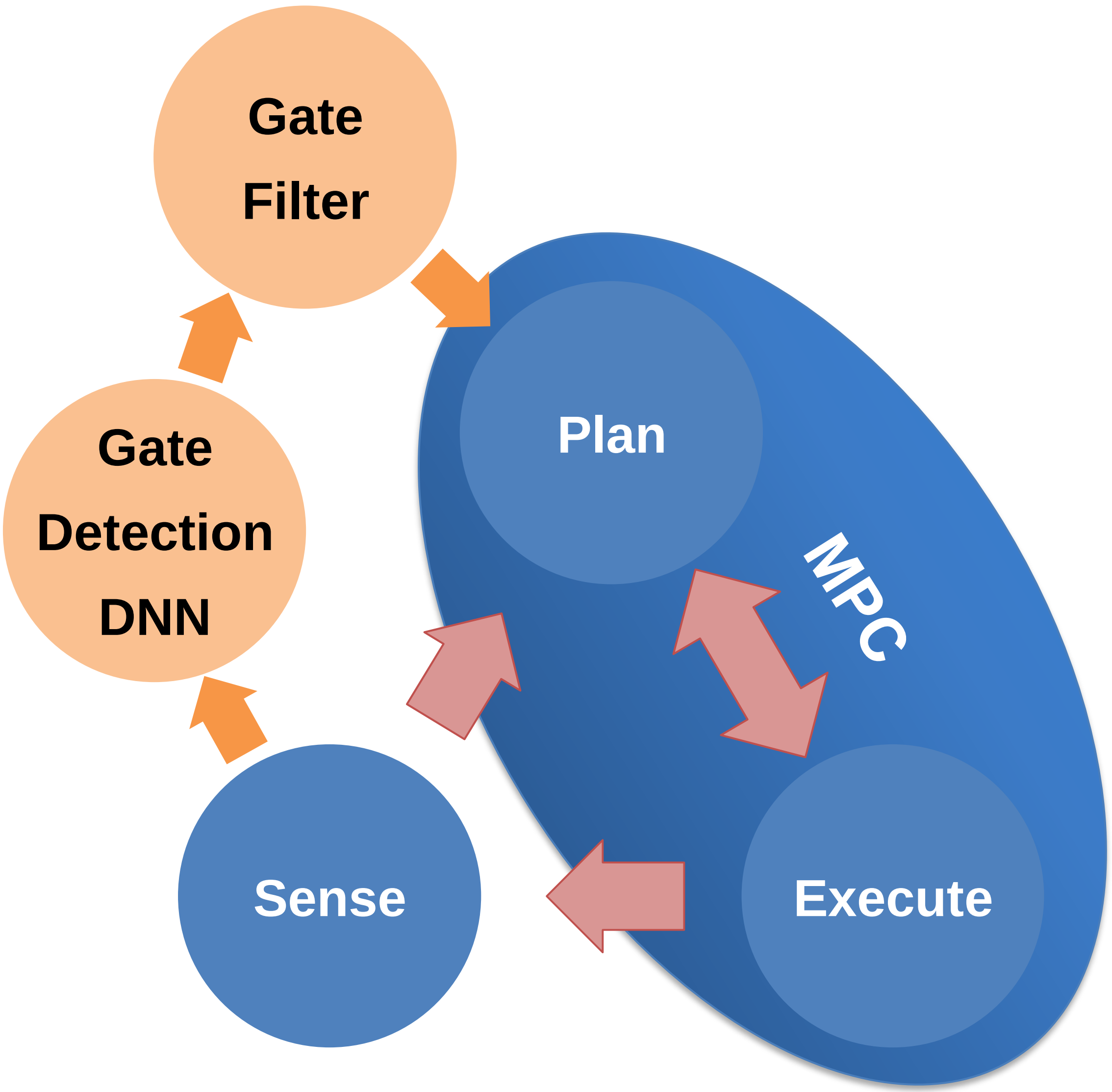


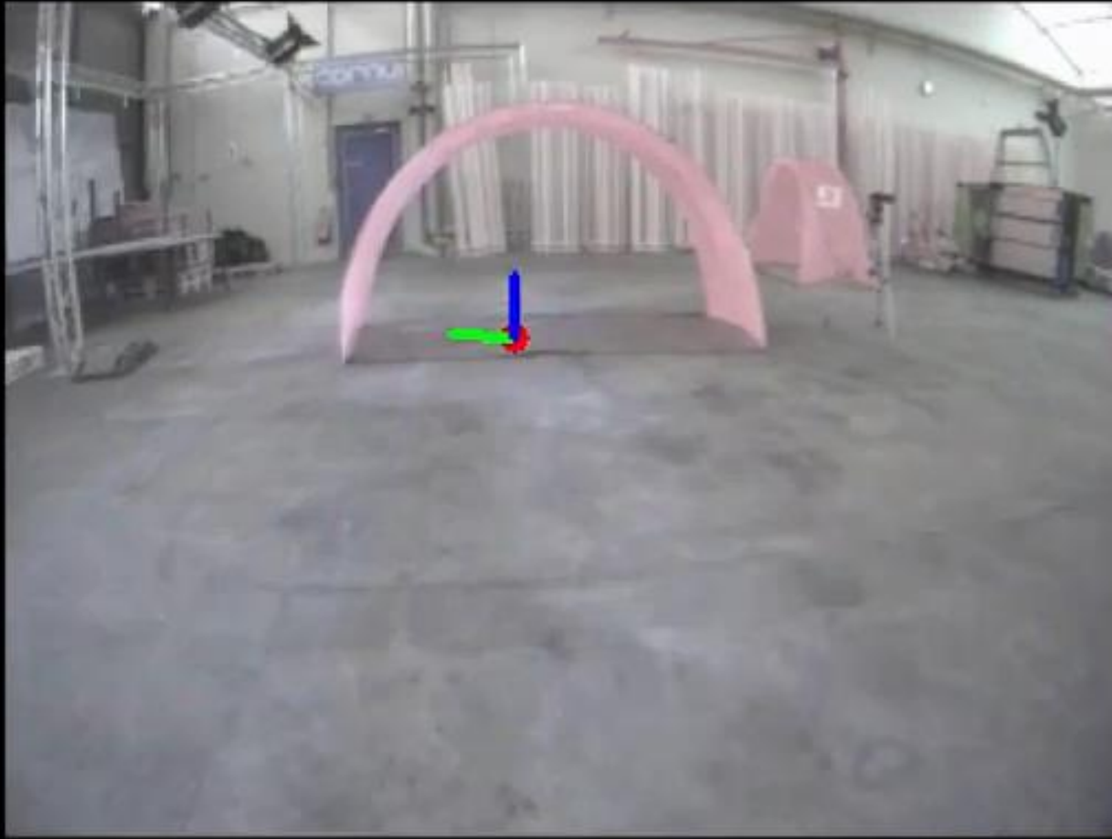
Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video

