GPU-ACCELERATED 3D SCENE RECONSTRUCTION FOR ROS2 SYSTEMS

Carlo Wiesse

13 September 2022

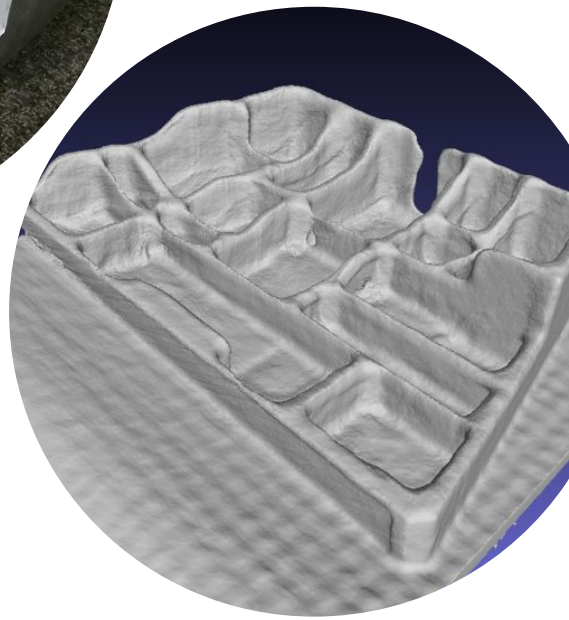
Agenda

- What is 3D scene reconstruction and how is it done?
- What tools are currently available in ROS2 for this purpose?
- What is RTAB-Map and how does this tool perform in simulation?
- What is Nvblox and how does this GPU-accelerated tool perform in simulation?

3D SCENE RECONSTRUCTION

Introduction and ROS2 tools

Introduction



What is it?

Scene reconstruction is the process of creating a **digital version of a real-world object** from pictures or scans of the object.

How does it work?

There are **many ways.** Some of the most popular ones being:

- Matching **key-points** from images to create **point-clouds**.
- Matching volumetric data structures like **TSDFs** to create **meshes**.

What is available on ROS2 for this purpose?

- **RTAB-Map** https://github.com/introlab/rtabmap_ros
- **Nvblox** https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nvblox

RTABMAP

Developed by Mathieu Labbé at IntroLab (University of Sherbrooke, Canada)

