Chair of Media Technology
Department of Electrical and Computer Engineering
Technical University of Munich

The Realistic Robotic Simulator

Edwin Babaians

Chair of Media Technology

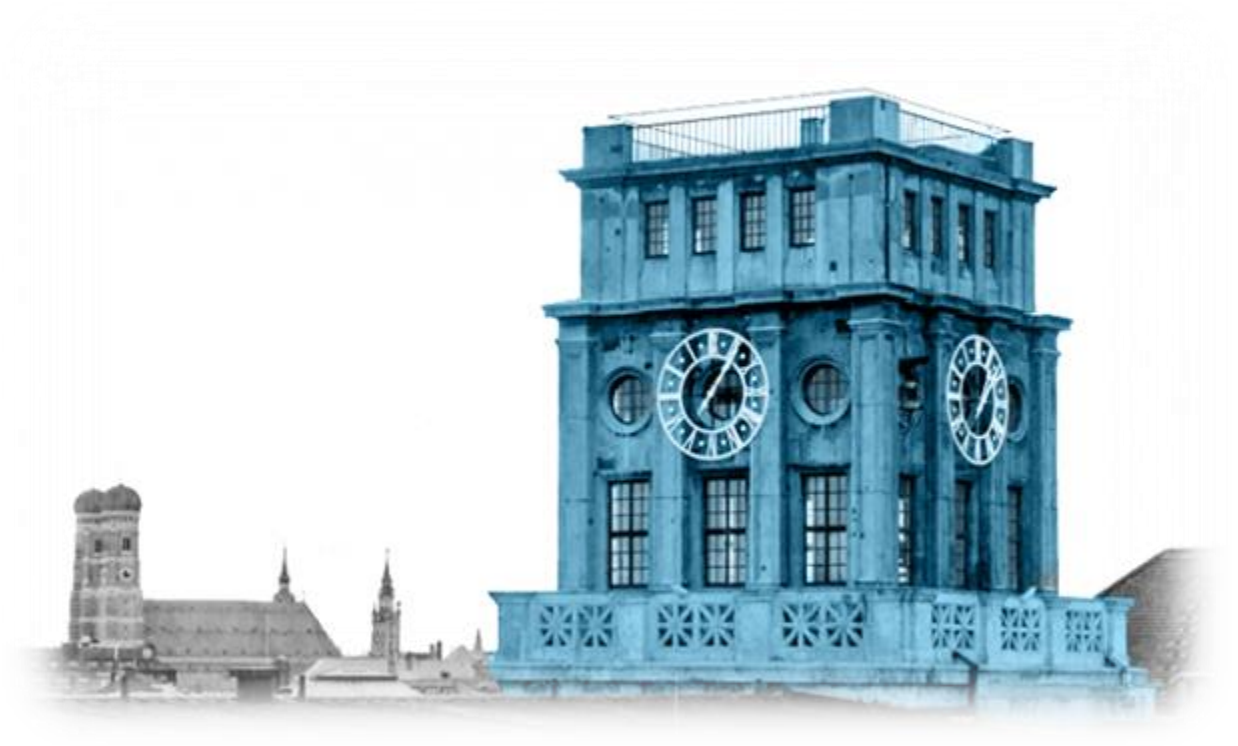
Department of Electrical and Computer Engineering

Technical University of Munich, Munich, Germany

Supervisor: **Prof. Dr.-Ing. Eckehard Steinbach**

Mentor: **Prof. Dr.-Ing. Wolfgang Kellerer**

04.Feb.2021



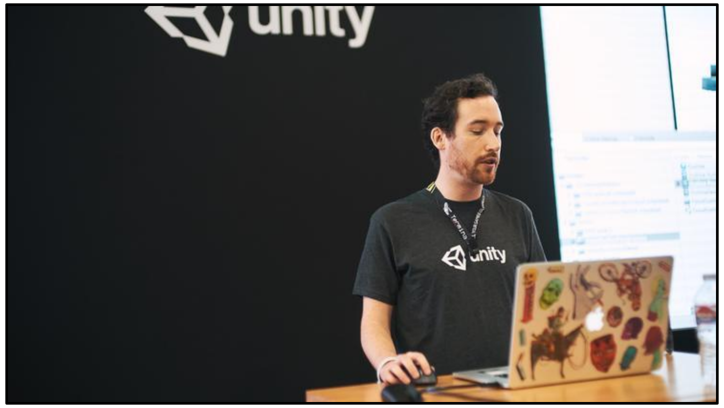
Outline

- Motivation
- Related Works
- System Overview
- Experiments
- Demo
- Conclusion and Future Work

“Motivation”

Why Unity3D?

Virtual Reality

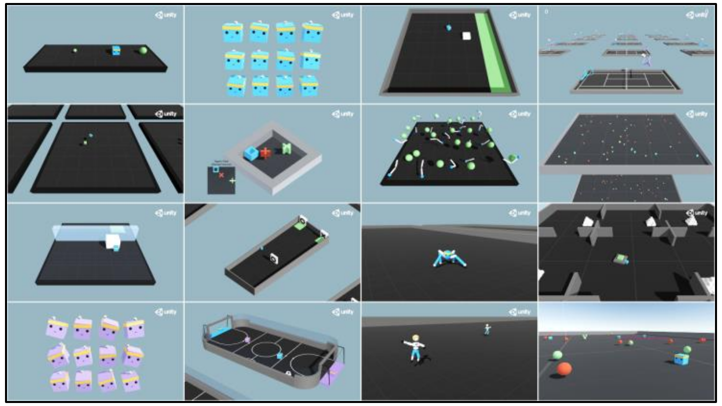


Community

Cross-platform

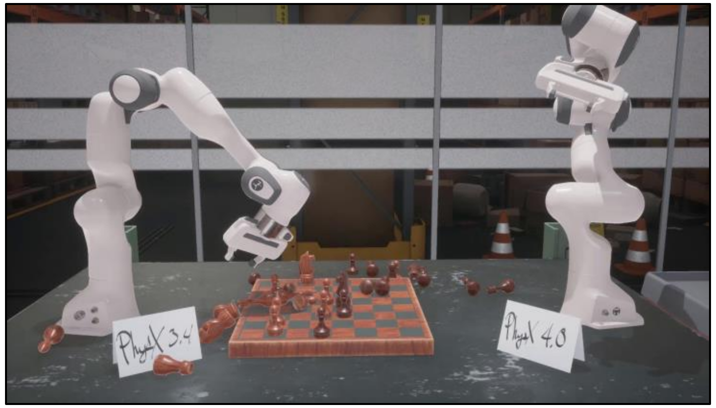
Advanced Graphics

Unity3D for Robotics?



OpenAI Gym Interface [1]

