Research, Development and Engineering Division

Towards **Self-Learning Self-Driving Vehicle**: reinforcement learning system for autonomous driving

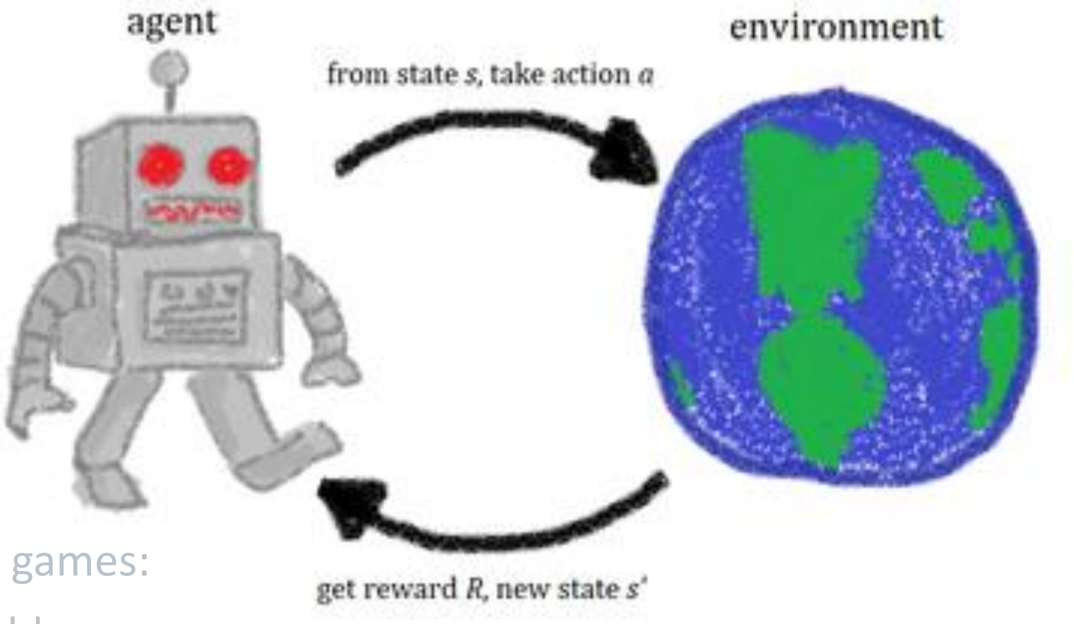
Dr. Refael Vivanti

RAFAEL

What is Reinforcement Learning

- Learning from self-experience.

- RL can learn:
  - Unknown game rules
  - Delayed rewards, no supervision
  - Actions affect the environment



- Recent achievements:
  - Superhuman performance in many games:
  - Chess, Go, Atari games, Control problems, …

What is Reinforcement Learning

- Learning from self-experience.

- RL can learn even if:
  - Unknown game rules
  - Delayed rewards, no supervision
  - Actions affect the environment

- Recent achievements:
  - Superhuman performance in many games:
  - Chess, Go, Atari games, Control problems, …

The mission

- Teach a vehicle to drive in a specific off-road area.
  - Will be tested in the training area only.
- Supervised Learning Requires large annotated datasets.
- Physical RL is too slow and unsafe
- **Solution: Copy-Paste the environment!**
  - realistic 3D modelling from aerial images
  - RL Training inside the model
  - Bonus: driving in currently un-approachable areas.

3D Model of the training area

The mission

- Teach a vehicle to drive in a specific off-road area.
  - Will be tested in the training area only.
- Supervised Learning Requires large annotated datasets.
- Physical RL is too slow and unsafe
- **Solution: Copy-Paste the environment!**
  - realistic 3D modelling from aerial images
  - RL Training inside the model
  - Bonus: driving in currently un-approachable areas.

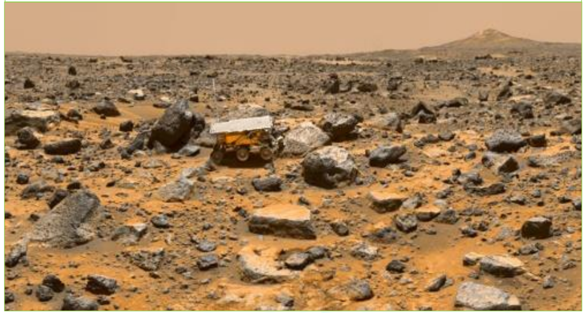
3D Model of the training area

The mission

- Teach a vehicle to drive in a specific off-road area.
  - Will be tested in the training area only.
- Supervised Learning Requires large annotated datasets.
- Physical RL is too slow and unsafe
- **Solution: Copy-Paste the environment!**
  - realistic 3D modelling from aerial images
  - RL Training inside the model
  - Bonus: driving in currently un-approachable areas.