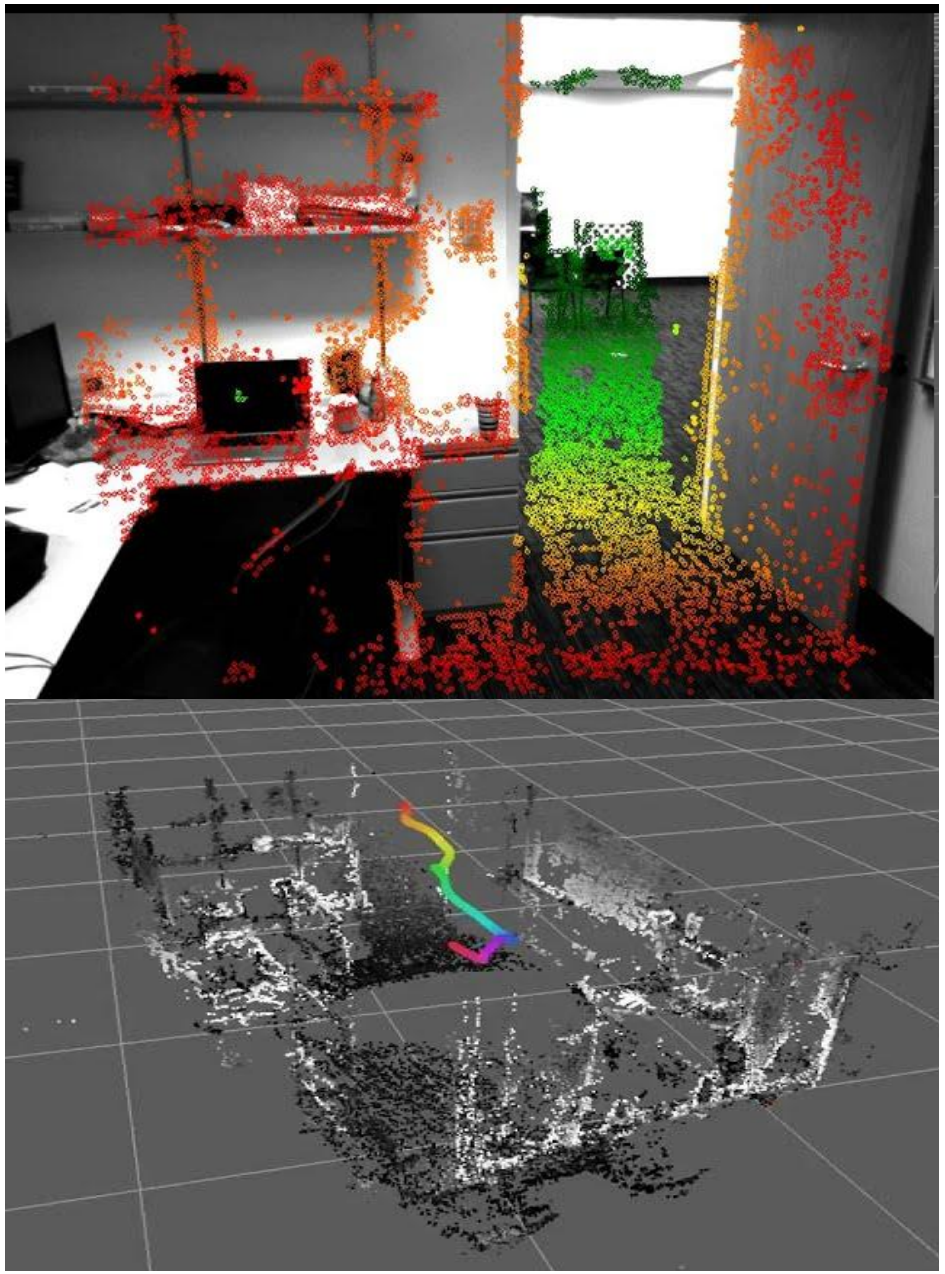
CREATING GROWTH, ENHANCING LIVES

What is it?

Real-Time Appearance-Based Mapping, also known as **RTAB-Map**, is a Simultaneous Localization and Mapping (SLAM) approach that uses RGB-D information to create a point-cloud representation of the 3D scene.

How does it work?

- This approach extracts SURF features and key-points from RGB images to later be used in a bag-of-words approach.
- The loop closure detector then determines how likely a new image comes from a previous location or a new location based on feature matching.
- Finally, the matched key-points and the corresponding depth images are used to optimize the final point-cloud representing the 3D scene.