Nvidia Flex Interface

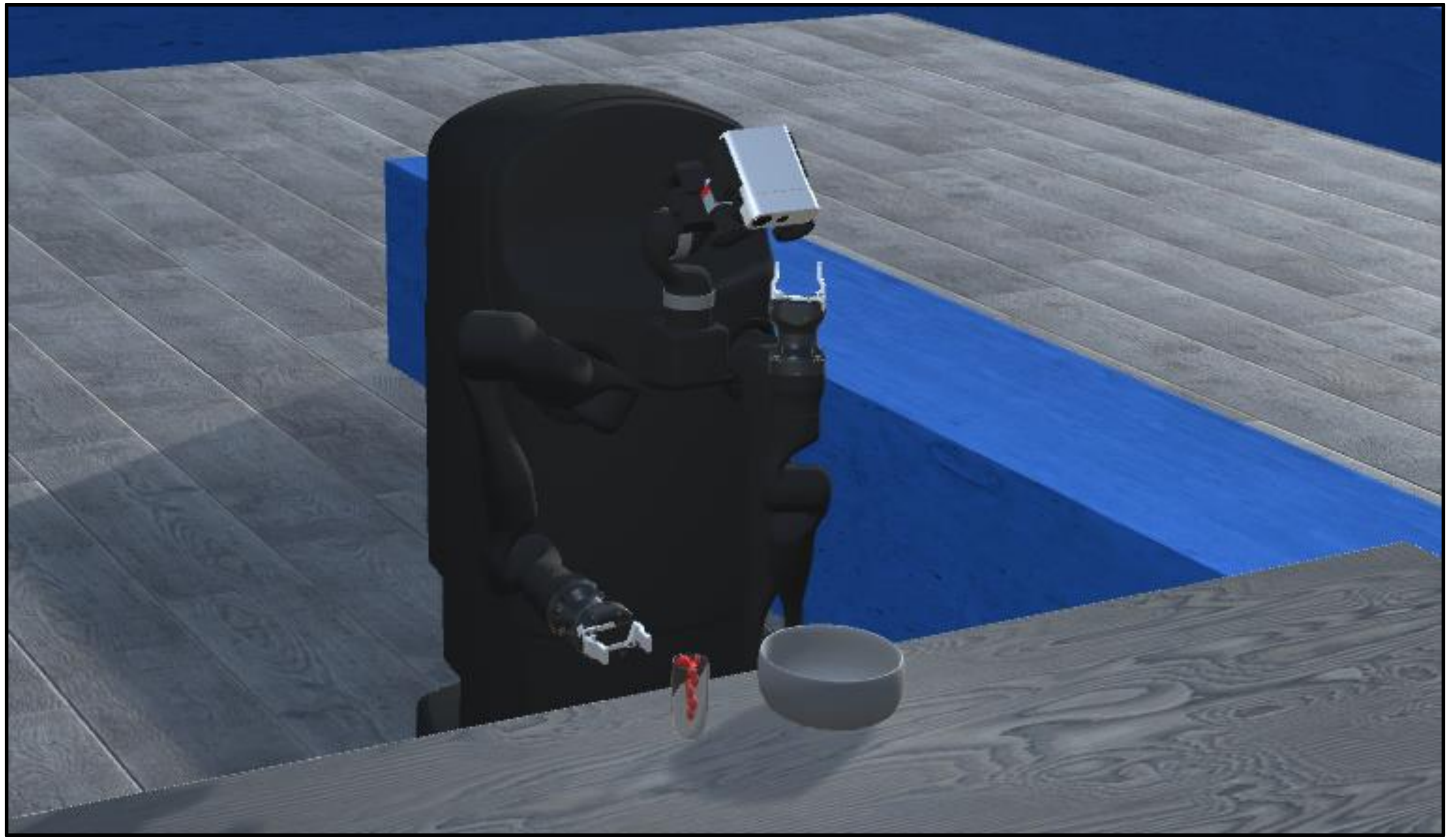


Nvidia Physics 4.1

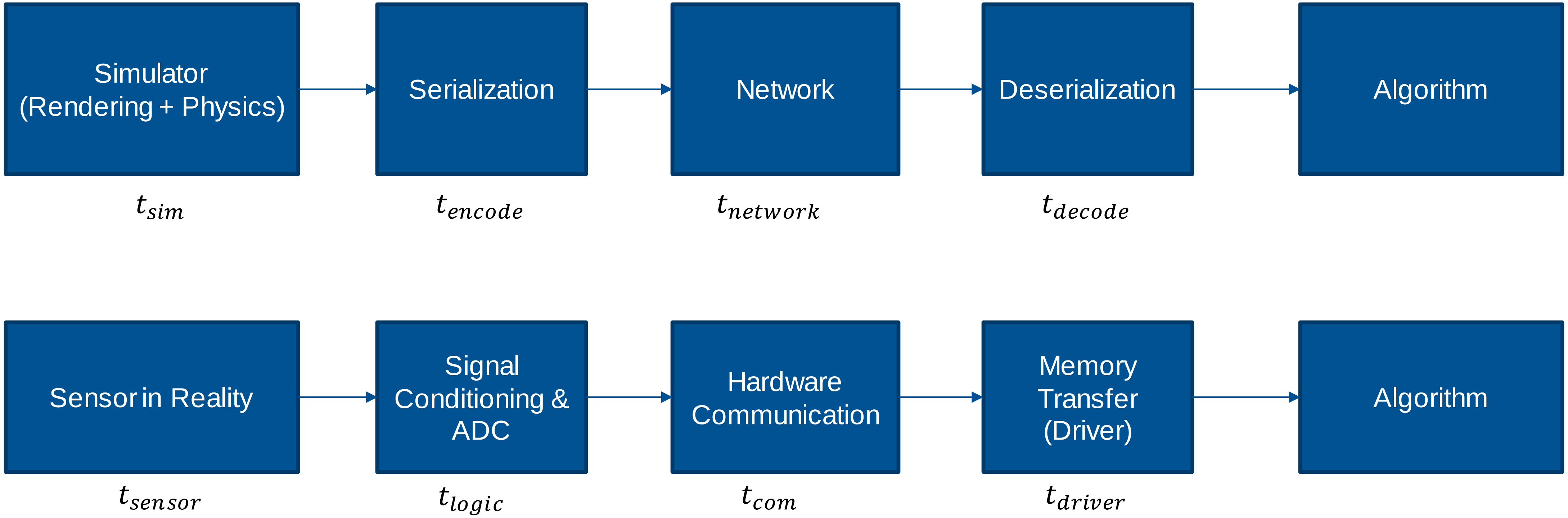
Intuitive GUI designer

Unity3D for Haptic-enabled Robotics?

"Imagine we want to teach robot how to pour liquids using simulation"



Simulation-to-Algorithm vs Reality-to-Algorithm



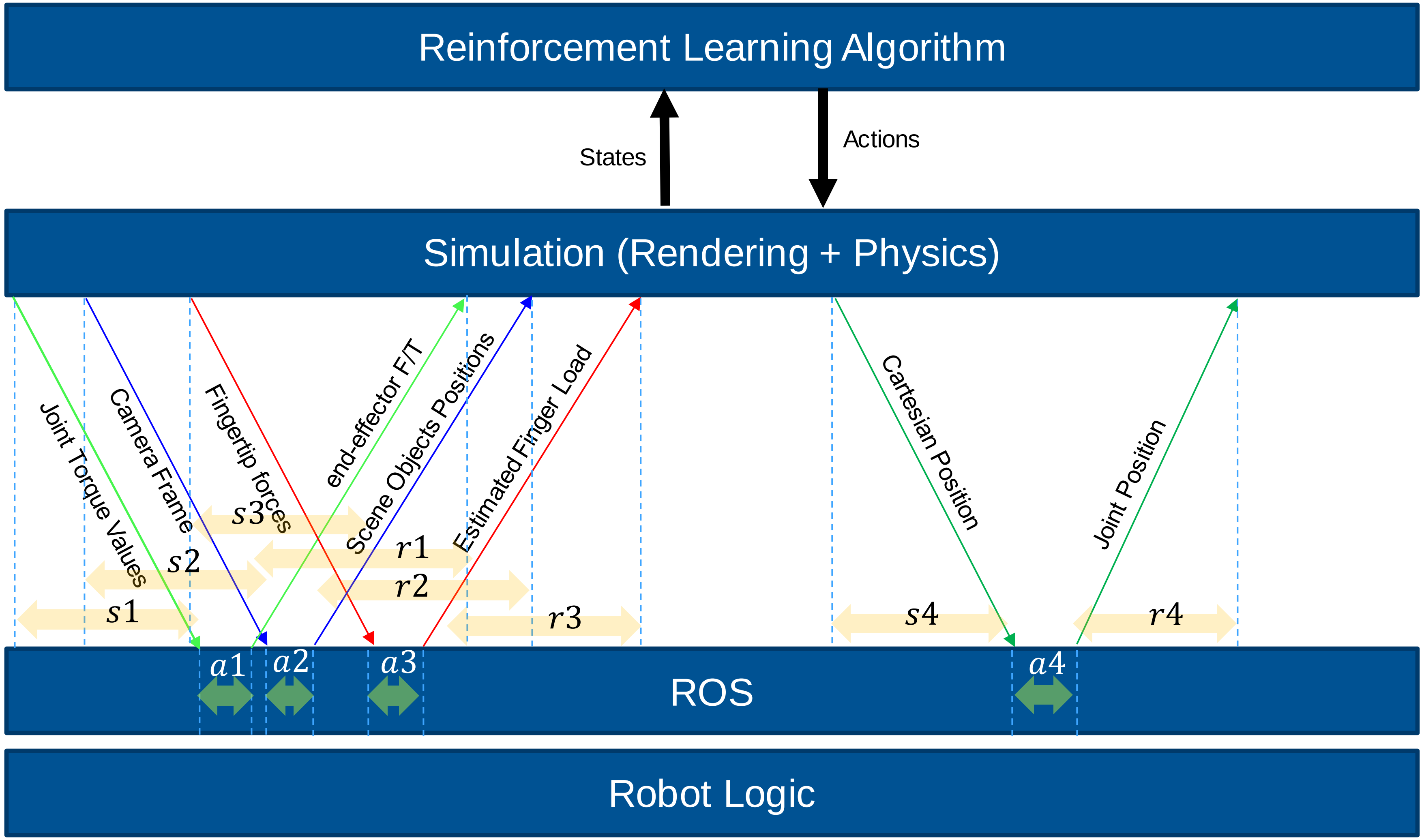
$$t_{sim} + t_{encode} + t_{network} + t_{decode} = t_{sim-to-algorithm}$$

$$t_{sensor} + t_{logic} + t_{com} + t_{driver} = t_{reality-to-algorithm}$$

$$\boxed{t_{sim-to-algorithm} \leq t_{reality-to-algorithm}}$$

$$\boxed{t_{encode} + t_{network} + t_{decode} = t_{bridge}}$$

Where is the Bottleneck?



Faster than Realtime?

Robot logic is not cabale of being faster than real-time

Huge frame rate drop

More Robot Instances ?

This means more system resources and network bandwidth

Is limited due to the system resources

Then?

We can speedup the loop !

t_{bridge} $t_{algorithm}$

$t_{bridge_i} = t_{si} + t_{ri}$

$t_{bridge_i} \leq t_{algorithm_i}$

Reinforcement Learning Algorithm

...

Reinforcement Learning Algorithm $Iteration\ x_i$

States

Actions

$$t_{states} = \sum_{k=1}^{src}(s_k + r_k) + \sum_{k=1}^{src} a_k + Sim_{render} + Sim_{physics} \approx t_{bridge} + t_{simulator}$$

$src = state\_request\_count$

$$t_{action} = \sum_{k=1}^{arc}(s_k + r_k) + \sum_{k=1}^{arc} a_k + Sim_{render} + Sim_{physics} \approx t_{bridge} + t_{simulator}$$

$arc = action\_request\_count$

ROS

Robot Logic

“Related Works”