3D Model of the training area

Driving in unvisited area



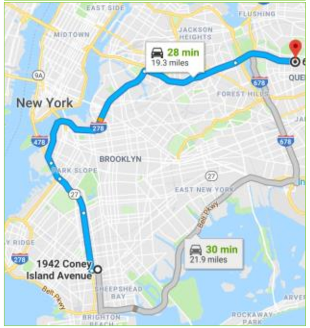
The challenge

- Driving involves two tasks: **Navigation** and **Avoidance**
  - Both affect location and pose

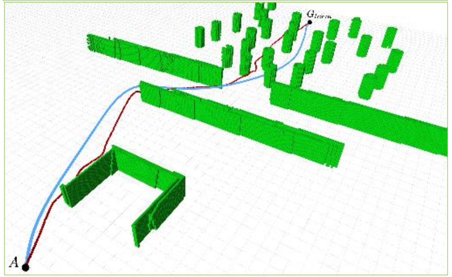
Navigation – Path planning	Obstacle avoidance
strategical task	tactical task
sparse rewards	dense rewards
geometric input	visual input

- Doing both together is hard

Navigation



Avoidance



Our Solution: split

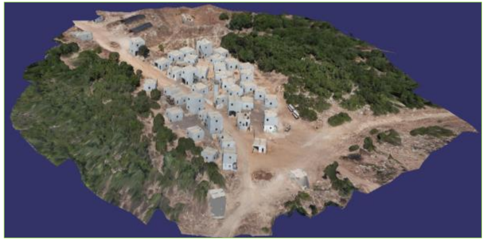
Navigation - Before mission

Current location

Target location

Navigation system

3D Model



Track

Our Solution: split

Navigation - Before mission

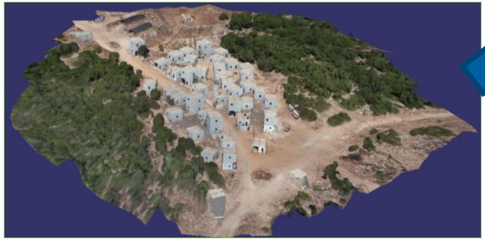
Current location

Target location

Navigation system

Track

3D Model



Rendering

Avoidance - During mission

Rendered Image

RL Agent

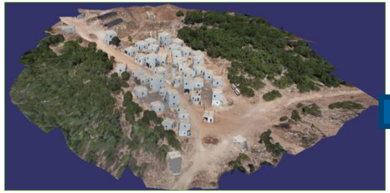
Driving Decision

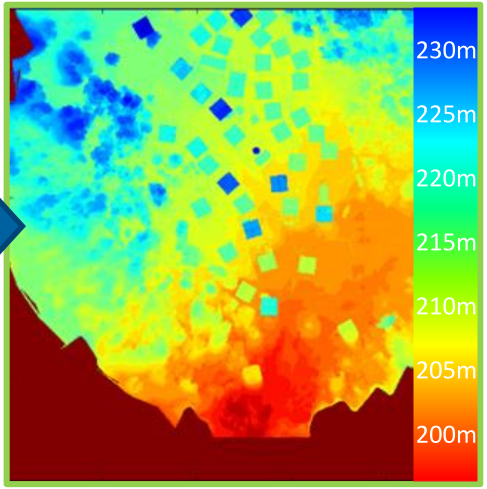
New pose

