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

Dr. Refael Vivanti
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.
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.
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.
The challenge

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

<table>
<thead>
<tr>
<th>Navigation – Path planning</th>
<th>Obstacle avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>strategical task</td>
<td>tactical task</td>
</tr>
<tr>
<td>sparse rewards</td>
<td>dense rewards</td>
</tr>
<tr>
<td>geometric input</td>
<td>visual input</td>
</tr>
</tbody>
</table>

• Doing both together is hard
Our Solution: split

Navigation - Before mission

Current location → Target location → Navigation system → Track

3D Model
Our Solution: split

Navigation - Before mission

Current location

Target location

Navigation system

Track

3D Model

Avoidance - During mission

Rendered Image

RL Agent

Driving Decision

New pose

Rendering

Before mission

Our Solution: split

Current location

Target location

Navigation system

Track

3D Model

Avoidance - During mission

Rendered Image

RL Agent

Driving Decision

New pose

Rendering

Before mission
Navigation – Before mission

1. 3D model geometry
Navigation – Before mission

1. 3D model geometry

2. Segmentation by passability

3D model

UNCLASSIFIED

Ref.: Navigation – Before mission

11
Navigation – Before mission

1. 3D model geometry
2. Segmentation by passability
3. Watershed: trails + width

3D model

Ref.: Navigation – Before mission

3D model

1. 3D model geometry
2. Segmentation by passability
3. Watershed: trails + width

Ref.: Navigation – Before mission
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

<table>
<thead>
<tr>
<th>Lon</th>
<th>Lat</th>
</tr>
</thead>
<tbody>
<tr>
<td>32.6532</td>
<td>35.6686</td>
</tr>
<tr>
<td>32.6638</td>
<td>35.2684</td>
</tr>
<tr>
<td>32.3530</td>
<td>35.4683</td>
</tr>
<tr>
<td>32.8532</td>
<td>35.2784</td>
</tr>
<tr>
<td>32.6535</td>
<td>35.2694</td>
</tr>
<tr>
<td>32.6236</td>
<td>35.2680</td>
</tr>
<tr>
<td>32.2538</td>
<td>35.2683</td>
</tr>
</tbody>
</table>
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.
• Compared 3 SOTA Actor-Critic based RL algorithms:
  - **PPO** Proximal Policy Optimization
  - **A2C** Advantage Actor Critic
  - **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

[Diagram showing network architecture with convolutional layers and fully connected layers.]
Training

• Compared 3 SOTA Actor-Critic based RL algorithms:
  - **PPO** Proximal Policy Optimization
  - **A2C** Advantage Actor Critic
  - **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
• **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**

- PPO
- A2C
- ACKRT
- Human mean performance

**Game Scores**

- PPO
- A2C
- ACKRT
- Human

**Crash frequency**

- PPO
- A2C
- ACKRT
- Human

---

Proprietary of Rafael – Advanced Defense Systems Ltd
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**
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**
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
THANK YOU

www.rafael.co.il

Proprietary of Rafael – Advanced Defense Systems Ltd