
EKUMEN

Gerardo Puga

Electronic Engineer by training.

In a previous life I specialized in real-time embedded systems applied to global positioning system receivers.

Six years ago I started working at Ekumen, and since then I've taken part in a number of projects related to mobile robotics, embedded systems and some gaming.

EKUMEN

Beluga AMCL

A modern Monte Carlo Localization implementation for ROS

Outline

- Introduction
- Beluga core library
- Beluga AMCL
- Validation
- Availability
- Future work

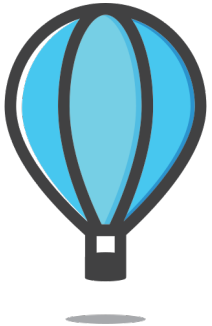
BELUGA

Introduction

Why?

Nav2 AMCL already exists!

- Key component of the ROS navigation stack for many years.
- It's widely tested in the field.
- Lots of people know how to use it.

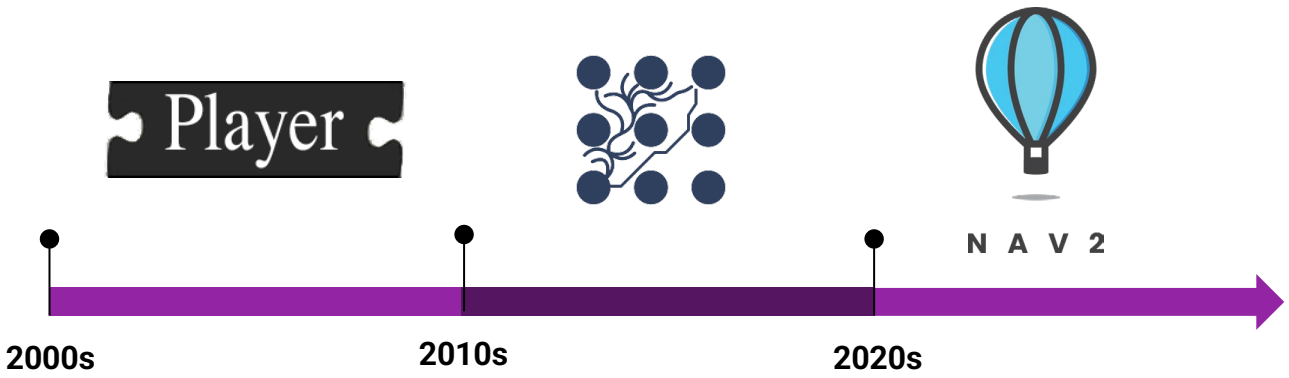


N A V 2

Why?

but...

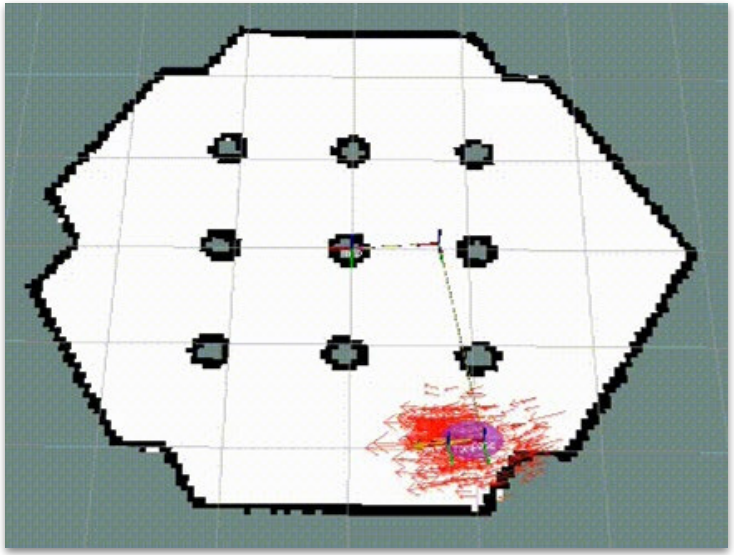
- Aging codebase; parts of the code are 24 years old.
- Development has slowed down over the years.
- Low modularity and poor test coverage make it hard to extend.