Navigation – Before mission

1. 3D model geometry

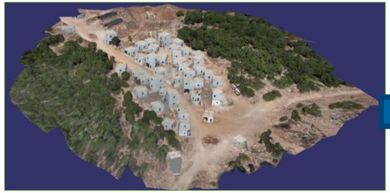
3D model



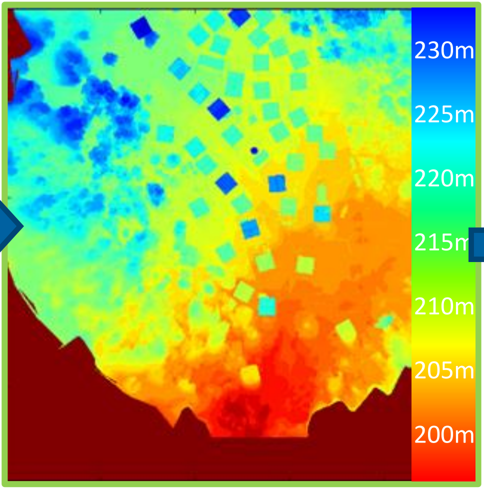


Navigation – Before mission

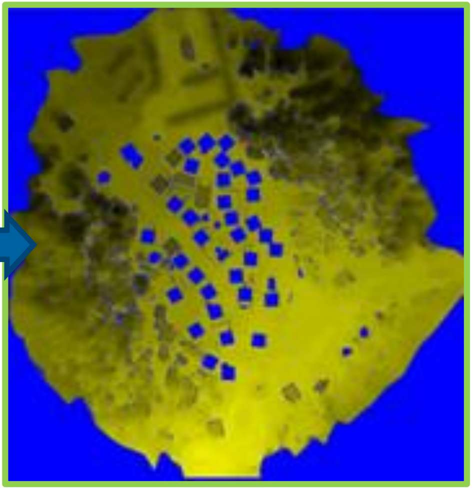
3D model



1. 3D model geometry

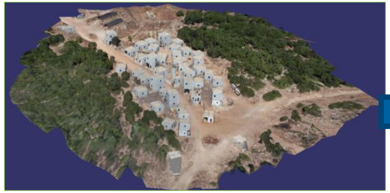


2. Segmentation by passability

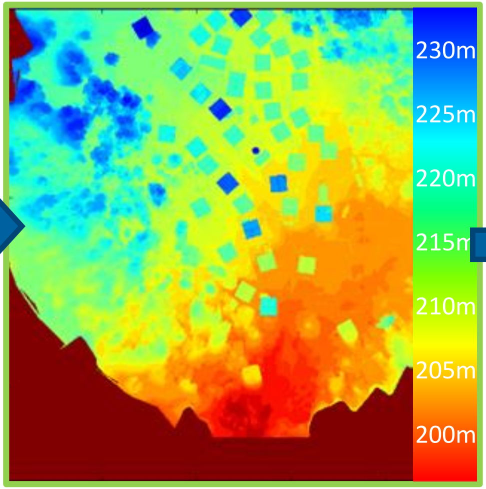


Navigation – Before mission

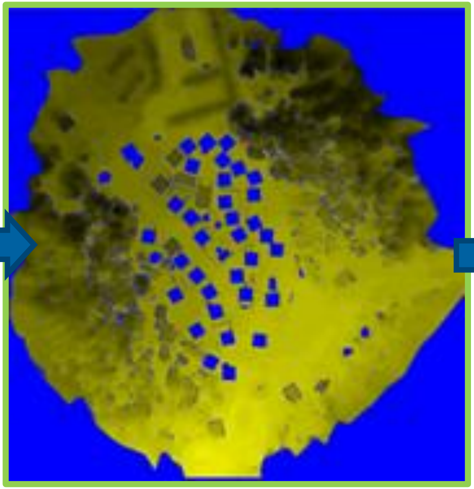
3D model



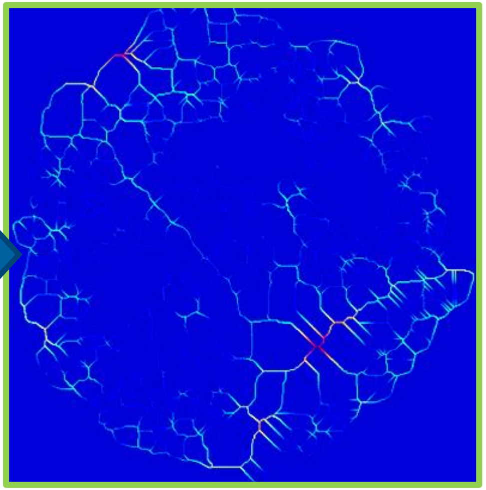
1. 3D model geometry



2. Segmentation by passability

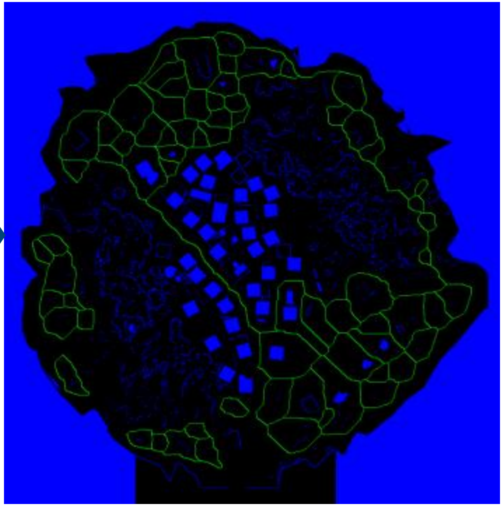


3. Watershed: trails + width



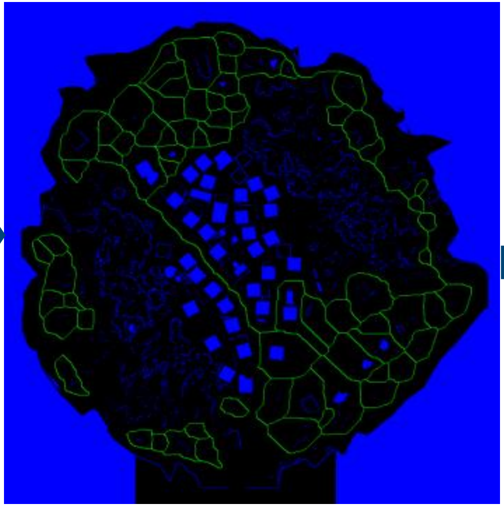
Navigation – Before mission

4. Remove narrow trails

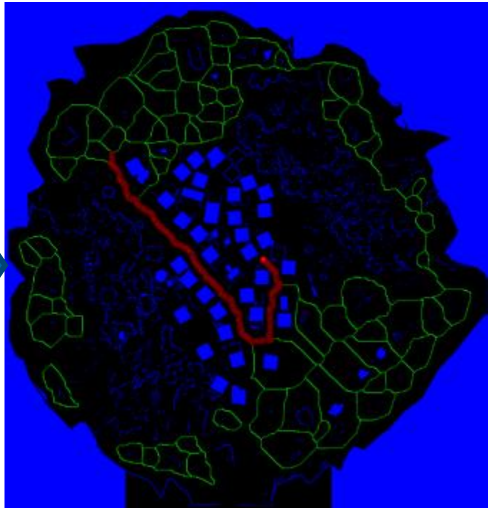


Navigation – Before mission

4. Remove narrow trails

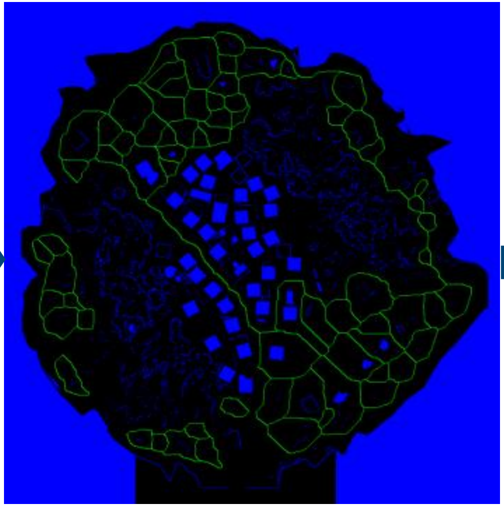


5. Shortest path

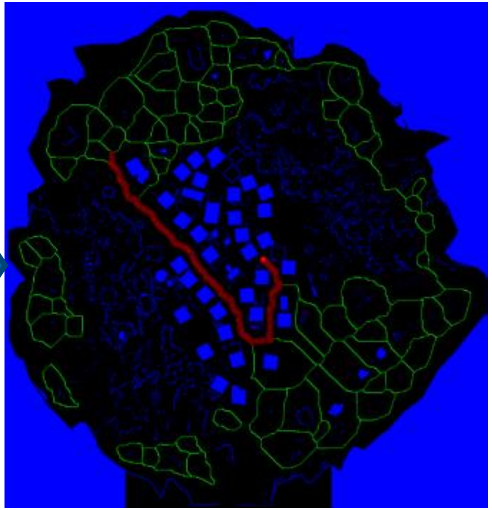


Navigation – Before mission

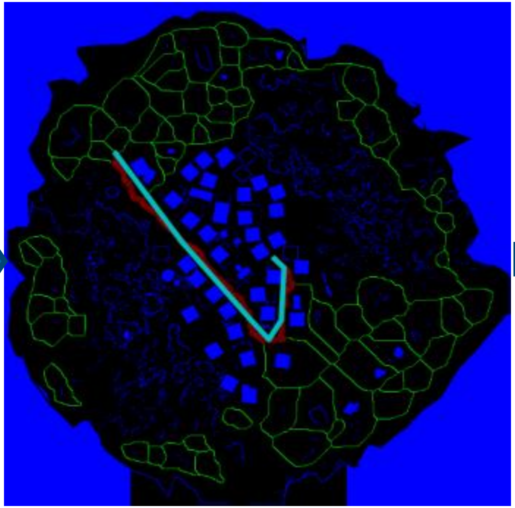
4. Remove narrow trails



5. Shortest path



6. Track summary to way-points



waypoints

Lon	Lat
32.6532	35.6686
32.6638	35.2684
32.3530	35.4683
32.8532	35.2784
32.6535	35.2694
32.6236	35.2680
32.2538	35.2683

Avoidance - During mission

- In each driving step:
  - The model is rendered to the agent location
  - The agent pose is such that the next waypoint is always in front of it.
  - The agent uses the rendered image to avoid obstacles, while "unknowingly" progress towards the target.

- We wrapped the driving simulator with a **game** with scores, and trained RL agent.

Avoidance - During mission

- In each driving step:
  - The model is rendered to the agent location
  - The agent pose is such that the next waypoint is always in front of it.
  - The agent uses the rendered image to avoid obstacles, while "unknowingly" progress towards the target.

- We wrapped the driving simulator with a **game** with scores, and trained RL agent.