Deployed to win the IROS Autonomous Drone Racing Competition, *IROS'18*. Video.

Let’s sum it up

- **DNN** predicts relative gate pose and measurement covariance
- **Kalman Filter** fuses measurement and prior map via covariance over time
- **MPC** generates **feasible** predictions and commands simultaneously
- Allows for **reactive** and **stable** control of dynamic systems with high-level **DNN**

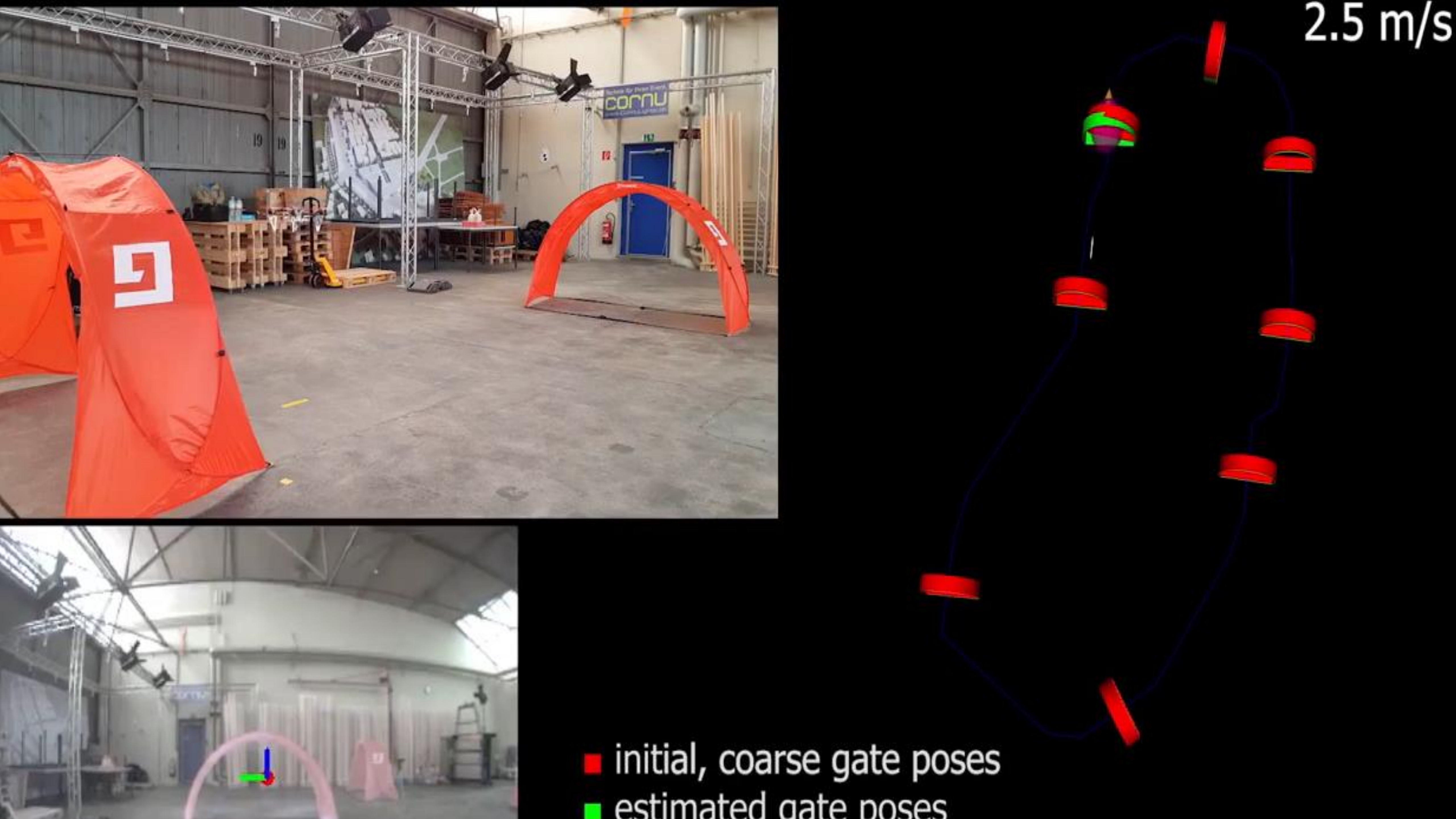




3rd person view

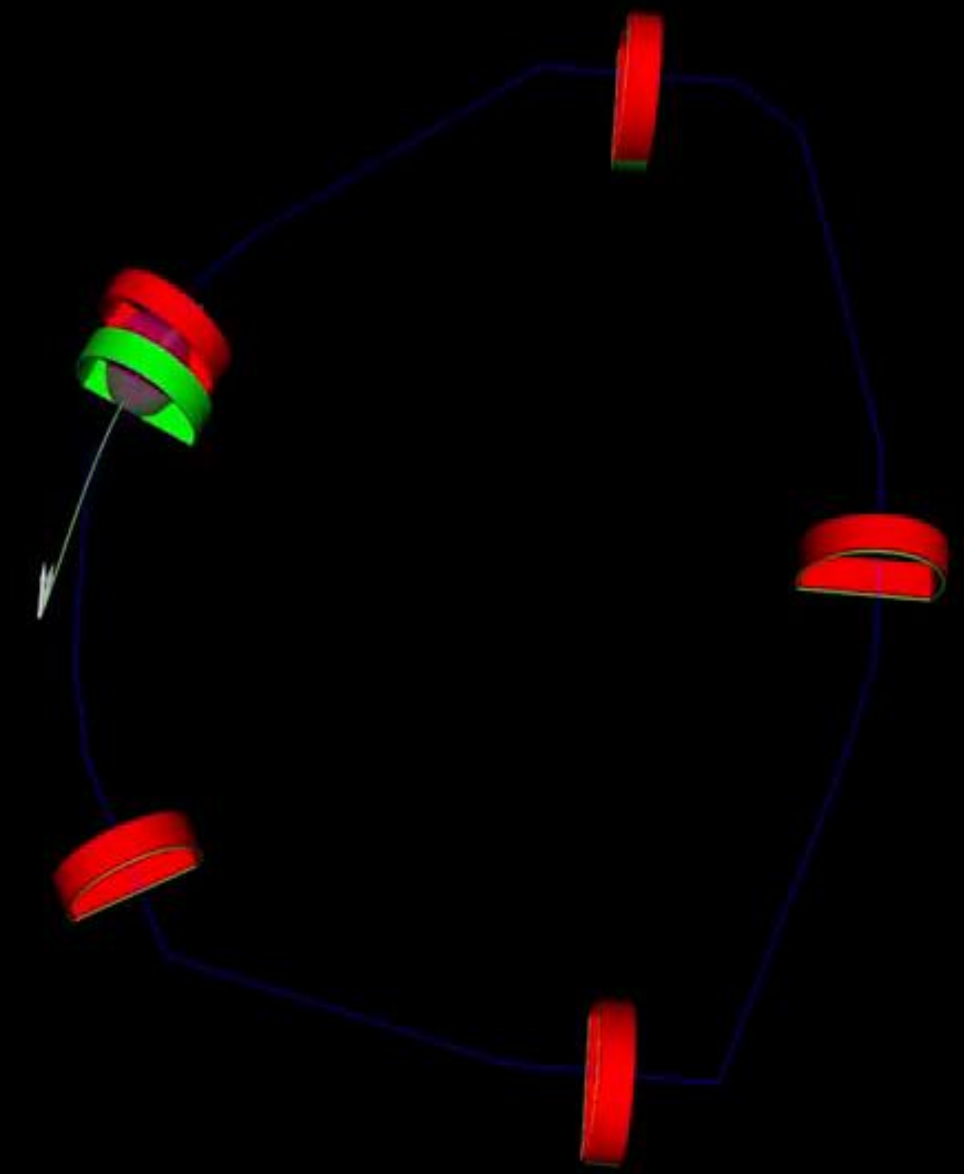
1st person view
(predicted gate pose overlaid)

Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video
Deployed to win the IROS Autonomous Drone Racing Competition, *IROS'18*. Video.



2.5 m/s
cornu
initial, coarse gate poses
estimated gate poses

2.5



■ initial, coarse gate poses

■ estimated gate poses

IROS

IROS2018 AWARD
for
Autonomous Drone Racing
First Place

University of Zurich

Philipp Föhn, Elia Kaufmann, Julien Kohler, Davide Falanga, Mathias Gehrig, Davide Scaramuzza