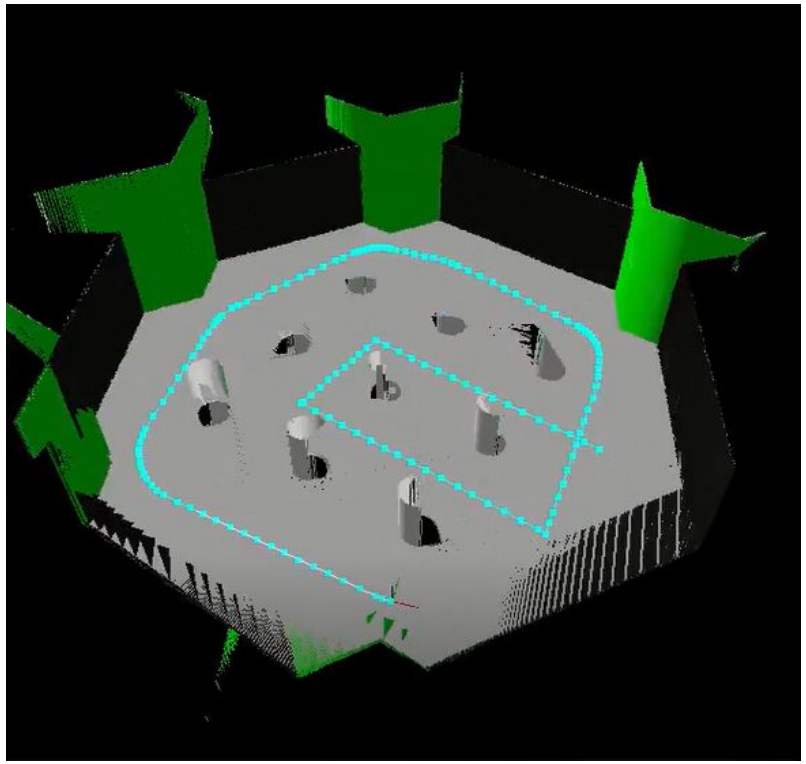
Applications

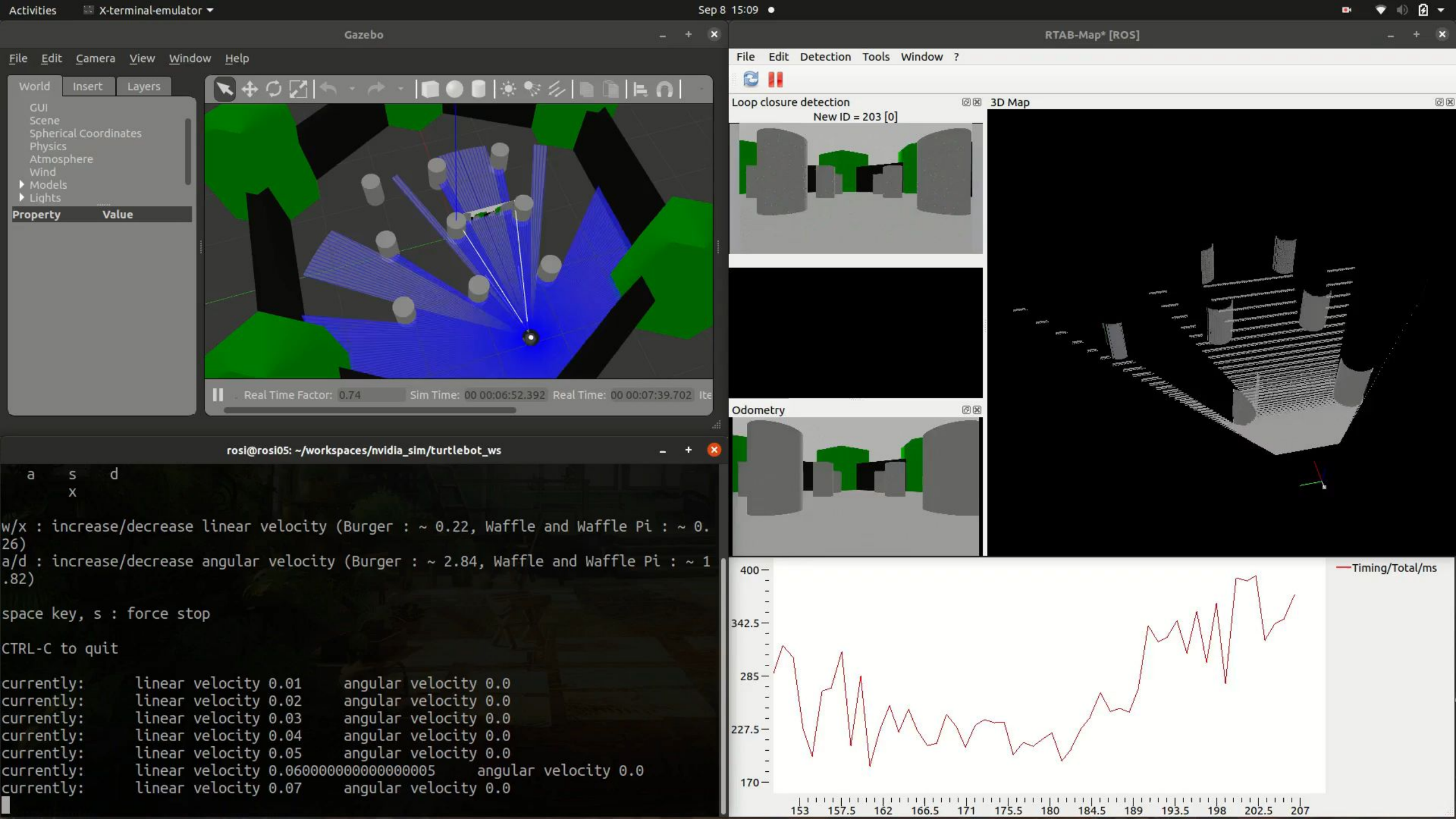
- Car mapping and localization with CitySim simulator and a CAT Vehicle (2022)
- Indoor drone visual navigation example using move_base, PX4 and mavros (2021)