Avoidance - During mission

- In each driving step:
  - The model is rendered to the agent location
  - The agent pose is such that the next waypoint is always in front of it.
  - The agent uses the rendered image to avoid obstacles, while "unknowingly" progress towards the target.
- We wrapped the driving simulator with a **game** with scores, and trained RL agent.

Avoidance - During mission

- In each driving step:
  - The model is rendered to the agent location
  - The agent pose is such that the next waypoint is always in front of it.
  - The agent uses the rendered image to avoid obstacles, while "unknowingly" progress towards the target.

- We wrapped the driving simulator with a **game** with scores, and trained RL agent.

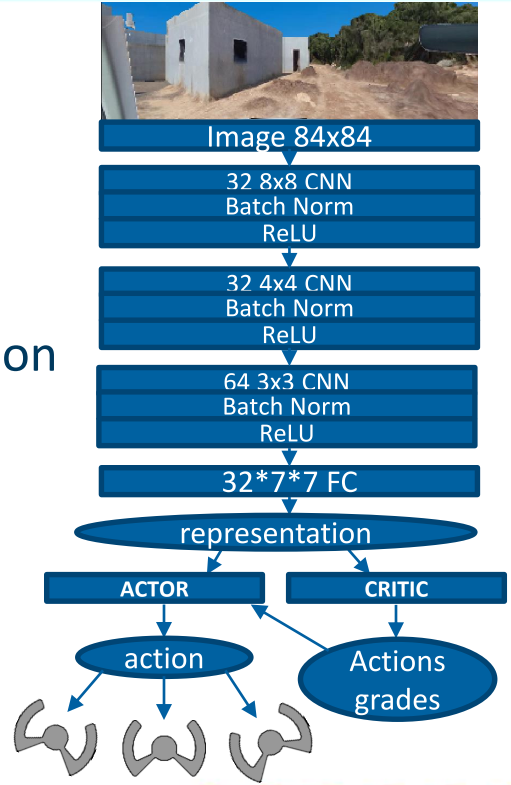
Training

- Compared 3 SOTA Actor-Critic based RL algorithms:
  - **PPO** Proximal Policy Optimization

    Schulman J. et al. "Proximal policy optimization algorithms." *arXiv:1707.06347* 2017
  - **A2C** Advantage Actor Critic

    Mnih V. et al. Asynchronous methods for deep reinforcement learning. ICML2016
  - **ACKRT** Actor Critic using Kronecker-factored Trust Region

    Wu Y. et al. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. NIPS2017
- Sharing convolutional layers between actor and critic
  - The joint network learns image representation
  - Actor and critic each use the representation differently
  - Both are 1 Fully Connected layer



Ref.:

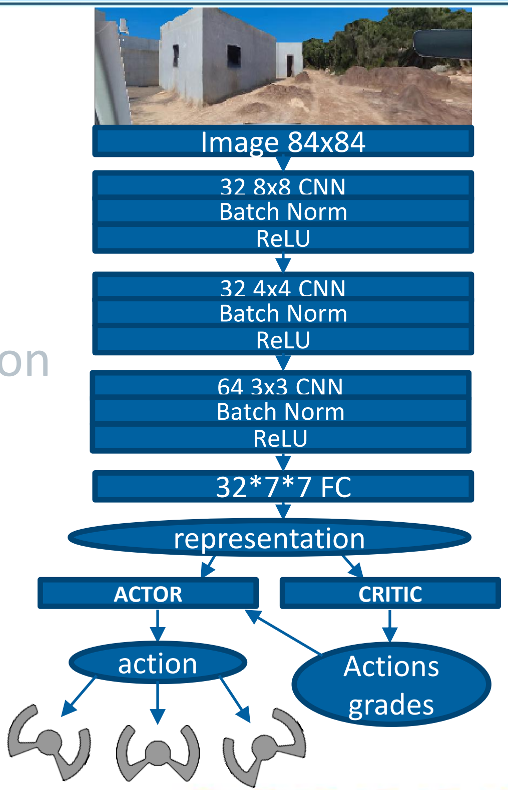
Training

- Compared 3 SOTA Actor-Critic based RL algorithms:
  - **PPO** Proximal Policy Optimization

    Schulman J. et al. "Proximal policy optimization algorithms." *arXiv:1707.06347* 2017
  - **A2C** Advantage Actor Critic

    Mnih V. et al. Asynchronous methods for deep reinforcement learning. ICML2016
  - **ACKRT** Actor Critic using Kronecker-factored Trust Region

    Wu Y. et al. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. NIPS2017
- Sharing convolutional layers between actor and critic
  - The joint network learns image representation
  - Actor and critic each use the representation differently
  - Both are 1 Fully Connected layer



Ref.:

Training

- **Random tracks**:
  - New track every game
- **Random obstacles**:
  - Same street, new parking cars
- **Multi-process**:
  - parallel games, one agent