This Certificate is presented to the competitors in recognition of the first place in Autonomous Drone Racing Competition in

INTERNATIONAL CONFERENCE on INTELLIGENT ROBOTS and SYSTEMS 2018

Madrid, Spain
October 3, 2018

Carlos Balaguer
IROS2018 General Chair

Hyungpil Moon
IROS2018 ADR Competition Chair

October 3, 2018 – Winning the IROS 2018 Autonomous Drone Race, outracing the second-placing team by a factor of two. Video.

What about Data Generation?

- Requires a **laborious and error-prone** process
- Should be **repeated** for every **new environment/gate**

What about training a policy in simulation?

Deep Drone Racing:
From Simulation to Reality with Domain Randomization

Arxiv, 2019

PDF. Video

Elia Kaufmann

Antonio Loquercio

René Ranftl

Alexey Dosovitskiy

Vladlen Koltun

Davide Scaramuzza

Related work

Sadeghi RSS’17
(S. Levine’s lab)

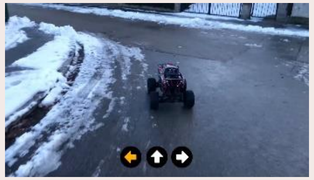
CAD2RL: Real single-image flight without a single real image

James CoRL’17
(A. Davison’s lab)

Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task

Müller CoRL’18
(V. Koltun’s lab)