ROS#

ROS2

LMT

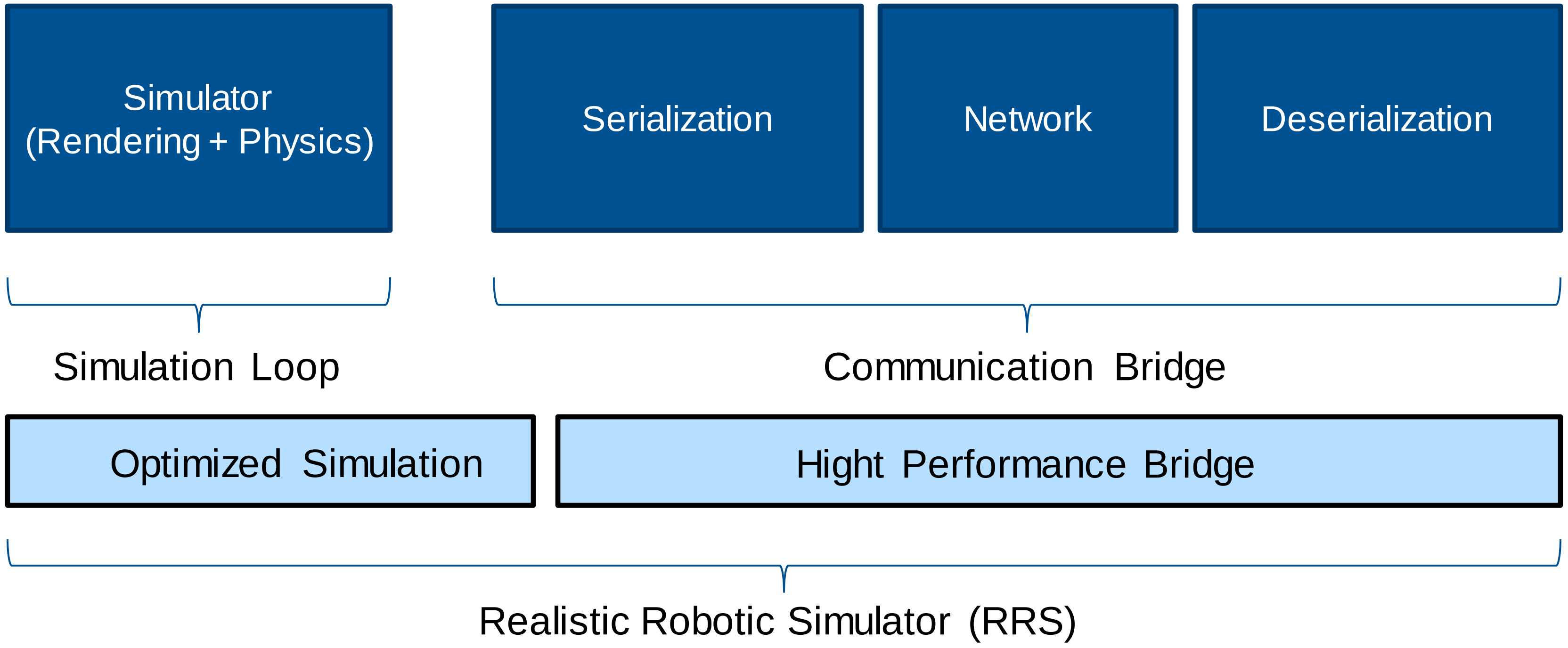
What are the state-of-the-art Bridges?

Name	ROS# [1]	ROS2 [2]	LMT[3]
Affiliation	Simens AG	Dyno Robotics	TUM
Published Year	2016	2019	2021
Network System	WebSocket's + ROS Bridge Server	DDS	ZMQ TCP/IP and UDP
Serialization System	JSON	DDS + (ROS2-ROS1 Bridge)	Google Proto Buffers
Time Synchronization	No	TNS	NTP,PTP
Simulated Sensors	Camera	LiDAR	Camera, Depth Camera, LiDAR, IMU
Simulated Actuators	Based on Rigid Bodies	Based on Rigid Bodies	Based on Articulation Bodies
Tested Platforms (Only Bridge)	Windows, Linux	Windows	Windows, Linux, Android
Included Robot	TurtleBot	TurtleBot	Kinova Movo
Multi Instance Robot	No	No	Yes

[1] source: https://github.com/siemens/ros-sharp
[2] source: https://github.com/DynoRobotics/UnityRos2
[3] source: *Babaians, Edwin, Mohsen Tamiz, Yaser Sarfi, Amir Mogoei, and Esmaeil Mehrabi. "ROS2Unity3D; High-Performance Plugin to Interface ROS with Unity3d engine." In 2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium, pp. 59-64. IEEE, (2018).*

“System Overview”

How we can tackle this problem?

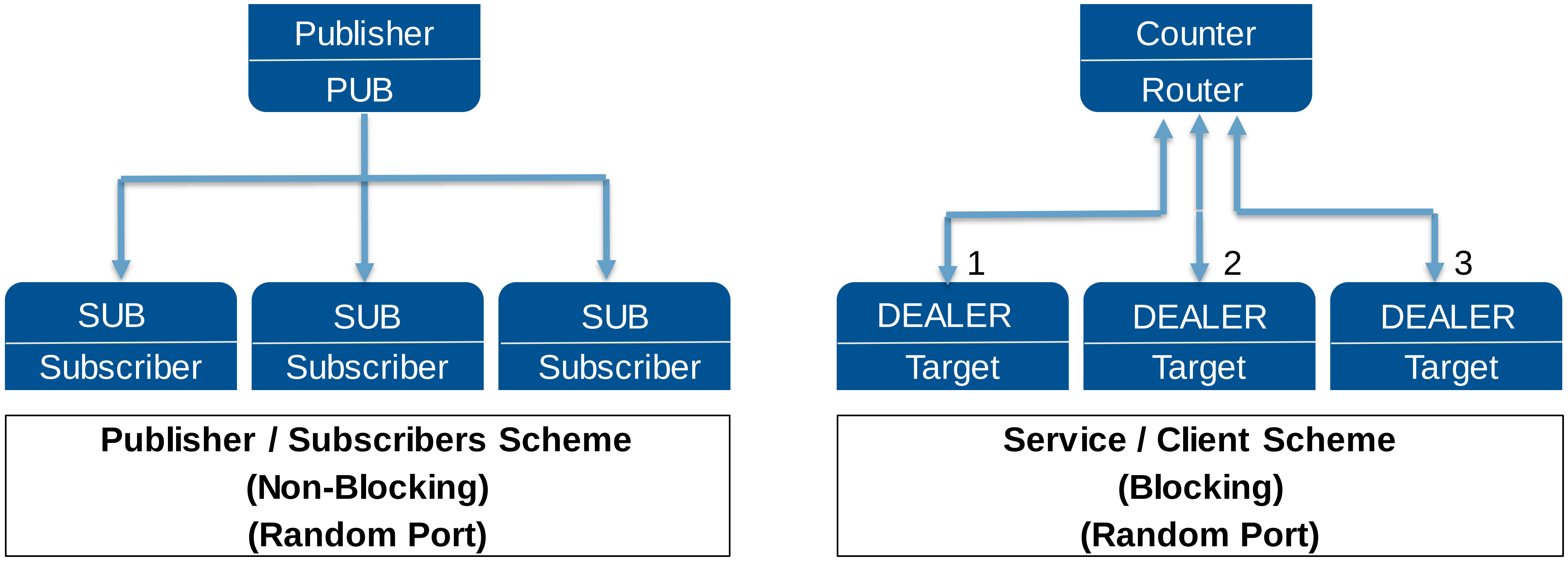


“System Architecture”

LMT Bridge

ZMQ	Proto buffers
Service Discovery	

ZMQ Network Architecture



Google Proto buffers

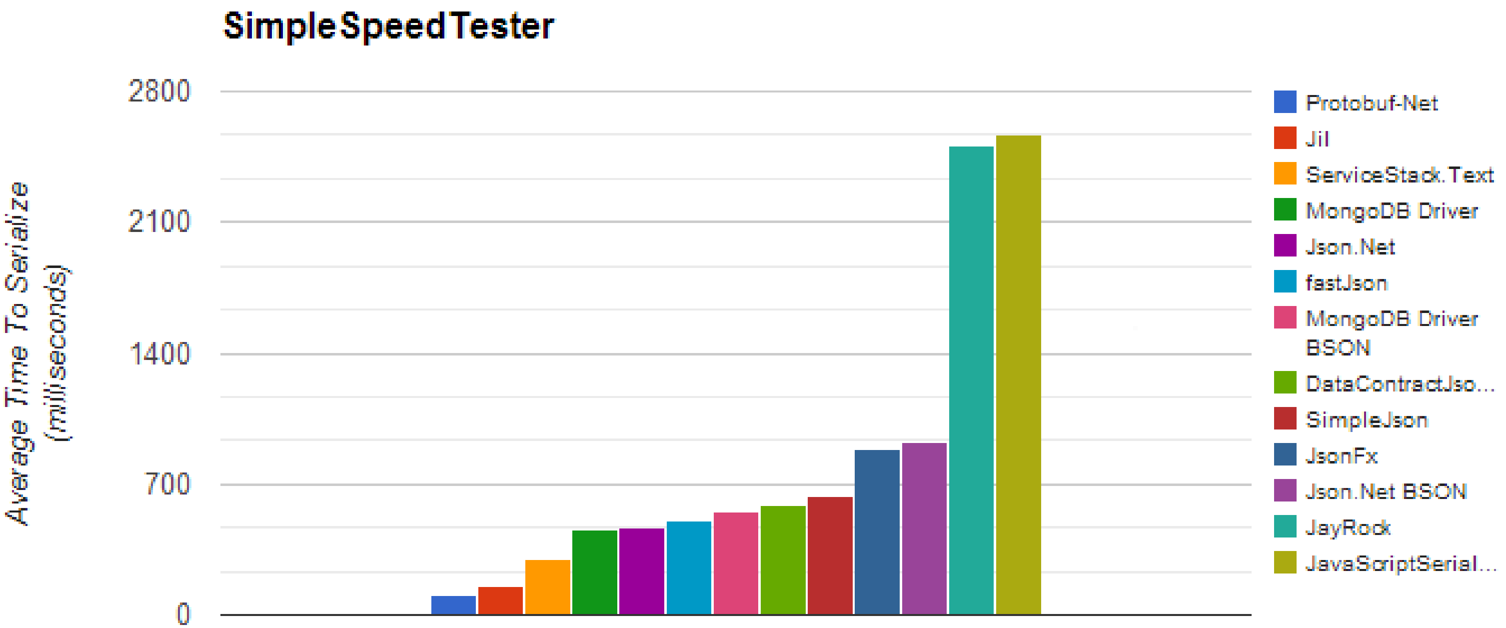
```
//polyline.proto
syntax = "proto2";

message Point {
  required int32 x = 1;
  required int32 y = 2;
  optional string label = 3;
}

message Line {
  required Point start = 1;
  required Point end = 2;
  optional string label = 3;
}

message Polyline {
  repeated Point point = 1;
  optional string label = 2;
}
```

Format Example



Time Synchronization / Service Discovery

Current System:

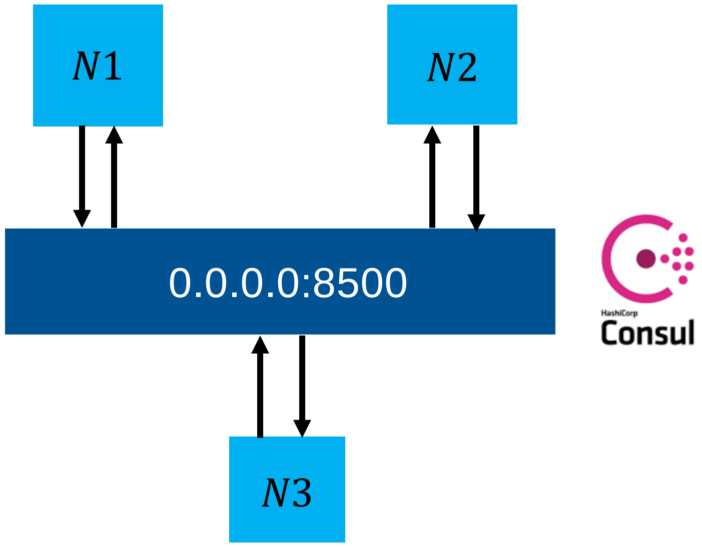
- Network Time Protocol (NTP)
- Port:123

Next Version:

- Precision Time Protocol (PTP)
- Deterministic Systems, needs higher accuracy in time synchronization (1 us)
- IEEE 1588 (Time-sensitive networking)

Service Version:

- REST Api
- Topic Names
- Last heart beats (Health Checking)
- Port



“System Architecture”

RRS Components

CAMERA	LiDAR
Articulation Body	

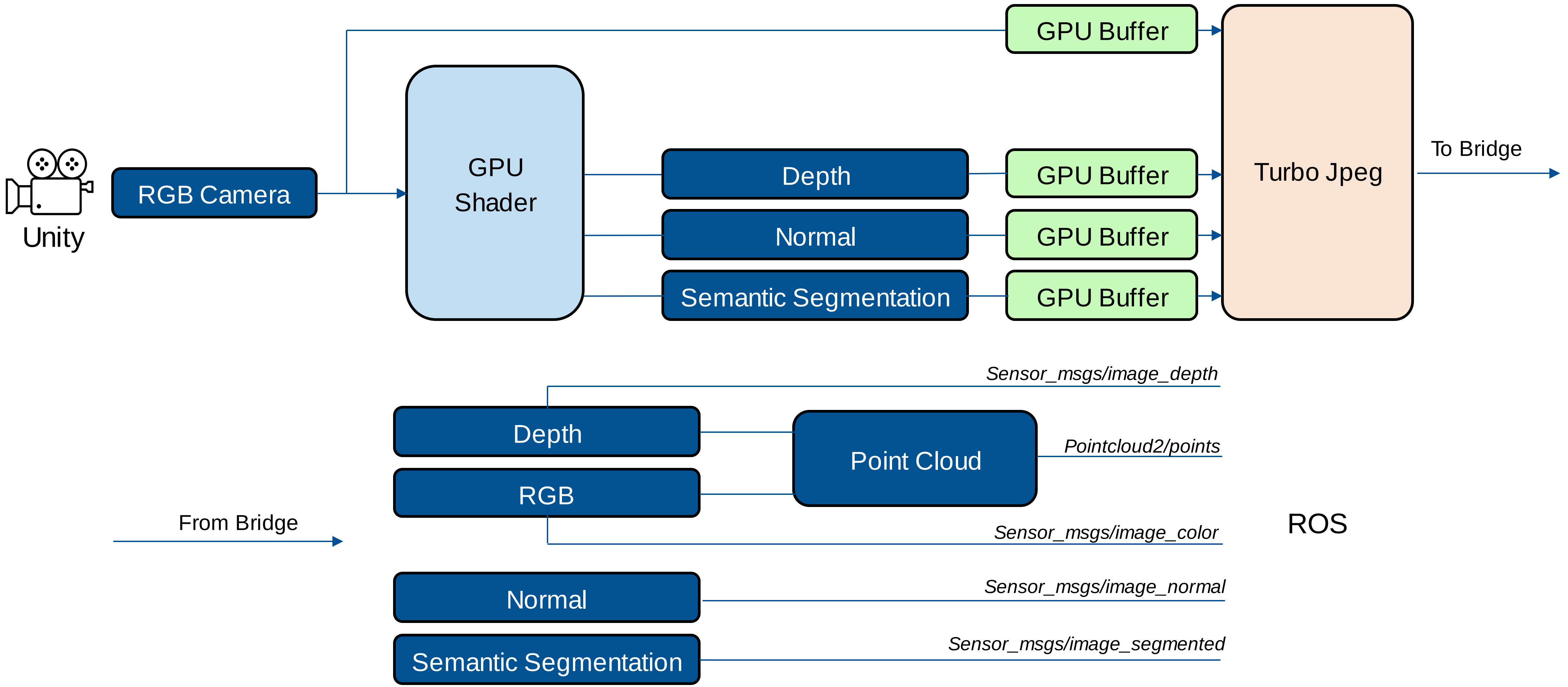
LMT's Kinova Movo Platform (Final Version)

Unity HDRP Default Scene

- 1x Stereo Camera
- 1x Color Camera
- 1x Depth Camera
- 17x Rotary Actuators
- 1x Linear Actuators
- 2x Parallel Grippers
- 2x Wrist F/T Sensors
- 2x Wrist Cameras
- 1x Holonomic Locomotion
- 1x IMU
- 2x 2D LiDAR