Goals

Two years ago, we began reimagining AMCL as an internal development exercise.

- Written in C++17
- Highly modular, layered design
- Conformant to currently accepted best practices
- High test coverage
- ROS native, but with ROS-independent core
- Compatible with Nav2 AMCL



EKUMEN

Beluga core library

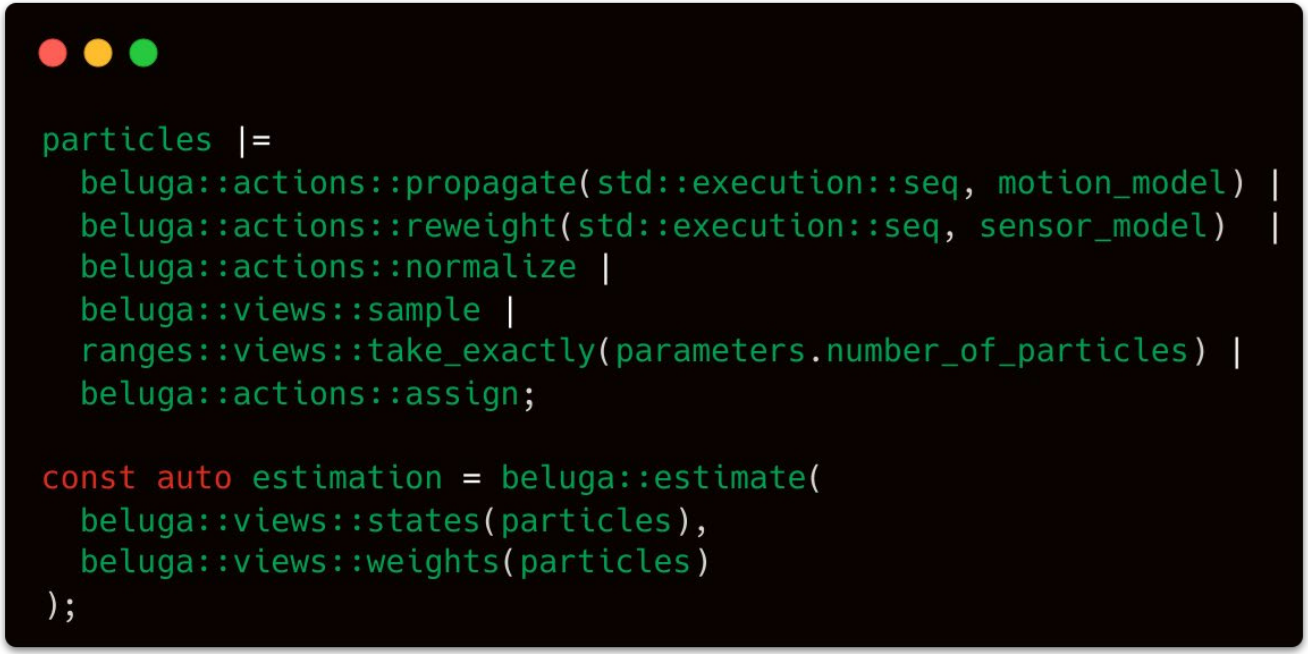
Beluga core library

- C++17, header-only, library of reusable components allowing the creation of Monte Carlo Localization particle filters.
- Relies on generic programming and ranges semantics to provide implementations that are both flexible and efficient.
- Can be used in any modern C++ project, not only ROS ones.

Provides components for

- Sensor models
- Motion models
- Resampling algorithms
- Estimation algorithms
- More ...