Driving policy transfer via modularity and abstraction

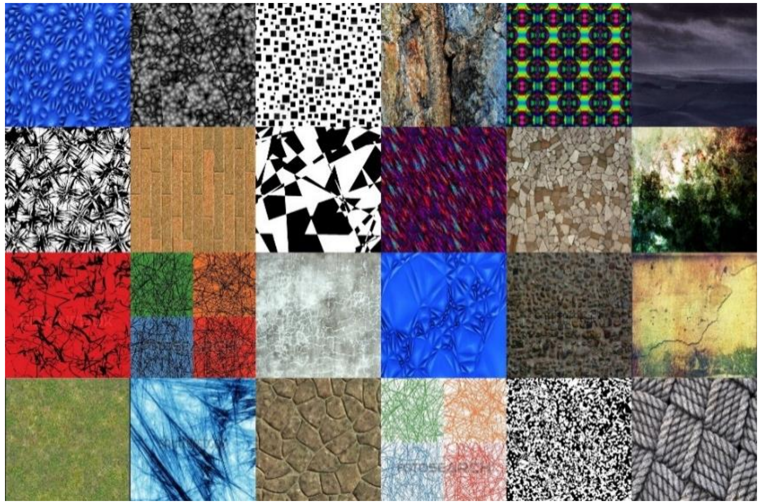


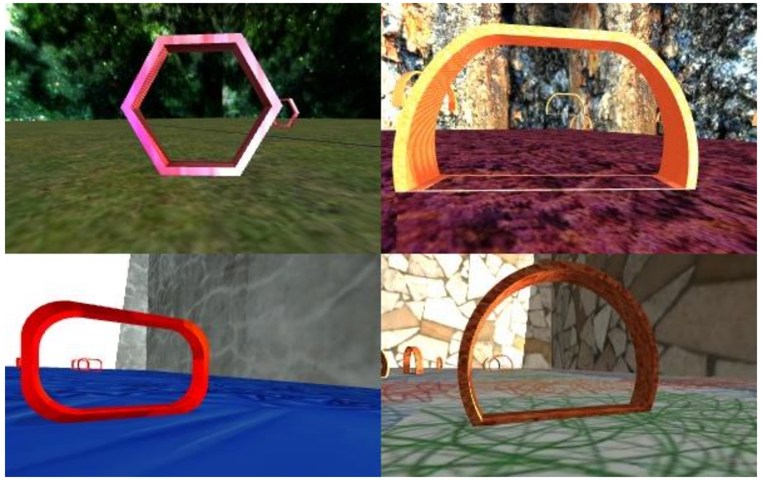
Simulation Data Generation

To achieve maximum transfer abilities, we perform **domain randomization.**

Randomized features are:

- Illumination
  - Ambient light property uniformly sampled in [0,1]
  - Emissive light property uniformly sampled in [0,0.3]
- Gate Appearance
  - Shape sampled from a set of 6 diverse models
  - Color sampled from a set of 10 red/orange textures
- Background & Floor
  - Sampled from a pool of 30 diverse images



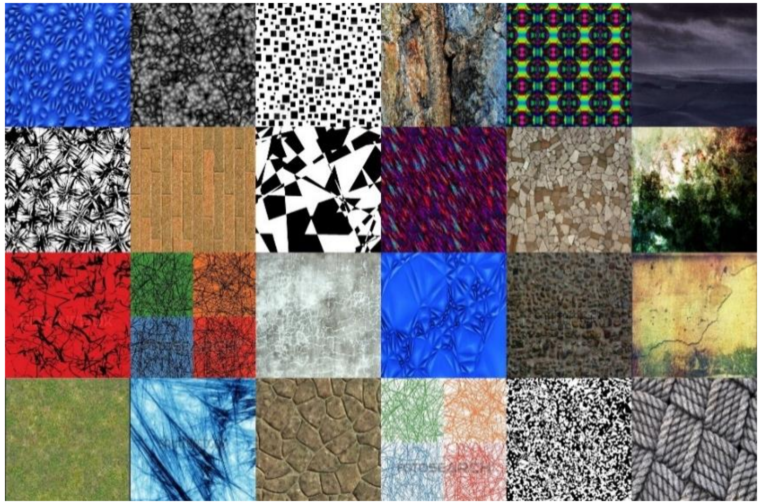


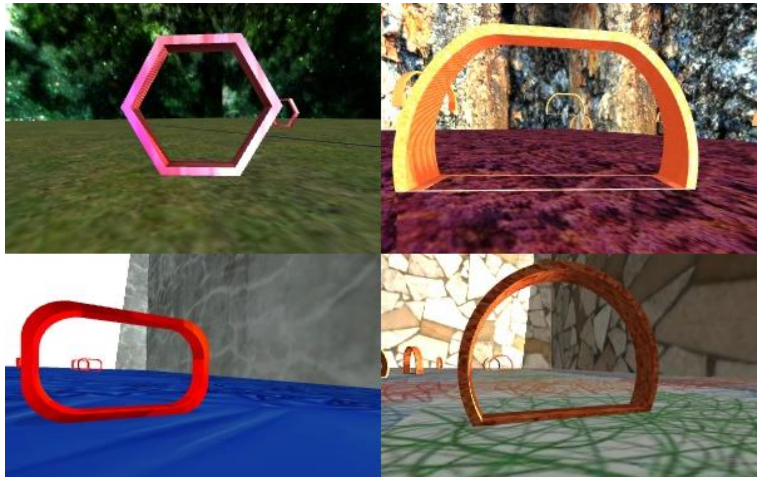
Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.

Simulation Data Generation

We generated a total of **100K simulated samples**.

Each sample corresponds to the first person view camera image plus the associated expert annotation (slide 8).