Training

- **Random tracks**:
  - New track every game
- **Random obstacles**:
  - Same street, new parking vehicles
- **Multi-process**:
  - parallel games, one agent

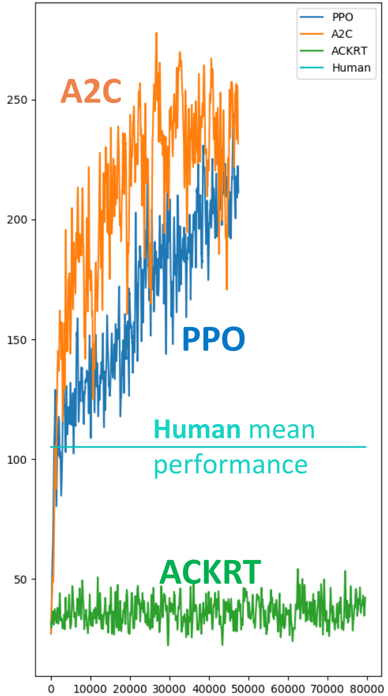
Training

- **Random tracks**:
  - New track every game
- **Random obstacles**:
  - Same street, new parking cars
- **Multi-process**:
  - parallel games, one agent

Results

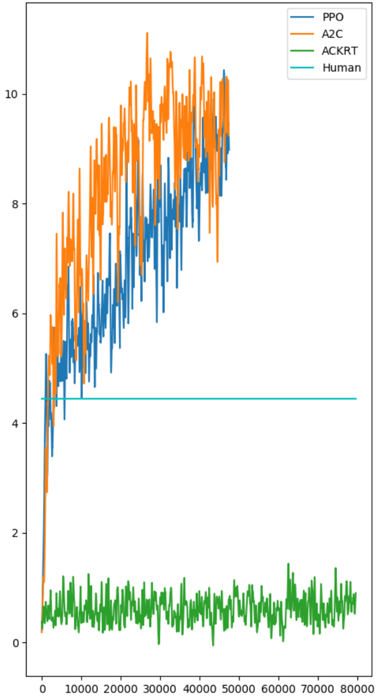
- **PPO**
  - Proximal Policy Optimization
- **A2C**
  - Advantage Actor Critic
- **ACKRT**
  - Actor Critic using Kronecker-factored Trust Region
- Mean **Human**
- Early Super-human performance
- Volatile vs Monotonic