Beluga core library



```
particles |=
  beluga::actions::propagate(std::execution::seq, motion_model) |
  beluga::actions::reweight(std::execution::seq, sensor_model)  |
  beluga::actions::normalize |
  beluga::views::sample |
  ranges::views::take_exactly(parameters.number_of_particles) |
  beluga::actions::assign;

const auto estimation = beluga::estimate(
  beluga::views::states(particles),
  beluga::views::weights(particles)
);
```

EKUMEN

Beluga AMCL

Beluga AMCL

Beluga AMCL is a ROS package for 2D localization built out of Beluga components

Compatible with Nav2 AMCL by design

- Same features, same sensor and motion models, same resampling algorithms.
- Same topics and services
- Same parameters

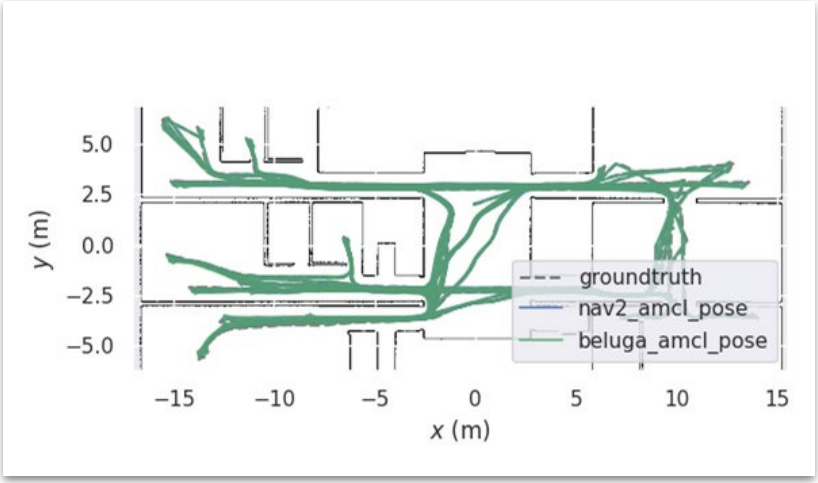
But also brings benefits

- Project in active development.
- Modern codebase based on the Beluga core library
- > 95% test coverage
- Available for both ROS 1 and ROS 2 from the same unified ROS agnostic core, no porting changes back and forward between versions

Validation

Early tests

- The first few versions of the software were validated using a combination of our own robots along with simulation.
- Very valuable data was also provided to us by people in our network running tests using their own real world robot datasets.



Benchmarking

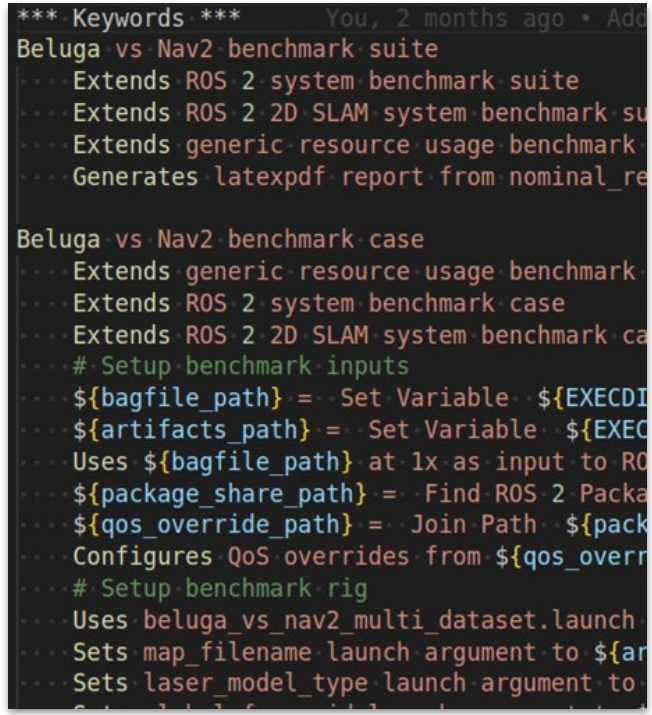
As the complexity grew, however, we needed better tooling to be able to perform **systematic** and **repeatable** assessments.

Enter LAMBKIN, a separate project we recently made publicly available, providing:

- Reproducible environments
- Declarative benchmark definitions
- Standard performance metrics
- Report generation

LAMBKIN is still a work in progress, but you can already check it out at:

https://github.com/Ekumen-OS/lambkin



Benchmarking

LAMBKIN does the heavy lifting of running the experiments, but we still needed the data to experiment with.