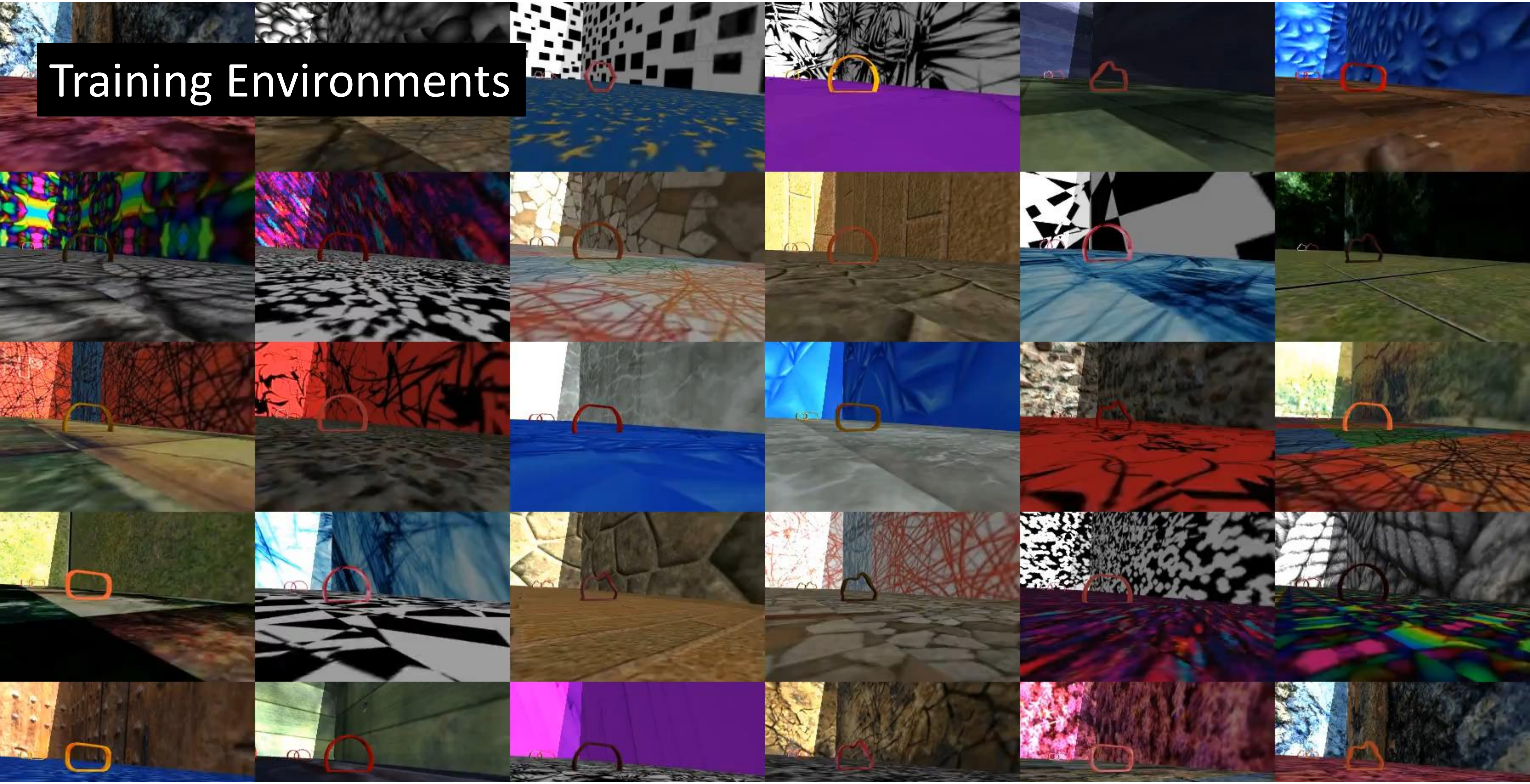


Data was generated according to these parameters:

- 90 experiments (each with randomized features)
- 1 lap per experiment (90s each)
- Samples saved at 12Hz

Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.

Training Environments



Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.

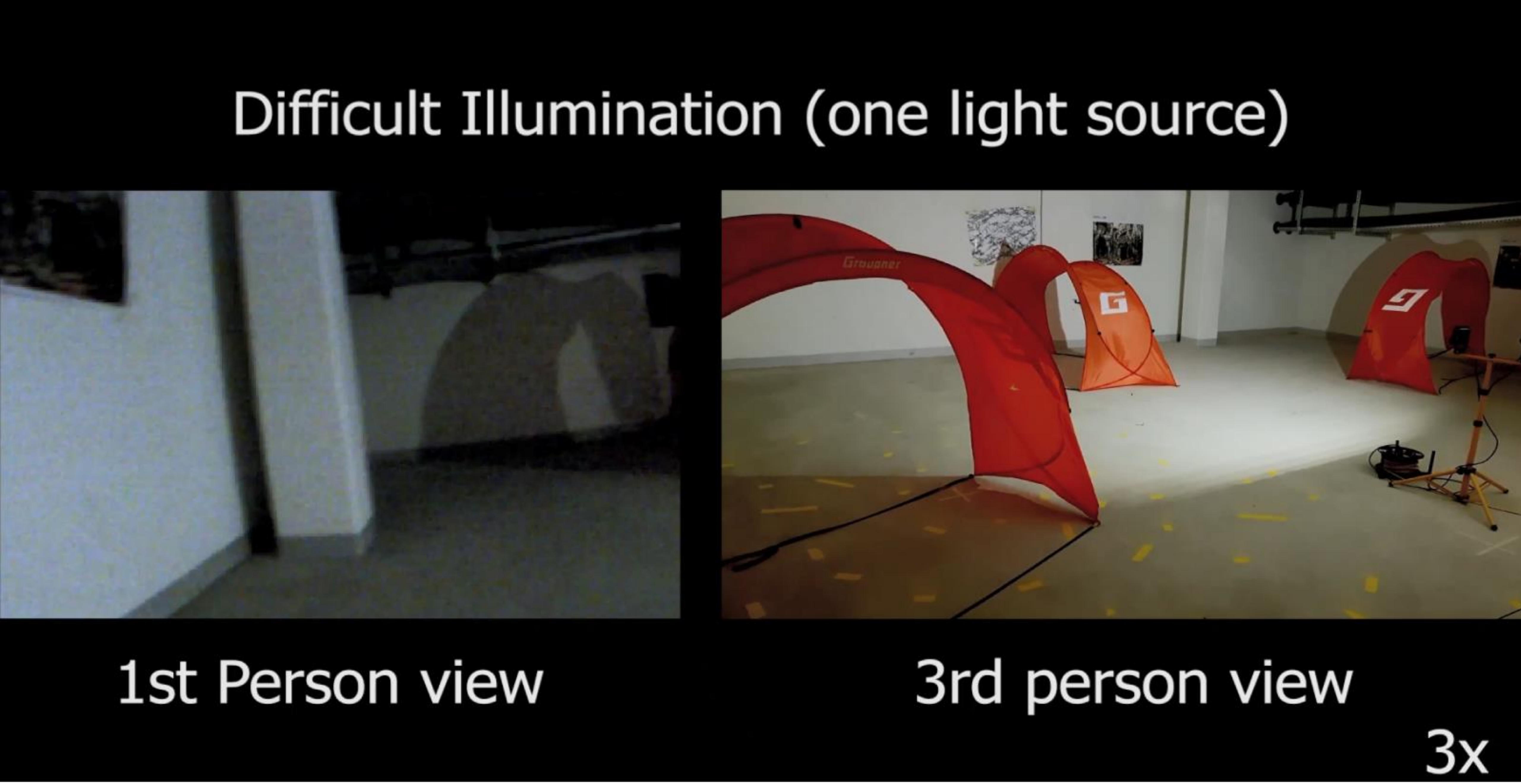
Transfer to the Real Platform

The policy **trained in simulation** is deployed on **a real platform**.