Game Length



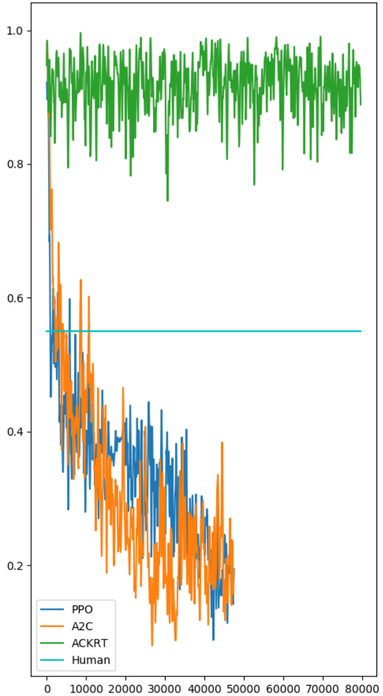
#Training Games

Game Scores



#Training Games

Crash frequency

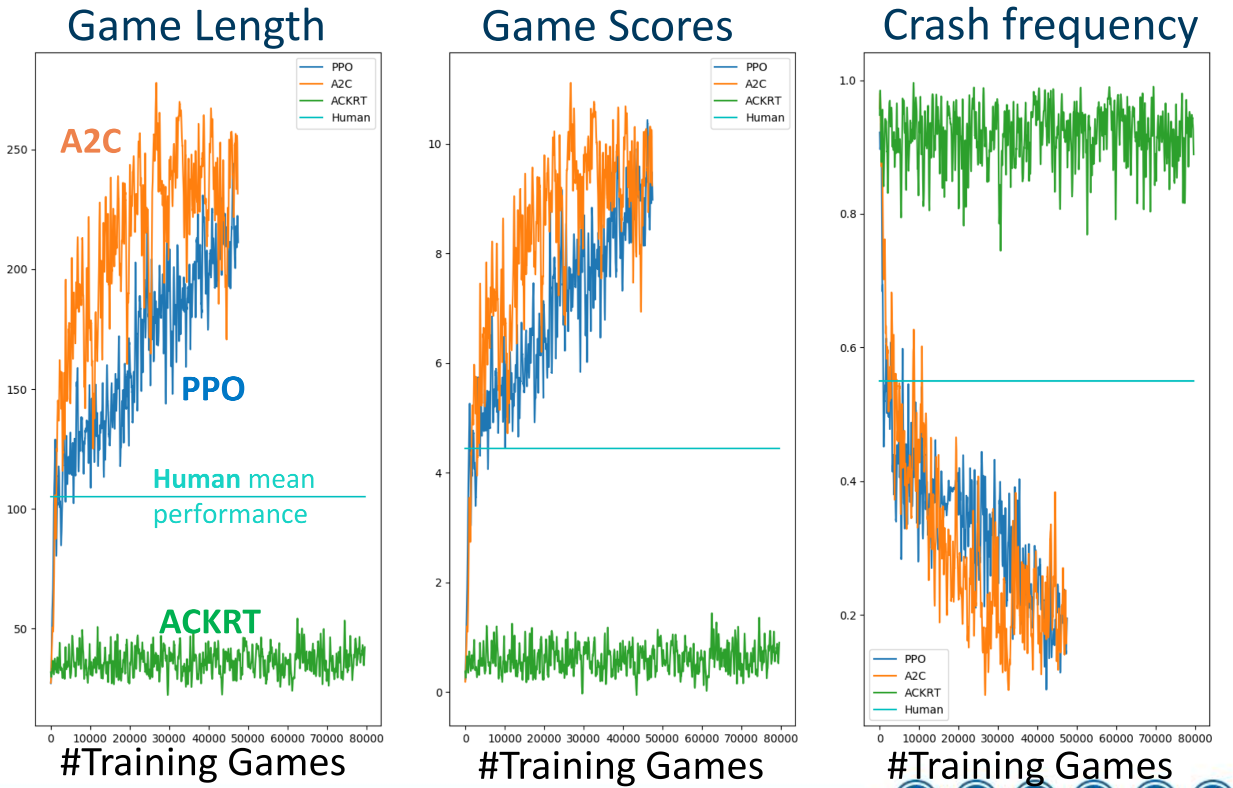


#Training Games

Ref.:

Results

- **PPO**
  - Proximal Policy Optimization
- **A2C**
  - Advantage Actor Critic
- **ACKRT**
  - Actor Critic using Kronecker-factored Trust Region
- Mean **Human**
- Early Super-human performance
- Volatile vs Monotonic

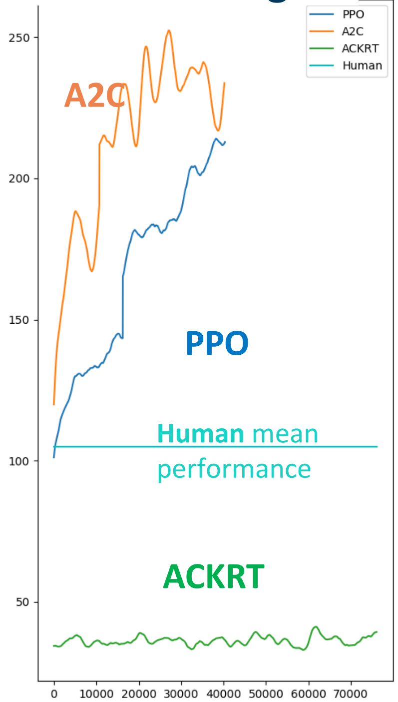


Ref.:

Results

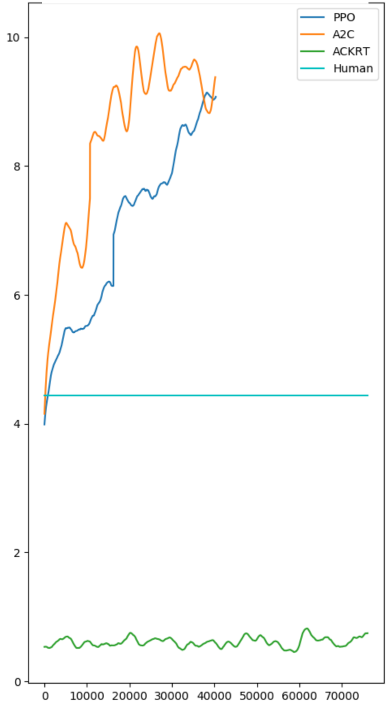
- **PPO**
  - Proximal Policy Optimization
- **A2C**
  - Advantage Actor Critic
- **ACKRT**
  - Actor Critic using Kronecker-factored Trust Region
- Mean **Human**
- Early Super-human performance
- Volatile vs Monotonic