See more at the IntroLab website (http://introlab.github.io/rtabmap)

TurtleBot Test Parameters

Ubuntu Distro:	20.04
ROS Distro:	Foxy
CPU:	Intel i7-7700HQ
GPU:	NVIDIA Quadro M1200 (CC 5)
Robot Model:	TurtleBot3 (Burger)
RGB-D Sensor:	RealSense D415 Camera
Map:	TurtleBot3 World
Simulator:	Gazebo




Activities
X-terminal-emulator
Sep 8 15:09
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Real Time Factor: 0.74 Sim Time: 00 00:06:52.392 Real Time: 00 00:07:39.702
RTAB-Map* [ROS]
File Edit Detection Tools Window ?
Loop closure detection
New ID = 203 [0]
Odometry
3D Map
rosi@rosi05: ~/workspaces/nvidia_sim/turtlebot_ws
a s d
x
w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle and Waffle Pi : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.82)
space key, s : force stop
CTRL-C to quit
currently: linear velocity 0.01 angular velocity 0.0
currently: linear velocity 0.02 angular velocity 0.0
currently: linear velocity 0.03 angular velocity 0.0
currently: linear velocity 0.04 angular velocity 0.0
currently: linear velocity 0.05 angular velocity 0.0
currently: linear velocity 0.060000000000000005 angular velocity 0.0
currently: linear velocity 0.07 angular velocity 0.0
Timing/Total/ms
400
342.5
285
227.5
170
153 157.5 162 166.5 171 175.5 180 184.5 189 193.5 198 202.5 207