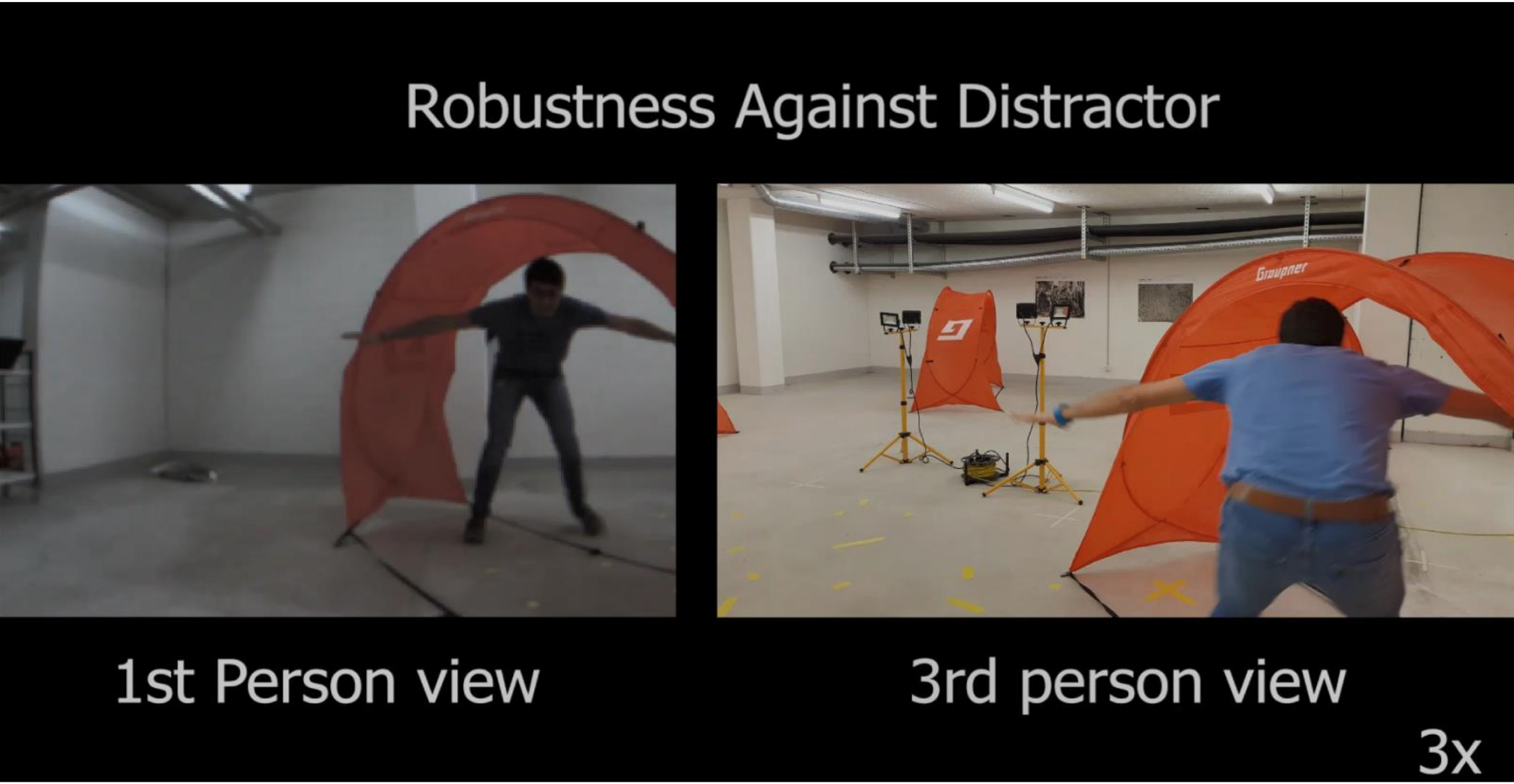
We evaluate the **robustness** against:

- changes in the **illumination**
  - Easy (Uniform)
  - Medium (2 light sources)
  - Hard (1 light source)
- **Distractors**
  - Field of View partially covered

Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.



Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.



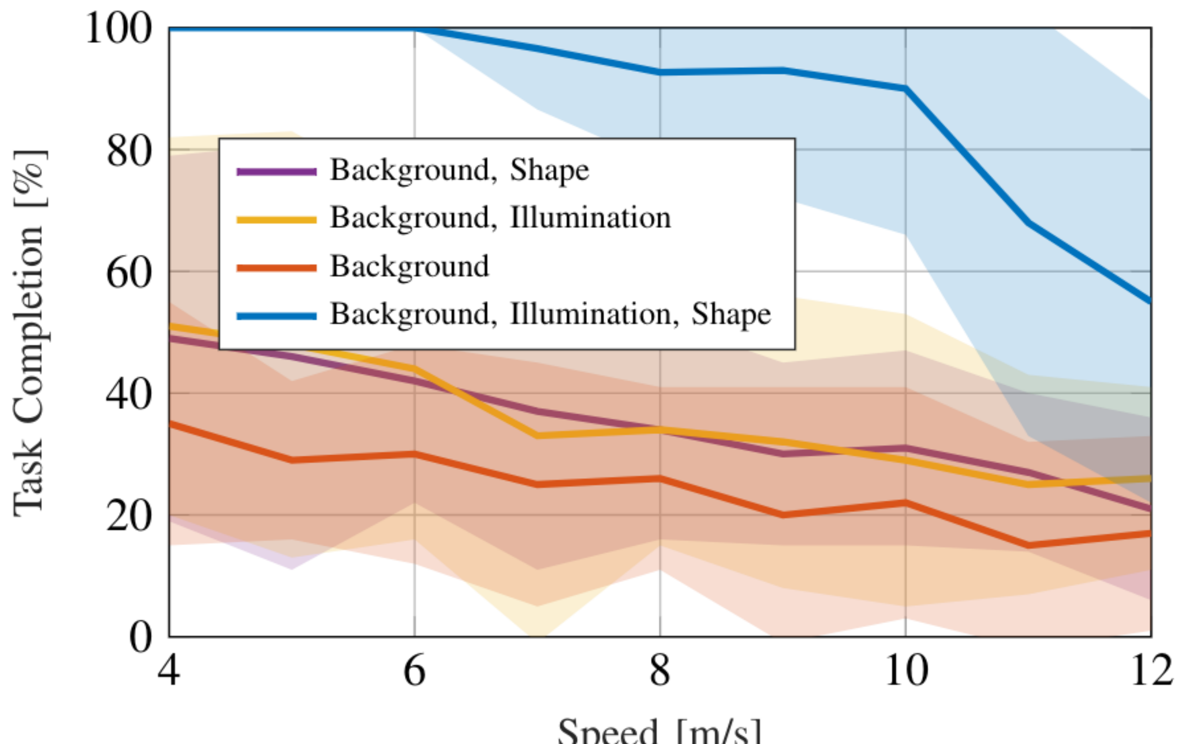
Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.