<u>31 Components !</u>

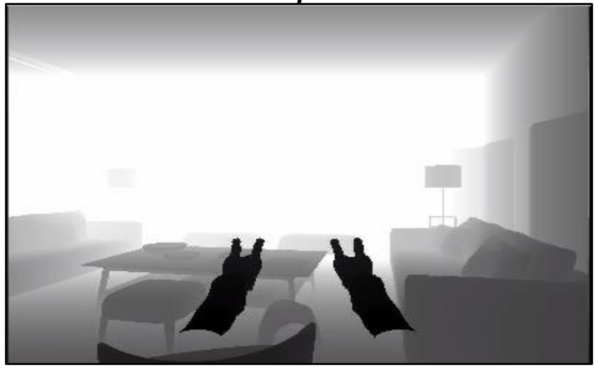
Camera Sensor



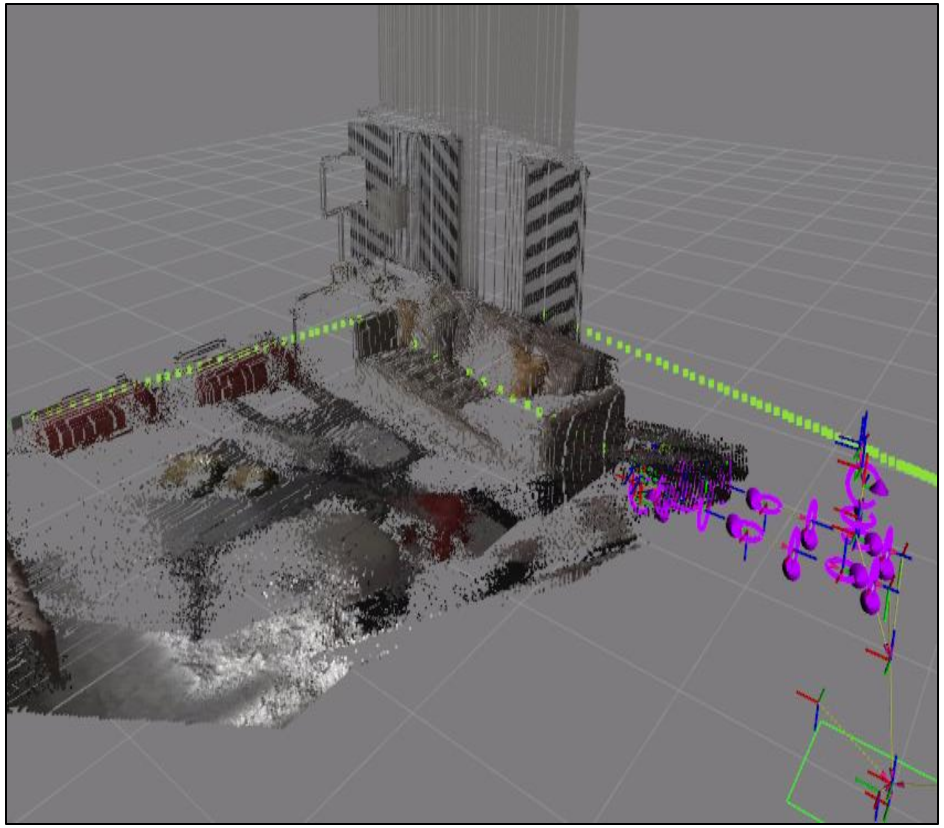
Camera Sensor

RGB

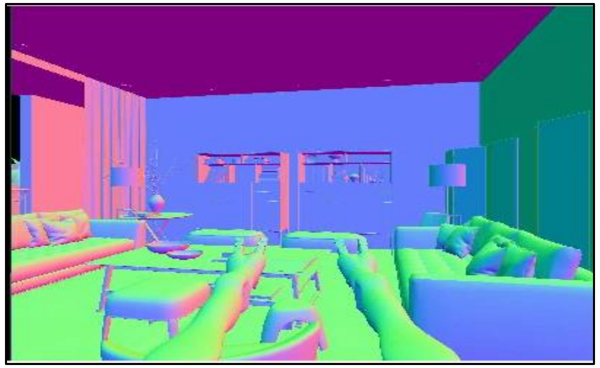
Depth



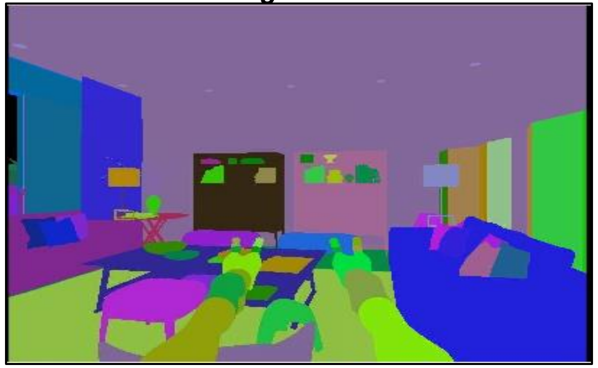
Point Cloud



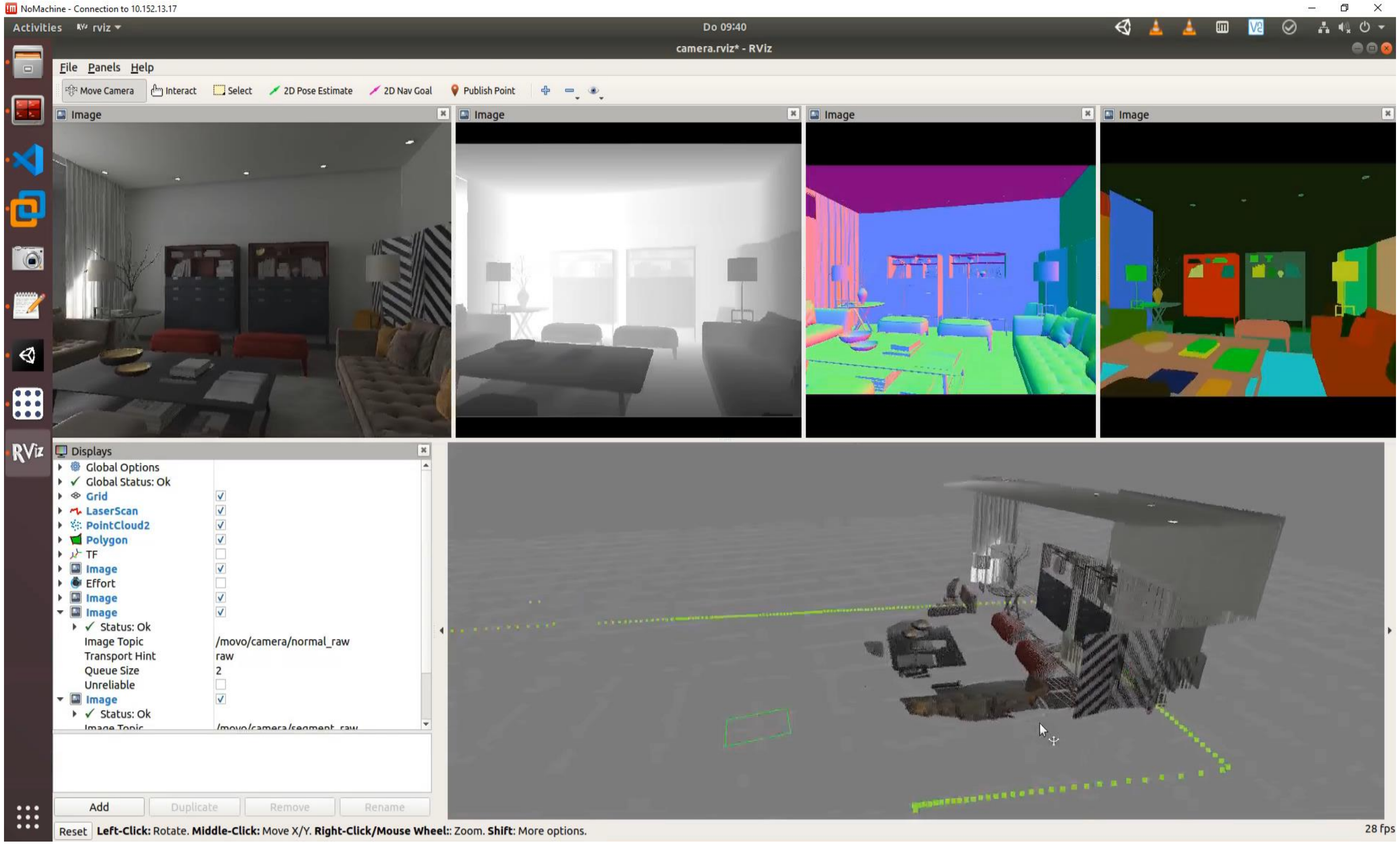
Normal



Segmented

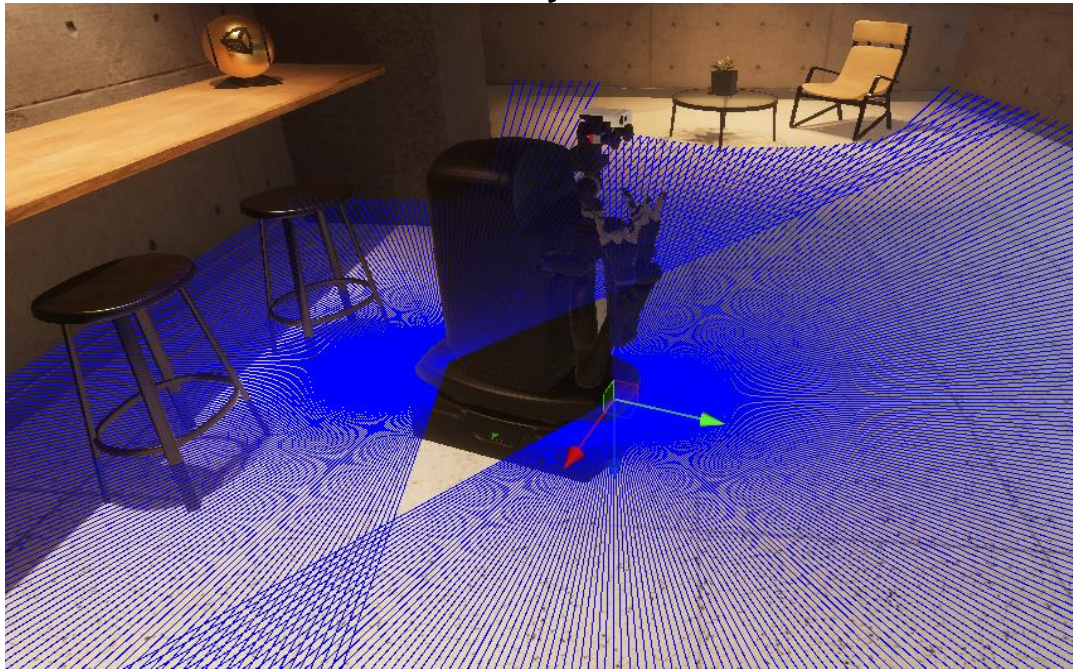


Live View

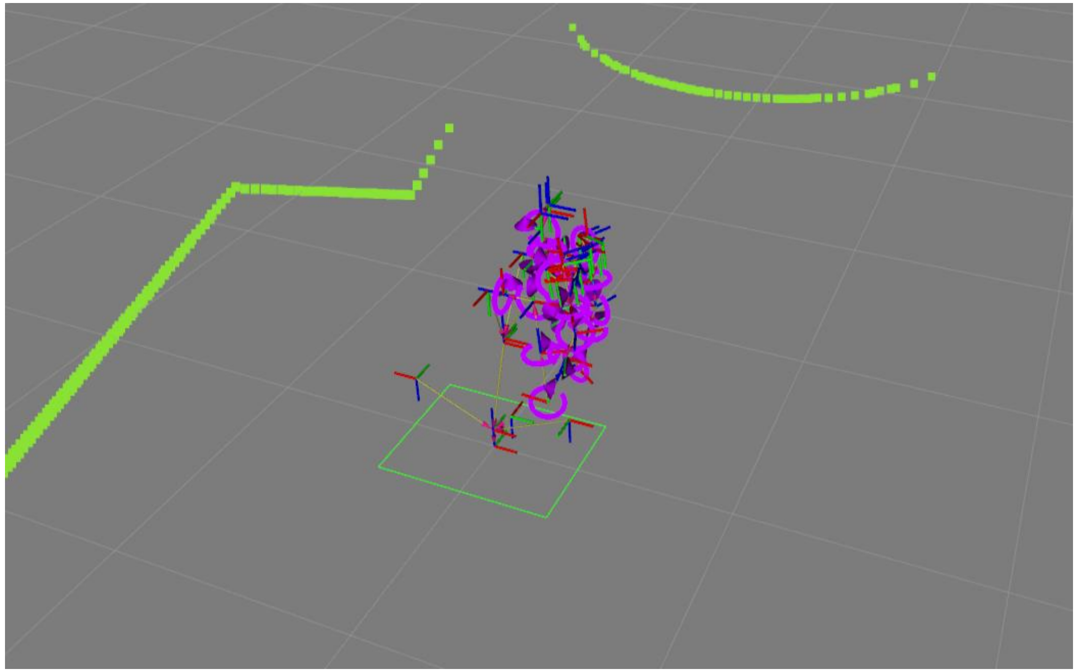

NoMachine - Connection to 10.152.13.17
Activities
rviz
Do 09:40
camera.rviz* - RViz
File Panels Help
Move Camera
Interact
Select
2D Pose Estimate
2D Nav Goal
Publish Point
Image
Image
Image
Image
Displays
Global Options
Global Status: Ok
Grid
LaserScan
PointCloud2
Polygon
TF
Image
Effort
Image
Image
Status: Ok
Image Topic /movo/camera/normal_raw
Transport Hint raw
Queue Size 2
Unreliable
Image
Status: Ok
Add
Duplicate
Remove
Rename
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
28 fps