NVBLOX

Developed by NVIDIA as part of the NVIDIA ISAAC ROS toolbox

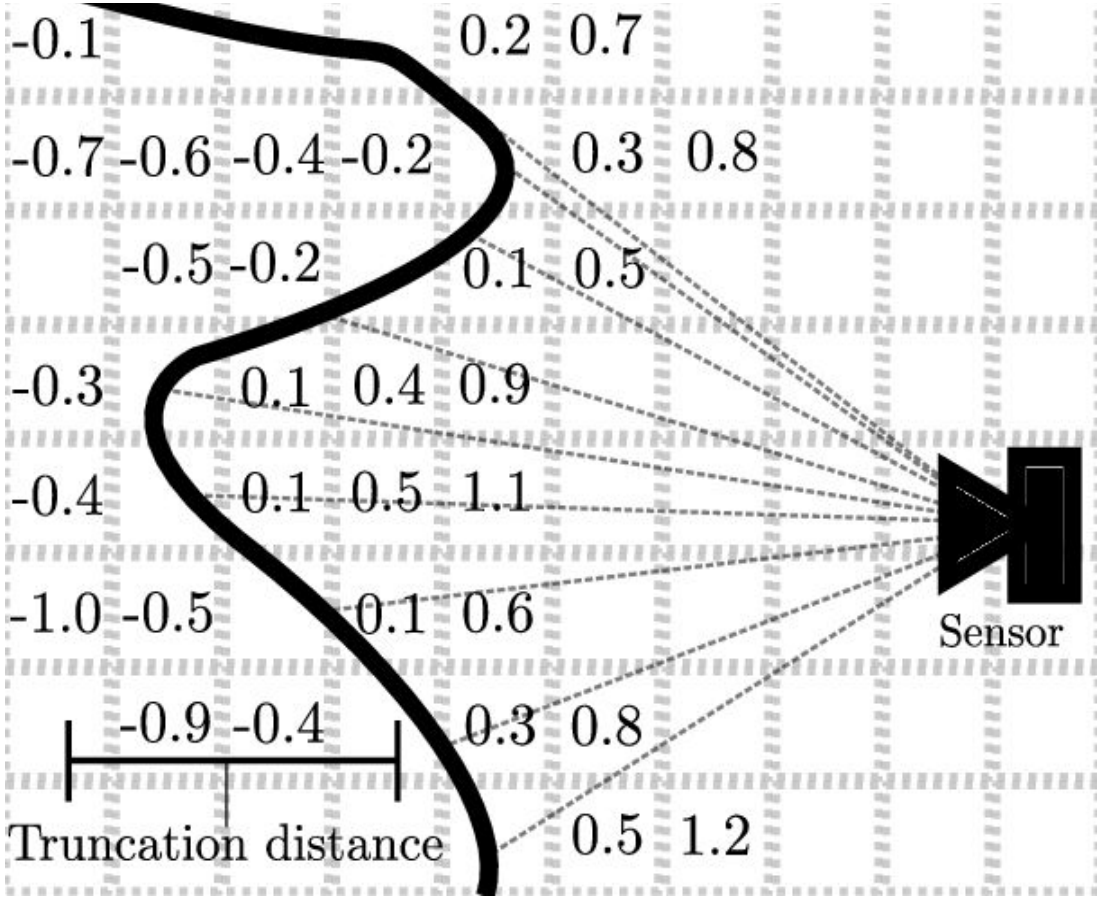
CREATING GROWTH, ENHANCING LIVES

What is it?

Nvblox is a GPU-accelerated approach that uses Truncated Signed Distance Fields (TSDFs) to create a mesh of the 3D scene.

How does it work?

- This approach produces TSDFs from depth images to later be used in an Iterative Closest Point (ICP) algorithm.
- The ICP algorithm then determines how likely a new image comes from a previous location or a new location based on the difference between TSDFs.
- Finally, the matched TSDFs are converted to a triangular mesh using the Marching Cubes algorithm.



Activities X-terminal-emulator Sep 9 15:06

Gazebo

File Edit Camera View Window Help

World Insert Layers

GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights

Property Value

Real Time Factor: 0.98 Sim Time: 00 00:05:53.433 Real Time: 00 00:05:56.256 Ite

RViz*

File Panels Help

Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Goal Pose Publish Point

Displays

Fixed Frame	map
Background Color	48; 48; 48
Frame Rate	30
Global Status: Ok	
Fixed Frame	OK
Grid	
Odometry	
Image	
Image	
NvbloxMesh	

Reference Frame
The TF frame this grid will use for its origin.