Game Length



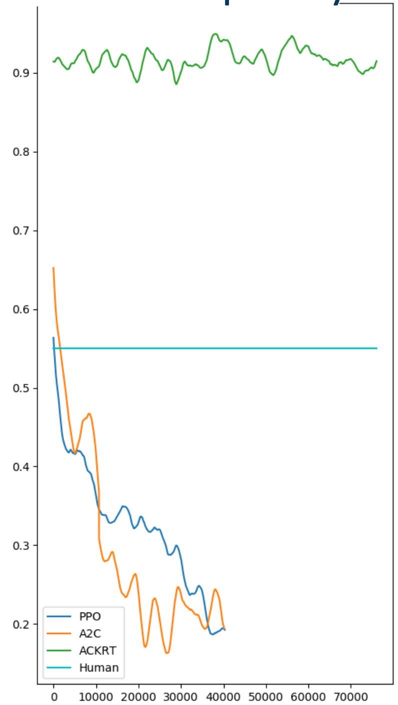
#Training Games

Game Scores



#Training Games

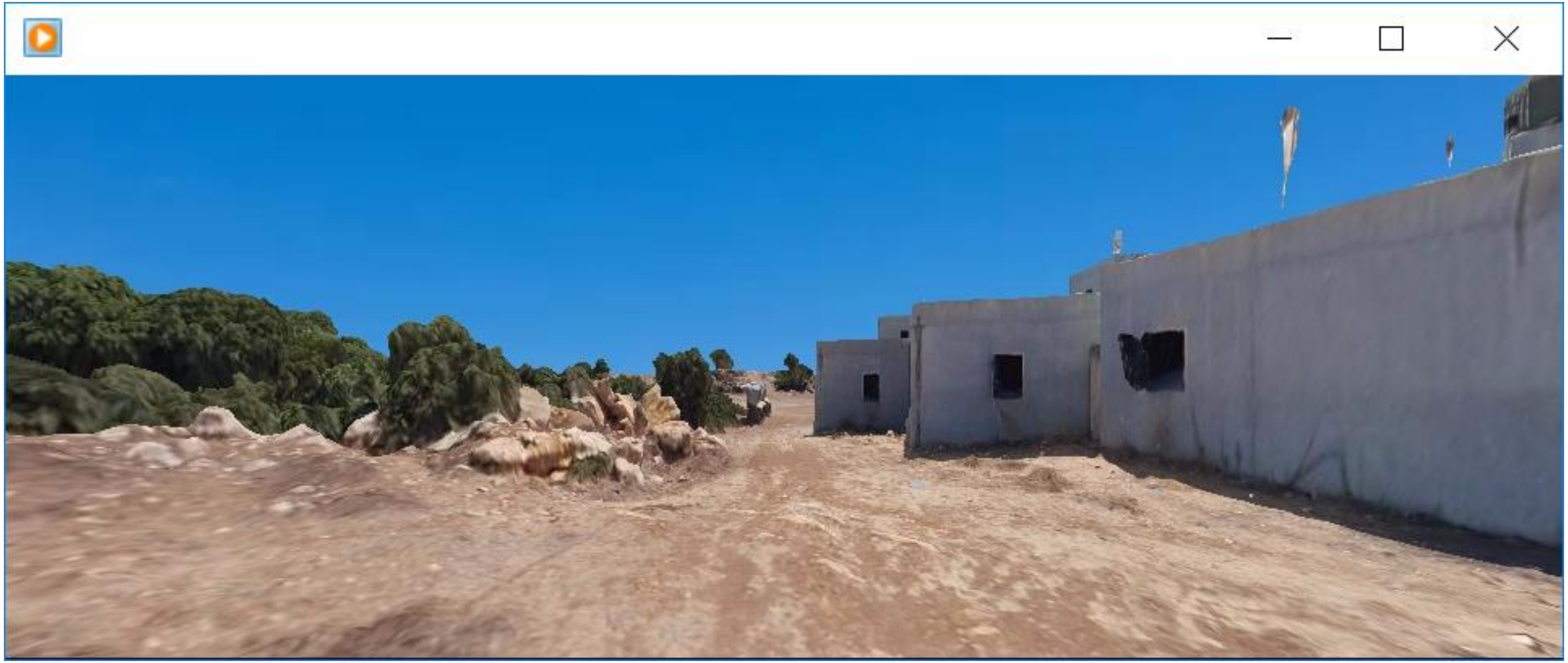
Crash frequency



#Training Games

Ref.:

Test drive



Limitations

- Moving obstacles
- Traffic rules
- Steering to movement direction
- Blocking obstacles
- No U turns

Future work

- Treating limitations
- GANs for even more realistic images
  - To look like current reality
  - Add rain, mud, darkness, fog, and dust
- Control Learning
  - Copy-Paste the vehicle behaviour
- Driving a real platform
  - Test in the modelled area

Future work

- Treating limitations
- GANs for even more realistic images
  - To look like current reality
  - Add rain, mud, darkness, fog, and dust
- Control Learning
  - Copy-Paste the vehicle behaviour
- Driving a real platform
  - Test in the modelled area

Future work

- Treating limitations
- GANs for even more realistic images
  - To look like current reality
  - Add rain, mud, darkness, fog, and dust
- Control Learning
  - Copy-Paste the vehicle behaviour
- Driving a real platform
  - Test in the modelled area

RAFAEL

THANK YOU

www.rafael.co.il