2D LiDAR

Unity



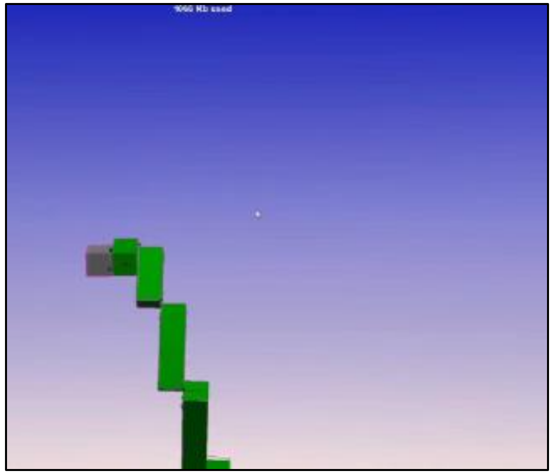
ROS



Articulation Body

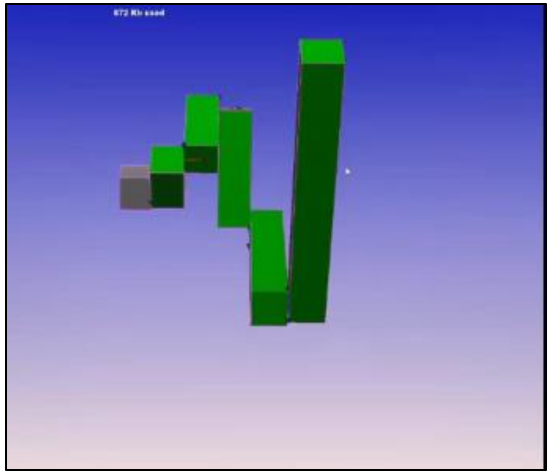
Projected Gauss-Seidel (PGS)

- Faster
- Friction is not constraints every iteration
- Not good to simulate robotic joints

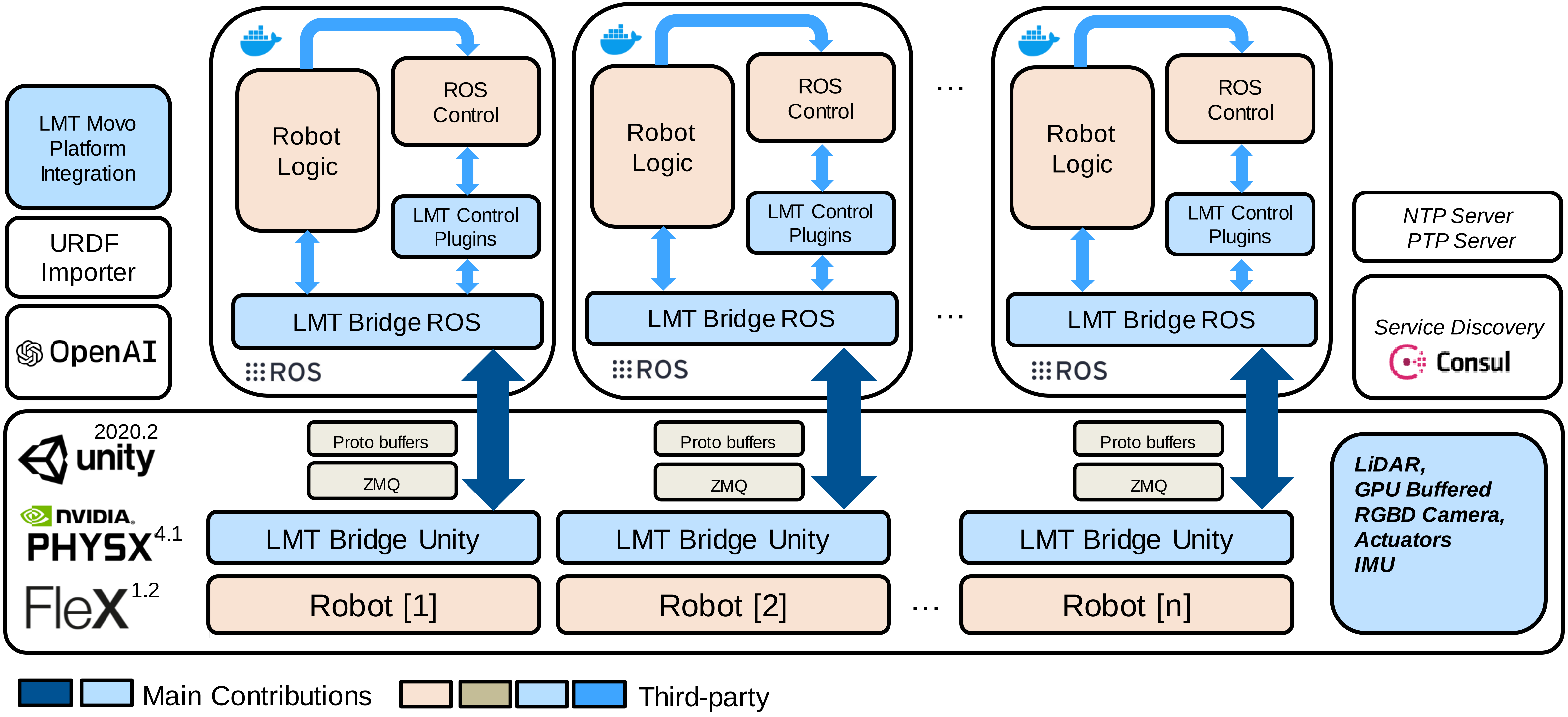


Temporal Gauss-Seidel (TGS)

- Slower
- Friction constraints every iteration
- Improved handling of high-mass ratios
- Improved joint drive accuracy
- Non-linear rigid body solver
- Inverse Dynamics



RRS System Architecture



“Experiments”

Single Instance

Gazebo and reality

Single Instance

Latency for average 10 messages

	IMU Latency (ms)	Lidar Latency (ms)	RGB Frame Latency (ms) Jpeg (Quality 85%) 800x600	Depth Frame Latency (ms) Jpeg (Quality 100%) 800x600	Point Cloud Latency (ms)
LMT	**0.88**	**1.22**	**8.42**	**9.16**	**15.68**
ROS#	2.57	4.57	12.68	13.56	-
ROS2	4.12	5.38	78.00	89.78	-

Message Size for average 10 messages

	IMU Size (byte)	Lidar Size (byte)	RGB Frame Size (byte) Jpeg (Quality 85%) 800x600	Depth Frame Size (byte) Jpeg (Quality 100%) 800x600	Point Cloud Size (byte)
LMT	**146**	**1835**	**16897**	**35985**	**0** (Is not transmitted)
ROS2	360	3547	19883	76036	-
ROS#	397	3769	23035	89841	-

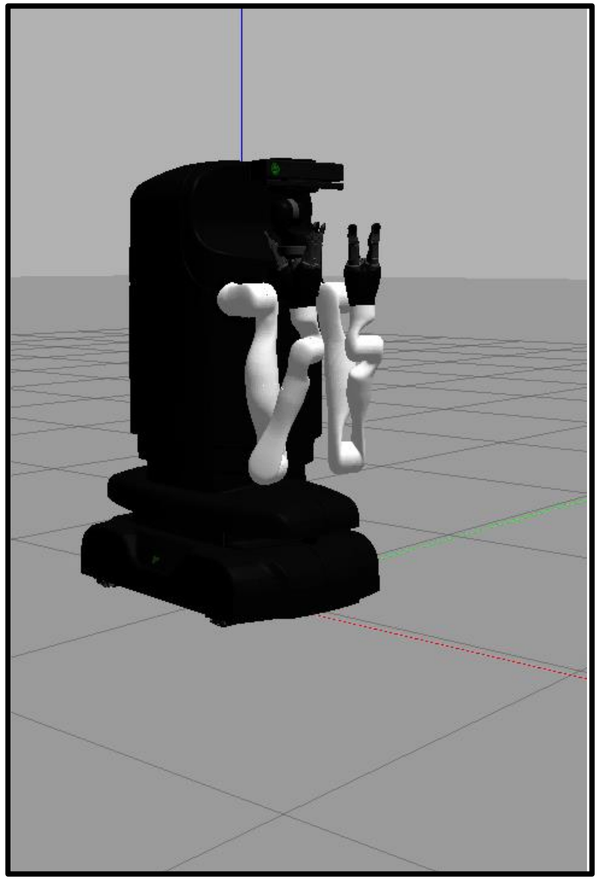
Sim-to-Algorithm Latency

ROS# Bridge	LMT Bridge	ROS2 Bridge

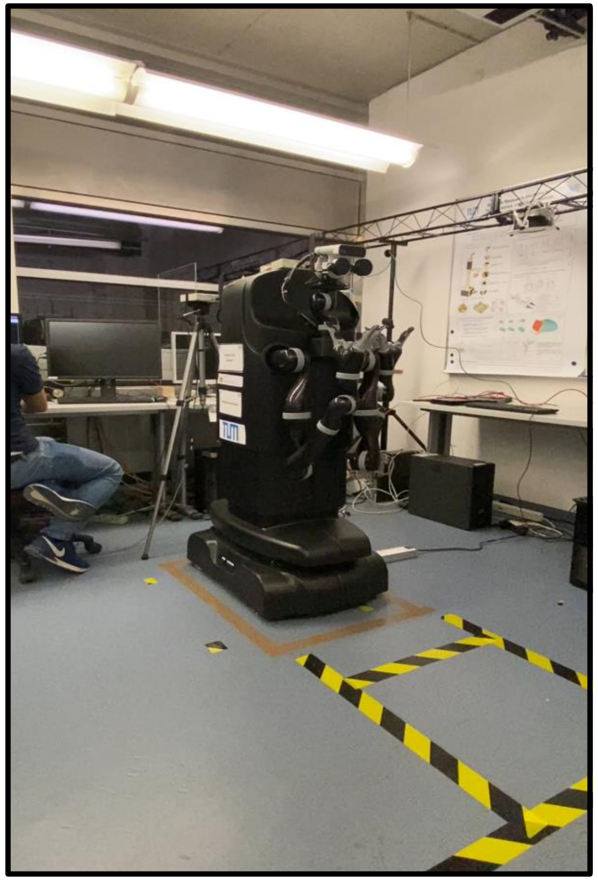
ROS# Bridge	LMT Bridge	ROS2 Bridge
Unity	Unity	Unity
JSON	Proto E	DDS
WebSocket Sharp	ZMQ	ROS2
ROS Bridge Server	RRS Bridge	ROS2 to ROS1 Bridge
ROS	ROS	ROS