Add Duplicate Remove Rename

Image

Image

Reset 31 fps

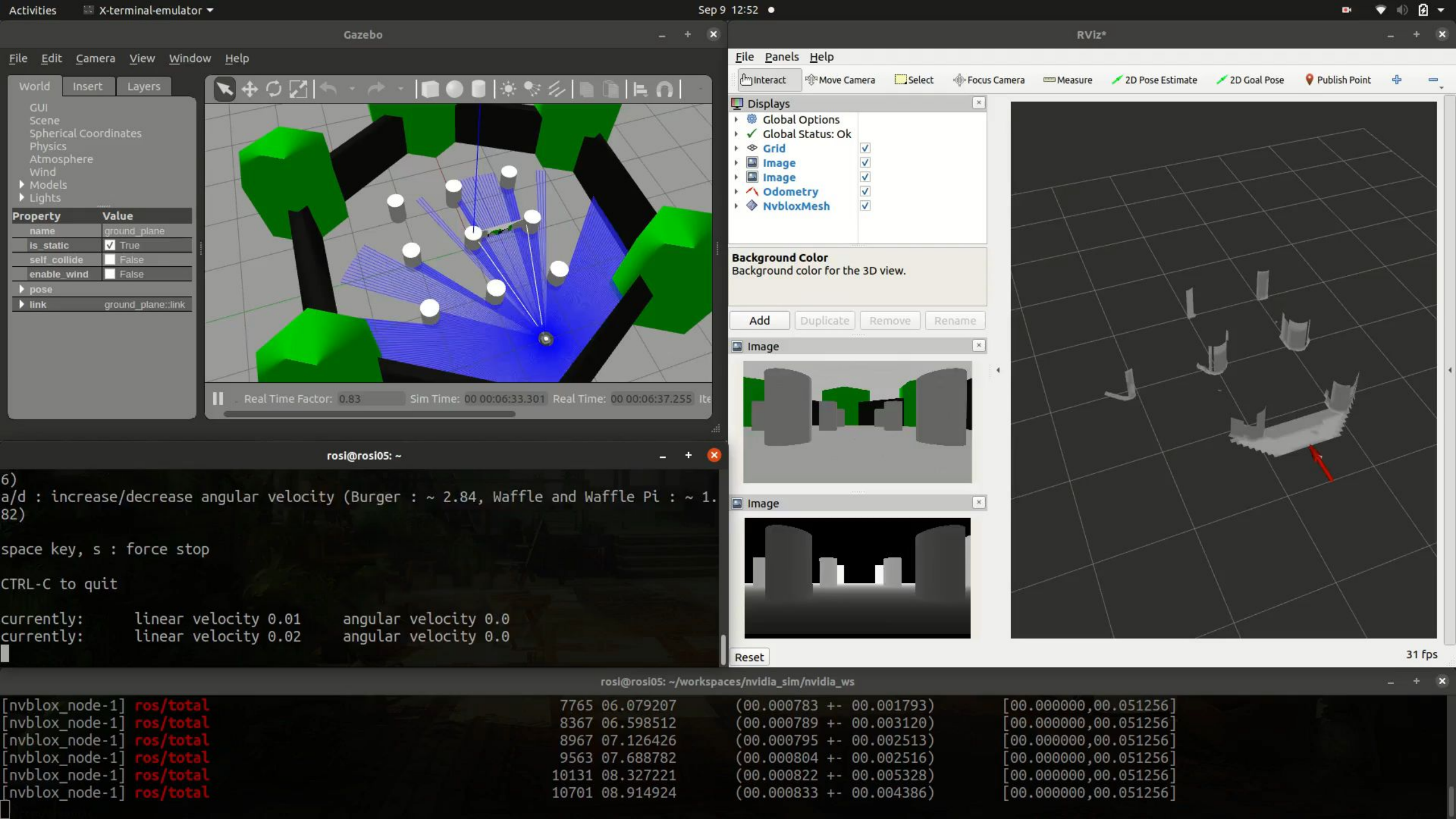
rosi@rosi05: ~

```
CTRL-C to quit

currently:      linear velocity 0.01     angular velocity 0.0
currently:      linear velocity 0.02     angular velocity 0.0
currently:      linear velocity 0.03     angular velocity 0.0
currently:      linear velocity 0.04     angular velocity 0.0
currently:      linear velocity 0.05     angular velocity 0.0
currently:      linear velocity 0.060000000000000005     angular velocity 0.0
currently:      linear velocity 0.07     angular velocity 0.0
```

rosi@rosi05: ~/workspaces/nvidia_sim/nvidia_ws

```
[nvblox_node-1] ros/total      4818 01.809493      (00.000376 +- 00.001220)      [00.000000,00.016872]
[nvblox_node-1] ros/total      5418 02.081984      (00.000384 +- 00.002197)      [00.000000,00.016872]
[nvblox_node-1] ros/total      6012 02.357988      (00.000392 +- 00.001439)      [00.000000,00.016872]
[nvblox_node-1] ros/total      6618 02.635399      (00.000398 +- 00.001460)      [00.000000,00.016872]
[nvblox_node-1] ros/total      7186 02.941767      (00.000409 +- 00.001965)      [00.000000,00.016872]
[nvblox_node-1] ros/total      7766 03.263581      (00.000420 +- 00.001921)      [00.000000,00.016872]
```