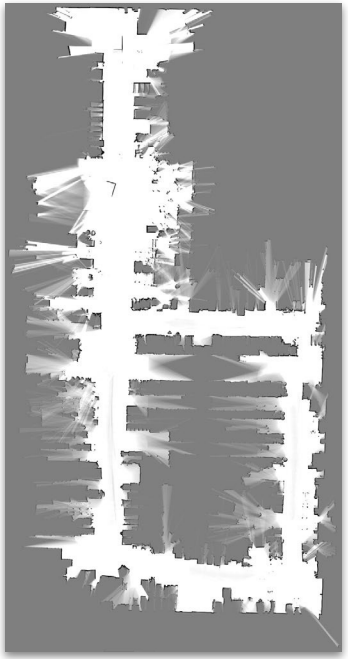
While not specifically created for MCL localization evaluation, a number of publicly available datasets were suitable for the task.

Real bot datasets:

- Willow Garage
- TorWIC Mapping
- TorWIC SLAM
- Magazino Dataset
- More...

Simulated bot datasets:

- 24hs diff drive robot
- 24hs omni robot.



Performance

Table 37: APE metrics aggregated across all iterations of `2022-10-12_hallway_straight_ccw` for the Likelihood field sensor model.

Implementation	Likelihood field sensor model			
	median	mean	std	worst-case
Beluga AMCL	0.077 m	0.089 m	0.054 m	0.278 m
Nav2 AMCL	0.088 m	0.095 m	0.055 m	0.274 m

Table 38: APE metrics aggregated across all iterations of `2022-10-12_hallway_straight_ccw` for the Beam sensor model.

Implementation	Beam sensor model			
	median	mean	std	worst-case
Beluga AMCL	0.052 m	0.058 m	0.036 m	0.203 m
Nav2 AMCL	0.054 m	0.058 m	0.033 m	0.182 m

Resource usage

Both nodes have matching average CPU and Peak Resident Set Size

Table 6: Average CPU and peak Resident Set Size (RSS) values.

Implementation	Likelihood Field		Beam	
	cpu	rss	cpu	rss
Beluga AMCL	5.4%	675 MB	8.5%	111 MB
Nav2 AMCL	5.8%	680 MB	7.9%	679 MB

Intel Core Ultra 7, 3.8GHz. 18 minutes running time, 300mx300m map @ 0.05m/px (6000x6000 px)

Resource usage (the omen)

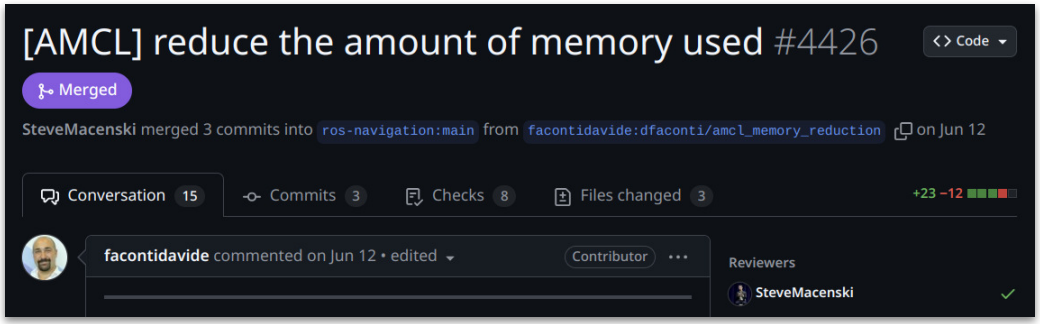
Both nodes have matching average CPU and Peak Resident Set Size

Table 6: Average CPU and peak Resident Set Size (RSS) values.

Implementation	Likelihood Field		Beam	
	cpu	rss	cpu	rss
Beluga AMCL	5.4%	675 MB	8.5%	111 MB
Nav2 AMCL	5.8%	680 MB	7.9%	679 MB

Intel Core Ultra 7, 3.8GHz. 18 minutes running time, 300mx300m map @ 0.05m/px (6000x6000 px)

Now, just before handing over these slides, we realized Davide Faconti had created PR 4426 back in June.