RRS vs Gazebo vs Reality

RRS



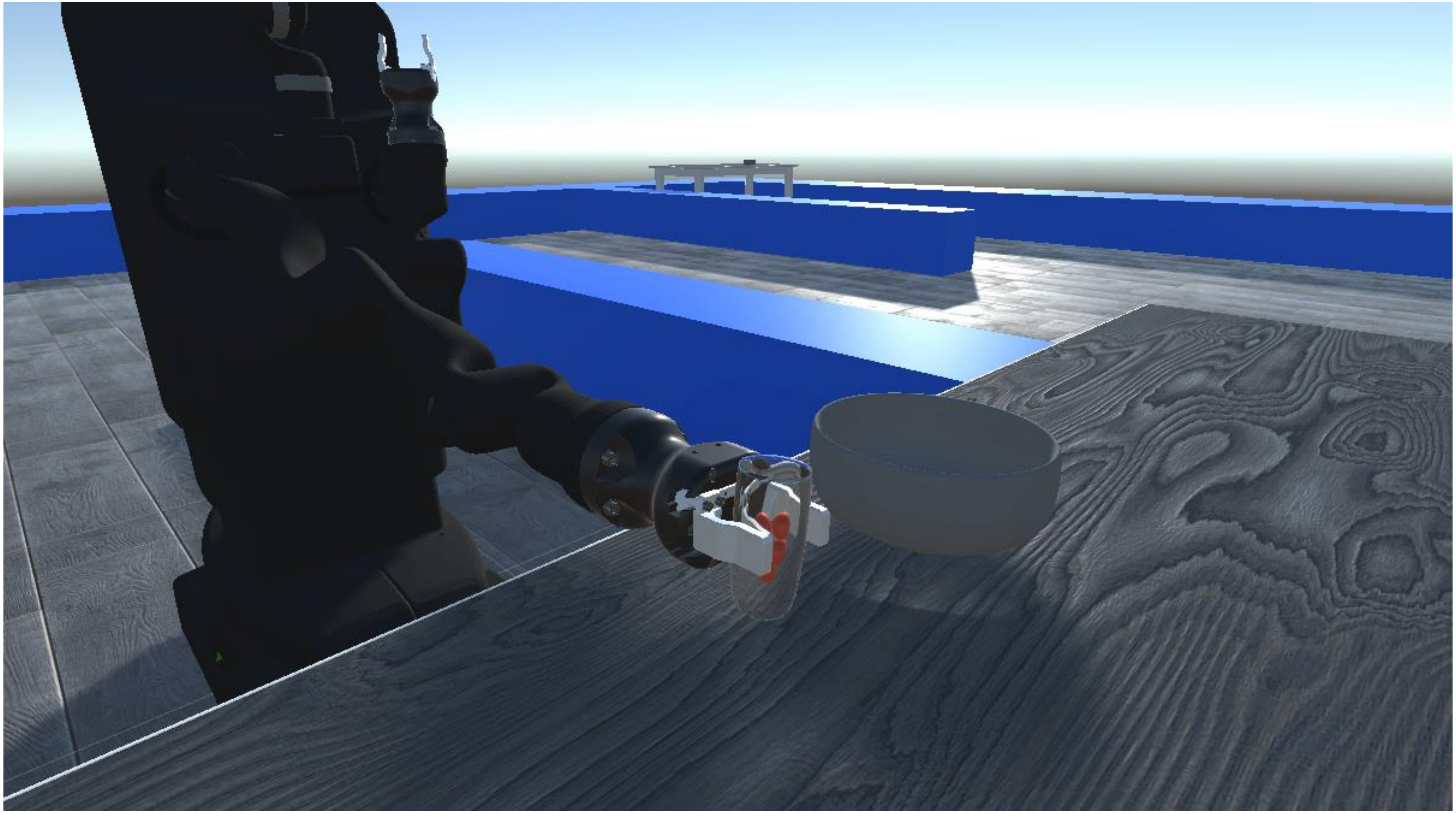
Gazebo

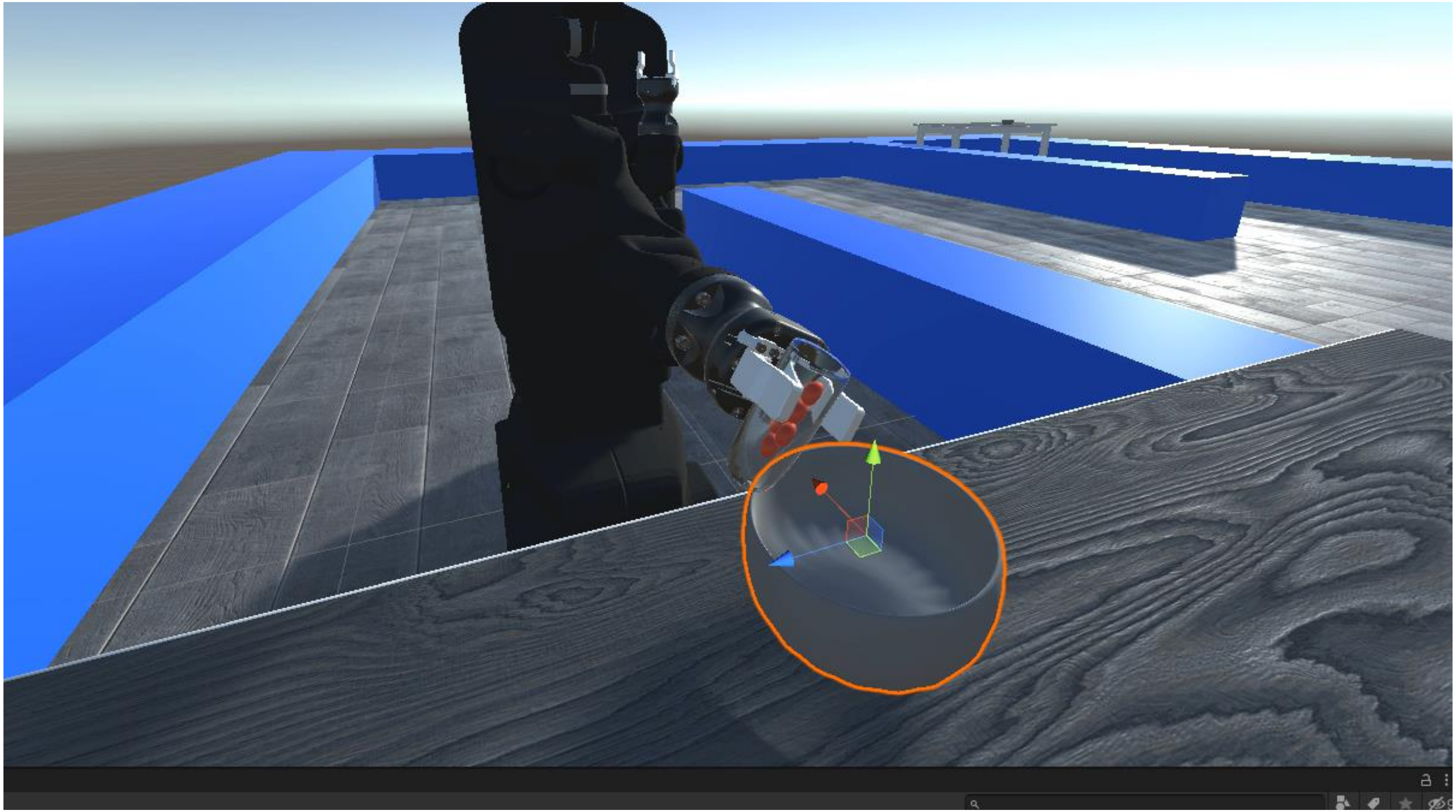


Reality

“RRS Demo”

Physics Interaction and Manipulation





“Conclusion and Future Work”

Conclusion and Future Work

Main Contributions:

- Mitigating Sim-to-real gap in robotic simulation by introducing the high-performance bridge
- High-performance robotic components (Camera, LiDAR, IMU and ROS control plugins) in Unity3D
- Kinova Movo digital twin as the evaluation platform

Limitations:

- Nvidia Flex Editor Plugin is available only for windows
- HDRP Mode is not supported for Flex

Future Works:

- More Benchmarking and sim-to-real evaluations
- Reinforcement learning Performance Comparison
- Project Alpha / Beta Releases

References

[1] Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym." arXiv preprint arXiv:1606.01540 (2016).

[2] Babaians, Edwin, Mohsen Tamiz, Yaser Sarfi, Amir Mogoei, and Esmaeil Mehrabi. "ROS2Unity3D; High-Performance Plugin to Interface ROS with Unity3d engine." In 2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium, pp. 59-64. IEEE, (2018).

[3] Juliani, Arthur, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy et al. "Unity: A general platform for intelligent agents." arXiv preprint arXiv:1809.02627 (2018).

[4] Tobin, Joshua P. "Real-World Robotic Perception and Control Using Synthetic Data." PhD diss., UC Berkeley, 2019.