Quantitative Results: Sim2Sim transfer

We evaluated the performance of a policy trained in simulation in an environment ***unseen*** at training time (background, illumination and gate shape changes).

Measure: Task Completion, which is 100% when 5 laps are completed without crashing.

Main finding: In order to generalize to previously unseen setups, it is necessary to randomize all the features together.

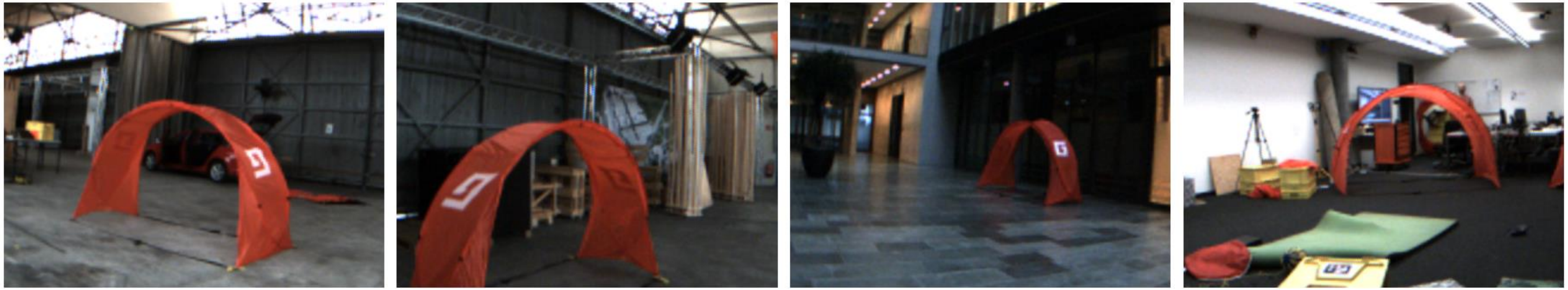


Quantitative Results: What is important for transfer?

We evaluated which features are the most important to unlock transfer.

Measure: RMSE on real world data (annotated by the model-based expert in hand-held mode).

	Illumination		No Illumination	
	Texture	No Texture	Texture	No Texture
Shape	0.199	0.213	0.243	0.311
No Shape	0.207	0.225	0.265	0.339



Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. [PDF](). [Video]().

What is important for transfer? Main Findings

It is possible to make the following observations:

- Illumination is the most important of the randomization factors.
- Gate shape randomization has the least effect.

Main reasoning: Gates are very similar in simulation and real world, but environments' characteristics are very different in the two domains.

	Illumination		No Illumination	
	Texture	No Texture	Texture	No Texture
Shape	0.199	0.213	0.243	0.311
No Shape	0.207	0.225	0.265	0.339

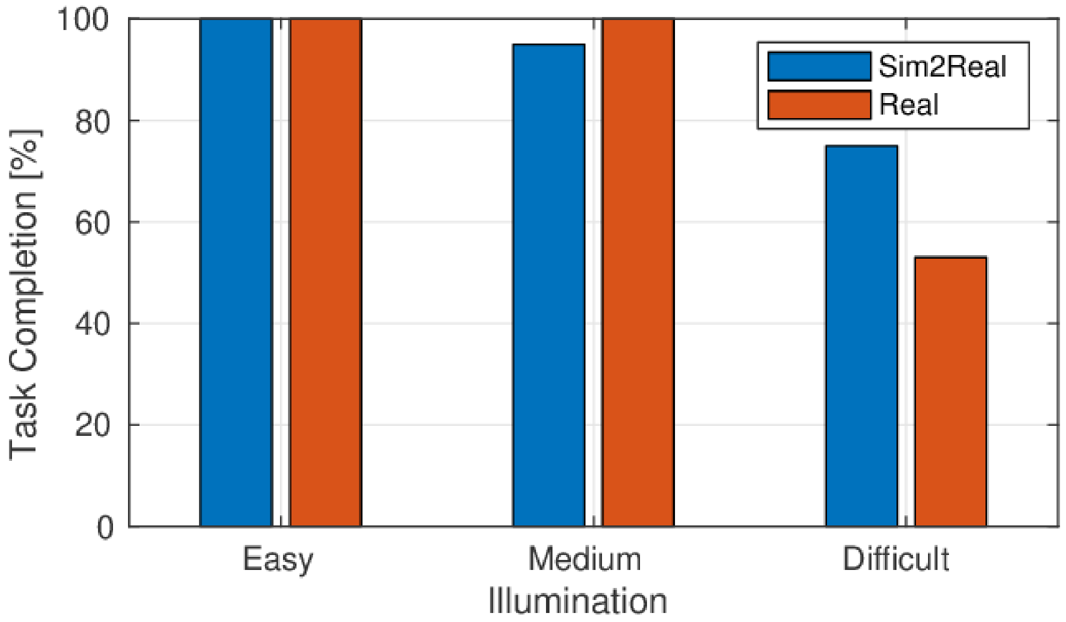
Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.

Quantitative Results: Simulated vs Real World Data

We compared the performance of a policy trained in simulation and one trained from real world data collected from the test track.

