LEARNING IN VIRTUAL ENVIRONMENTS
LEARNING: STATIC VS DYNAMIC LEARNING

• Learning: using experience from the environment to improve performance
• Static: learner cannot interact with the environment
• Dynamic: learner can interact with the environment
• State-of-the-art AI models are often trained with static datasets
• Human learning relies on interaction with environment
  • Dynamic learning
3D GENERATED STATIC DATASETS

- Computer-generated static datasets have been used in literature
  - KITTI (distance estimation)
  - TORCS (vehicle recognition)
  - Half-Life (surveillance tracking)
  - GTA V (pose estimation, object detection, autonomous driving)

- Cheap and fast creation of static annotated dataset

- Usually good performances on standard benchmarks
LEARNING IN VISUAL ENVIRONMENTS

• Human-like learning requires interaction with a dynamic environment
• The agent should be able to interact with it
• Using a robot is impracticable
  • Expensive
  • Dangerous
  • Slow
  • Experiments cannot be easily reproduced
LEARNING IN \textit{viR\textsc{T}\textsc{ual} en\textsc{vironments}}

• 3D Computer-generated and simulated environment

• Must-have features
  • The agent must be able to interact with it and stimulate responses
  • Photorealistic for easy transfer to real life application
  • Customizable and extendable to add new scenarios
  • Fast rendering

• Advantages
  • Decoupling learning from physical issues
  • Simulated time can be faster than real time
  • Cheap and reliable reproduction of experiments
WHAT RESOURCES ARE AVAILABLE?

- AI2-THOR and Habitat
- Most recent frameworks
- Photorealistic
- Habitat has larger scenarios
- AI2-THOR has more features
  - Customizable
  - Physics and interactions
- Custom graphics engines
EXISTING GRAPHICS ENGINES
RAGE (ROCKSTAR ADVANCED GAME ENGINE)

- Used by games such as GTA V
- Not natively customizable
  - Community reverse engineered the engine API
- Need to manually implement image annotation
  - Relatively easy to add Bounding Boxes
  - Very difficult to add Segmentation
- Can add moving objects and simulate real-life scenarios (cars, planes, people)
- Generation is very slow, around 50,000 images per hour
RAGE DEMO
HABITAT AI

- Supports highly realistic 3D environment dataset
  - SUNCG, MatterPort3D, Gibson, Replica
  - The user can provide own 3D environment
- Excellent performances: simulation is not a bottleneck for learning
  - Simulator developed in C++
  - Up to 10,000 frames per second
- Python API to interface with common learning frameworks
  - Move agent and obtain sensor data
- Customizable sensors and agent geometry and physics
HABITAT AI ENVIRONMENT

- Agents receive information on what they see
  - RGB, Depth, GPS
- Multiple built-in AI tasks for learning by imitation
  - Navigate to point or object
  - Simple QA tasks
- Habitat Challenge
  - Separate module with benchmarks and baselines
- No built-in interaction or moving objects
HABITAT AI DEMO

Matterport3D
AI2-THOR

- *Actionable* objects: the agent can interact with them
  - E.g. microwave door can be opened, bread can be sliced
- Good degree of photorealism
- Python API to interface with scenarios and objects
  - Moving agent, applying forces and interacting with objects, obtaining sensor data
- The user can provide own scenarios
- Modeling moving objects seems quite hard
AI2-THOR ENVIRONMENT

- Multiple Agents in the same environment
- Agents can move on a grid (configurable size)
- Agents receive information on what they see
  - RGB and Depth info
  - Segmentation and bounding boxes info
  - Object visibility and interactability
- Agents can interact with objects
  - Push, pull, grab, release, throw, slice, empty, fill, etc.
- Object free movement is very limited
AI2-THOR DEMO
CUSTOM GRAPHICS
ENGINES
<table>
<thead>
<tr>
<th>PROS</th>
<th>CONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Completely customizable</td>
<td>• No built-in features</td>
</tr>
<tr>
<td>• Easy to implement segmentation</td>
<td>• Behavior of objects must be manually</td>
</tr>
<tr>
<td>• Moving objects can be added quite easily</td>
<td>scripted</td>
</tr>
<tr>
<td></td>
<td>• Trajectory is determined by the script</td>
</tr>
</tbody>
</table>
UNITY ENGINE DEMO
CONCLUSIONS

• Static datasets do not allow human-like learning
• Literature used 3D computer-generated datasets: still static
• The virtual environment approach looks promising
  • There are a few interesting and useful resources
  • Although not fully mature and with a little community
• Existing work have shown the usefulness of the approach
• Extensions are needed to model more scenarios and moving object
• Custom engines offer more flexibility but it is harder to create realistic scenarios