Activities
X-terminal-emulator
Sep 9 12:52
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Real Time Factor: 0.83
Sim Time: 00 00:06:33.301
Real Time: 00 00:06:37.255
RViz*
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Goal Pose
Publish Point
Displays
Global Options
Global Status: Ok
Grid
Image
Image
Odometry
NvbloxMesh
Background Color
Background color for the 3D view.
Add
Duplicate
Remove
Rename
Image
Image
Reset
31 fps
rosi@rosi05: ~
6)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.
82)
space key, s : force stop
CTRL-C to quit
currently: linear velocity 0.01 angular velocity 0.0
currently: linear velocity 0.02 angular velocity 0.0
rosi@rosi05: ~/workspaces/nvidia_sim/nvidia_ws
[nvblox_node-1] ros/total 7765 06.079207 (00.000783 +- 00.001793) [00.000000,00.051256]
[nvblox_node-1] ros/total 8367 06.598512 (00.000789 +- 00.003120) [00.000000,00.051256]
[nvblox_node-1] ros/total 8967 07.126426 (00.000795 +- 00.002513) [00.000000,00.051256]
[nvblox_node-1] ros/total 9563 07.688782 (00.000804 +- 00.002516) [00.000000,00.051256]
[nvblox_node-1] ros/total 10131 08.327221 (00.000822 +- 00.005328) [00.000000,00.051256]
[nvblox_node-1] ros/total 10701 08.914924 (00.000833 +- 00.004386) [00.000000,00.051256]

SUMMARY / CONCLUSIONS

RTAB-Map vs. Nvblox

Both approaches have their merits and their limitations.

- RTAB-Map is slower, but more accurate than Nvblox.
- RTAB-Map isn’t suitable to deal with dynamic objects entering the scene.

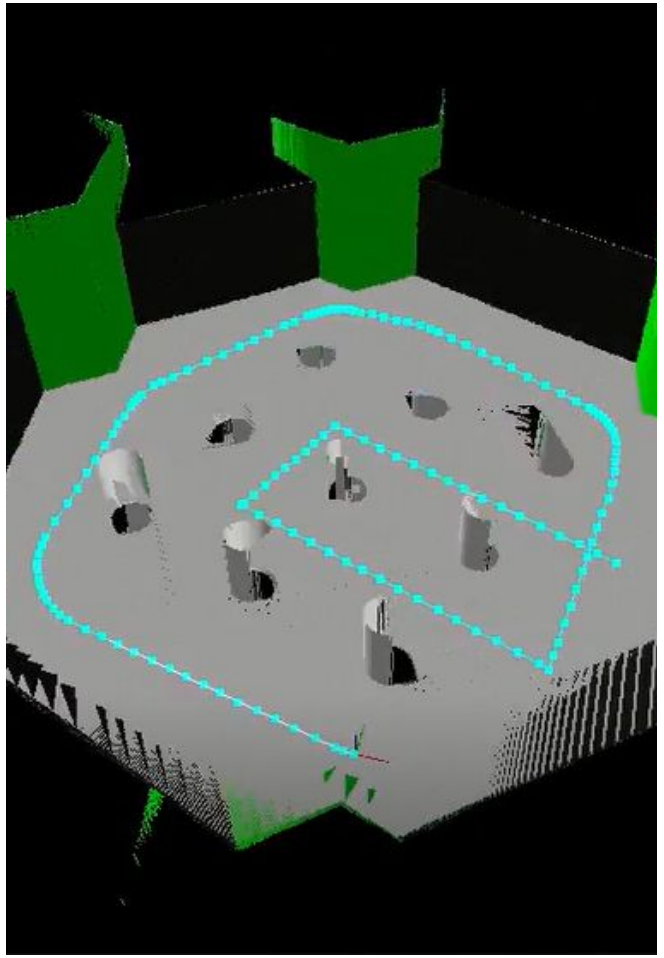
- Nvblox can become more accurate by decreasing voxel size. However, this will result in slower performance.
- Nvblox time performance decreases as the size of the mesh grows. Hence, for large reconstruction areas, the longer processing time will become less suitable for dealing with dynamic objects entering the scene.

Key-points vs. TSDFs

The TSDF algorithm can be efficiently **parallelized** on a general-purpose graphics processor, which allowed NVIDIA to accelerate its processing time.

The ICP algorithm works as a moving average of TSDFs, which serves to **reduce noise and errors**, resulting into smooth continuous surfaces.

Key-point-based approaches require **additional processing** to distinguish between actual and outlier features in erroneous data.



THANK YOU!

Carlo Wiesse

Email
carlo_wiesse@artc.a-star.edu.sg

RTAB-Map Github
https://github.com/introlab/rtabmap_ros

Nvblox Github
https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nvblox