Measure: Task Completion, 100% if 5 laps are completed without crashing.

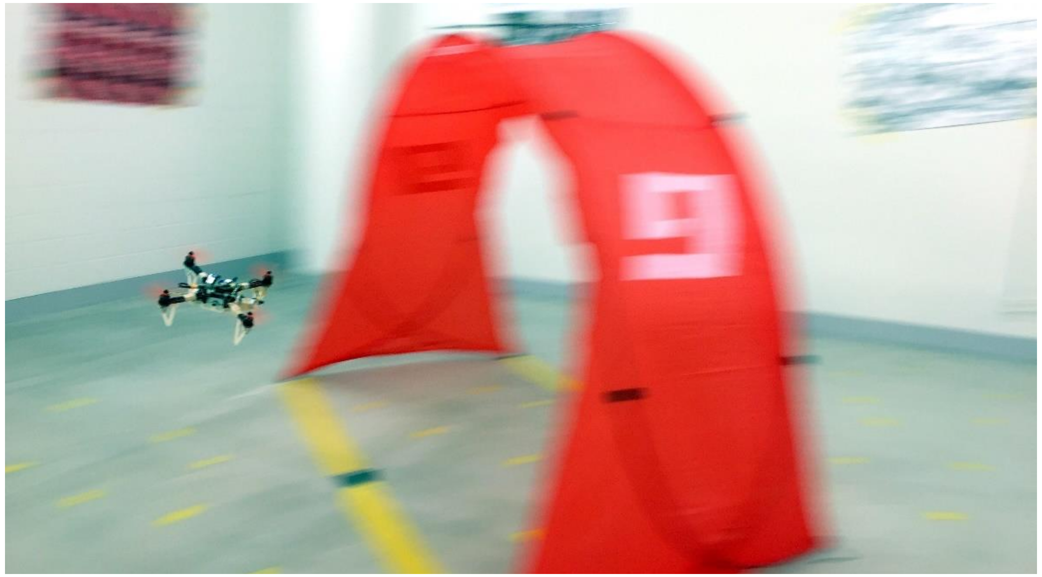
Main Conclusion: The simulated policy outperforms the real one where the latter has no sufficient data coverage.

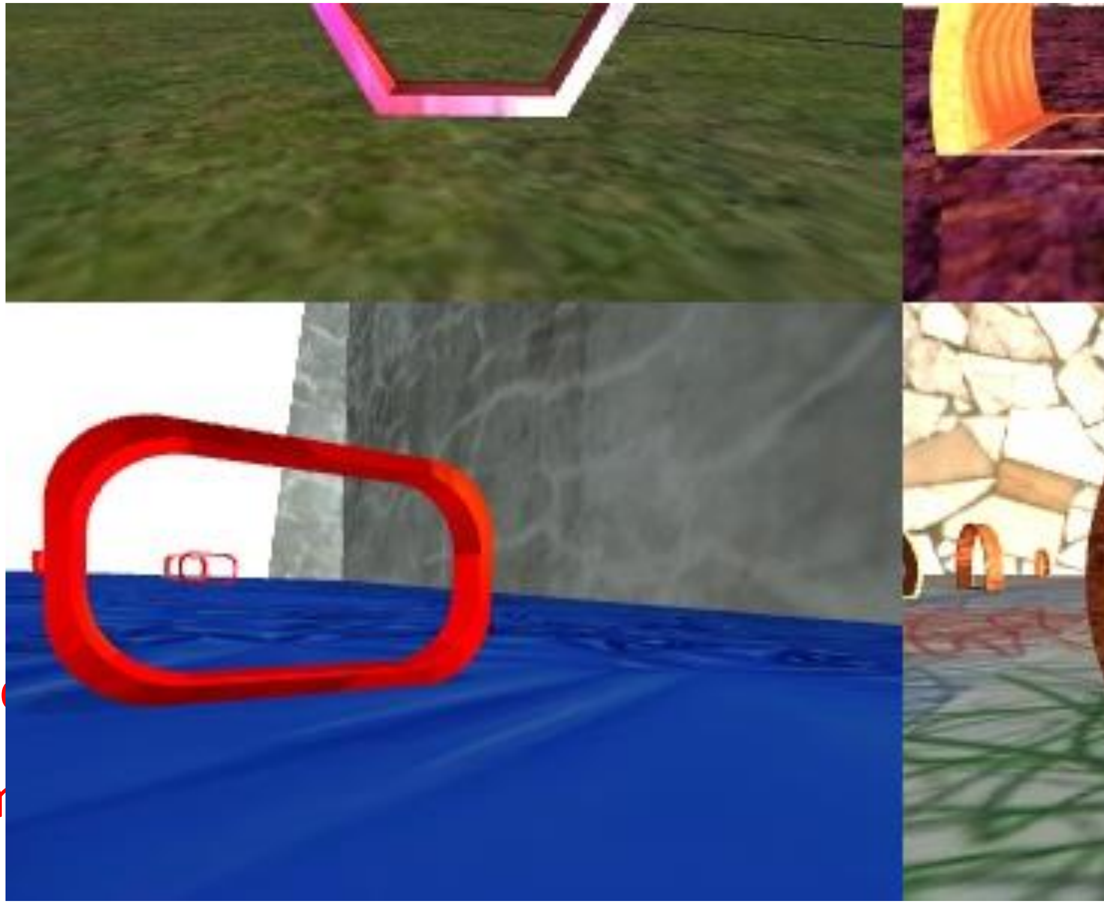


Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.

Sim2Real for Drone Racing

- Does not need to collect data for every new environemnt
- Robust to changes in the environemnt





- Limit
  optim

Loquercio, et al., *Deep Drone Racing with Domain Randomization*, Arxiv, 2019. PDF. Video.

Conclusions

- Hybrid systems based on machine learning (for perception) and models (for control) are more robust than systems based on exclusively one of the two.
- Given enough data, it is possible to train a robust perception system with data generated only in simulation.
- Several technical challenges have to be addressed before reaching super-human performance:
  - Low-latency Control & Perception
  - Long Term Planning
  - Adaptive Perception