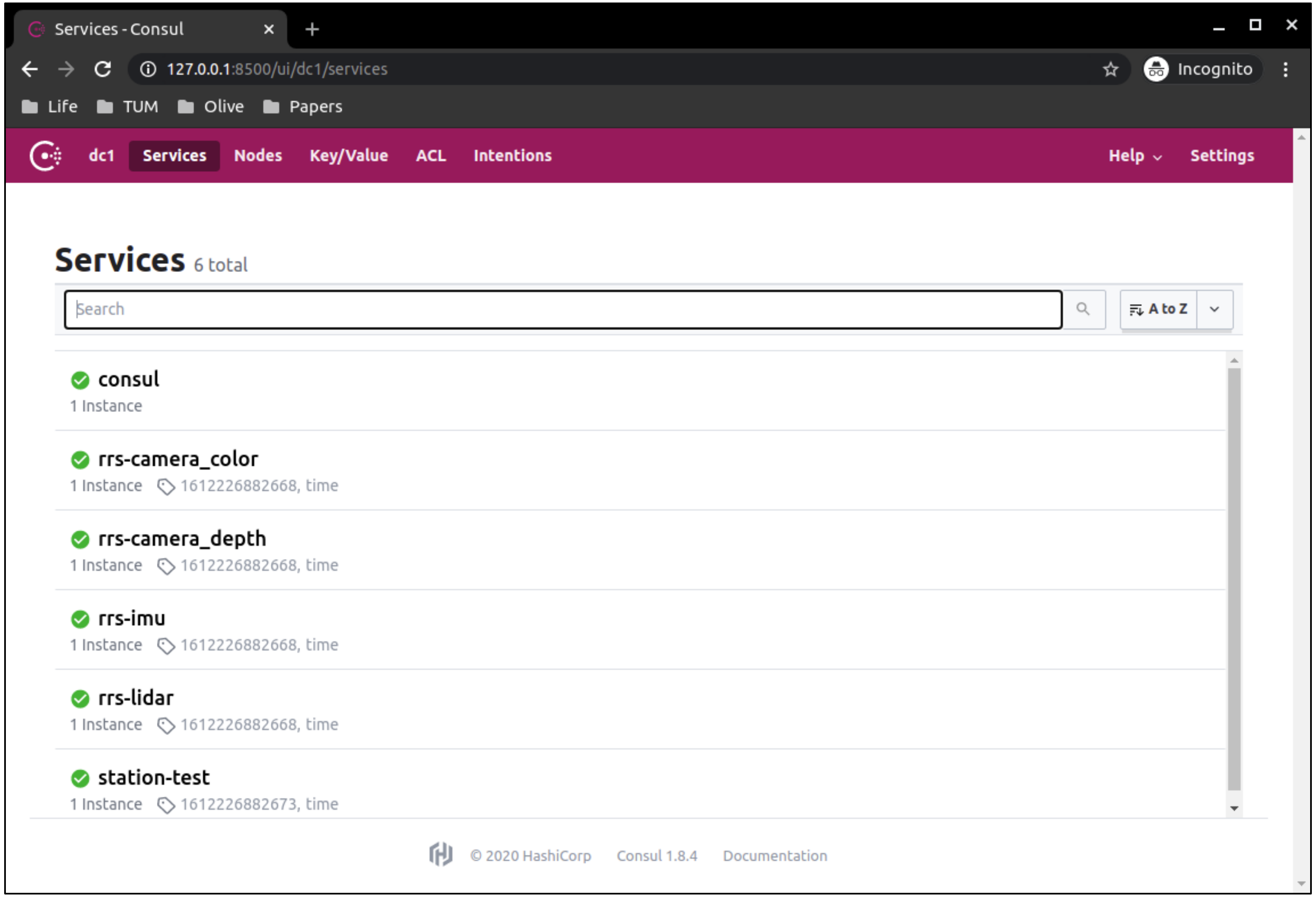
[5] Erez, Tom, Yuval Tassa, and Emanuel Todorov. "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx." In *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 4397-4404. IEEE, 2015.

[6] Kober, Jens, J. Andrew Bagnell, and Jan Peters. "Reinforcement learning in robotics: A survey." *The International Journal of Robotics Research* 32, no. 11 (2013): 1238-1274.

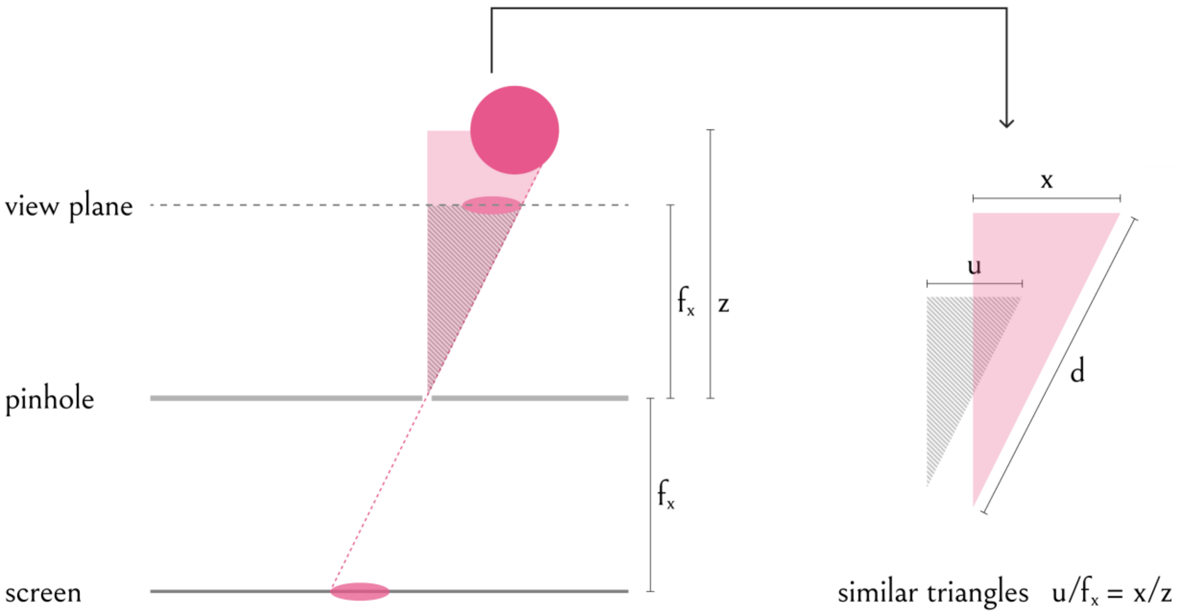
[7] Peng, Xue Bin, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. "Sim-to-real transfer of robotic control with dynamics randomization." In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803-3810. IEEE, 2018.

“Backup Slides”

Service Discovery and Health Checking (GUI)



Point Cloud Generation



$$\begin{bmatrix}u\\v\\1\end{bmatrix} = \frac{1}{z}\underbrace{\boldsymbol{K}}_{3\times3}\underbrace{[\boldsymbol{R}\mid\boldsymbol{t}]}_{3\times4}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

$$\begin{bmatrix}u\\v\\1\\1/z\end{bmatrix} = \frac{1}{z}\underbrace{\begin{bmatrix}\boldsymbol{K} & 0\\ \boldsymbol{0} & 1\end{bmatrix}}_{4\times4}\underbrace{\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ \boldsymbol{0} & 1\end{bmatrix}}_{4\times4}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

$$\begin{bmatrix}x\\y\\z\\1\end{bmatrix} = z\underbrace{\begin{bmatrix}1/f_x & 0 & 0 & 0\\ 0 & 1/f_y & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}}_{inverse\ with\ c_x, c_y, S=0}\begin{bmatrix}u\\v\\1\\1/z\end{bmatrix}$$ (projection matrix)

$$\begin{bmatrix}1/f_x & -S/(f_x f_y) & (S c_y - c_x f_y)/(f_x f_y)\\ 0 & 1/f_y & -c_y/f_y\\ 0 & 0 & 1\end{bmatrix}$$

$$x = \frac{uz}{f_x}$$

$$\boldsymbol{K} = \begin{bmatrix}f_x & S & c_x\\ 0 & f_y & c_y\\ 0 & 0 & 1\end{bmatrix}$$ (Intrinsic matrix)

RGB + Depth = Point Cloud

Chair of Media Technology
Department of Electrical and Computer Engineering
Technical University of Munich

Detailed Comparisions

Name	ROS#	ROS2	ISAAC SIM Unity*	LMT
Affiliation	Simens AG	Dyno Robotics	NVIDIA	TUM
Autor/Project Lead	Dr. Martin Bischoff	Fredrik Löfgren	Dr. David Weikersdorfer	Edwin Babaians
Published Year	2016	2019	2020	2021
Software Dependency	Nothing	ROS2	ISAAC GEM API	Nothing
Network System	WebSocket's + ROS Bridge Server	DDS	Native TCP/IP	ZMQ TCP/IP and UDP
Serialization System	JSON	DDS + (ROS2-ROS1 Bridge)	Cap'n Proto	Google Proto Buffers
Time Synchronization	No	DDS	app-clock	NTP
Simulated Sensors	Camera	LiDAR	Camera, Depth Camera, LiDAR, IMU,	Camera, Depth Camera, LiDAR, IMU
Simulated Actuators	Based on Rigid Bodies	Based on Rigid Bodies	Not Yet – They switched to omniverse	Based on Articulation Bodies
Tested Platforms	Windows, Linux	Windows	Windows	Windows, Linux, Android
Included Robot	TurtleBot	TurtleBot	Nvidia Robots	Kinova Movo
Hardware Dependency	No	No	Nvidia's Xavier, AWS,	No

*https://developer.nvidia.com/isaac-sdk

“Credits and Licenses”

Main Contributors:
1) Edwin Babaians
2) Leox Mojtaba Karimi
3) Premankur Banerjee

Student testing/improvements:
1) Mokchah, Roua (BA)
2) Dihab Malek (BA)
3) Premankur Banerji (FP and Working Student)
4) Rayene Massoud (BA)
5) Mohamad Motazedi (FP)

Special Thanks:
1) Yaser Sarfi (Arsam robotics)
2) Mohsen Tamiz (Arsam robotics)

Licensing:
[1] Google Proto buffers: https://github.com/protocolbuffers/protobuf
[2] ZMQ Project: https://zeromq.org/
[3] ML-Agents: https://github.com/Unity-Technologies/ml-agents
[4] Nvidia-Flex: https://github.com/NVIDIAGameWorks/FleX
[5] Unity Technologies: https://unity.com/
[6] ppconsul: https://github.com/oliora/ppconsul
[7] Hachi Corp: https://www.hashicorp.com/
[8] Turbo Jpeg: https://libjpeg-turbo.org/
[9] URDF Importer: https://github.com/Unity-Technologies/URDF-Importer