Resource usage (the optimizer)

Table 6: Average CPU and peak Resident Set Size (RSS) values.

Implementation	Likelihood Field		Beam	
	cpu	rss	cpu	rss
Beluga AMCL	5.4%	675 MB	8.5%	111 MB
Nav2 AMCL	5.8%	680 MB	7.9%	679 MB

Table 6: Average CPU and peak Resident Set Size (RSS) values.

Implementation	Likelihood Field		Beam	
	cpu	rss	cpu	rss
Beluga AMCL	5.2%	675 MB	8.6%	110 MB
Nav2 AMCL	5.5%	286 MB	7.6%	286 MB

Resource usage (the chase)

Table 6: Average CPU and Resident Set Size (RSS) values.

Implementation	Good Field		Beam	
	cpu	rss	cpu	rss
Beluga AMCL	8.6%	675 MB	8.6%	110 MB
Nav2 AMCL	7.6%	286 MB	7.6%	286 MB

Resource usage (the happy ending)

Fortunately, the problem was easy to find and fix .

Thank you Davide, you made us both better.

Table 6: Average CPU and peak Resident Set Size (RSS) values.

Implementation	Likelihood Field		Beam	
	cpu	rss	cpu	rss
Beluga AMCL	5.5%	250 MB	8.6%	110 MB
Nav2 AMCL	5.7%	285 MB	7.6%	285 MB

Availability

You can try it today!

The source code is available in Github (Apache 2.0 license).

Beluga packages are available for all current stable ROS distributions:

- ROS 2 Humble, Iron and Jazzy
- ROS 1 Noetic

Migration is simple!

Trying Beluga AMCL is as simple as updating your launch files.

```
    ),
    Node(
        package='nav2_amcl',
        executable='amcl',
        parameters=[localization_params_file],
    ),
    Node(
```

EKUMEN

Future work

NDT based localization

Sensor models for both planar and 3D sensor data.

Beluga AMCL ROS nodes for NDT-MCL localization.

2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Normal Distributions Transform Monte-Carlo Localization (NDT-MCL)

Jari Saarinen, Henrik Andreasson, Todor Stoyanov and Achim J. Lilienthal

Abstract— **Industrial applications often impose hard requirements on the precision of autonomous vehicle systems. As a consequence industrial Automatically Guided Vehicle (AGV) systems still use high-cost infrastructure based positioning solutions. In this paper we propose a map based localization method that fulfills the requirements on precision and repeatability, typical for industrial application scenarios. The proposed method - Normal Distributions Transform Monte Carlo Localization (NDT-MCL) is based on a well established probabilistic framework. In a novel contribution, we formulate the MCL localization approach using the Normal Distributions Transform (NDT) as an underlying representation for both map and sensor data. By relaxing the hard discretization assumption imposed by grid-map models and utilizing the piecewise continuous NDT representation the proposed algorithm achieves substantially improved accuracy and repeatability. The proposed NDT-MCL algorithm is evaluated using offline data sets from both a laboratory and a real-world industrial environments. Additionally, we report a comparison of the proposed algorithm to grid-based MCL and to a commercial localization system when used in a closed-loop with the control system of an AGV platform. In all tests the proposed algorithm is demonstrated to provide performance superior to that of standard grid-based MCL and comparable to the performance of the commercial infrastructure based positioning system.**

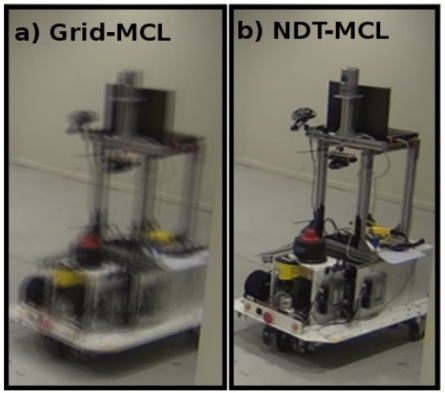


Fig. 1. Positioning accuracy visualized: Fifteen images recorded from a stationary camera overlaid in positioning accuracy test using a) grid-based MCL and b) NDT-based MCL.

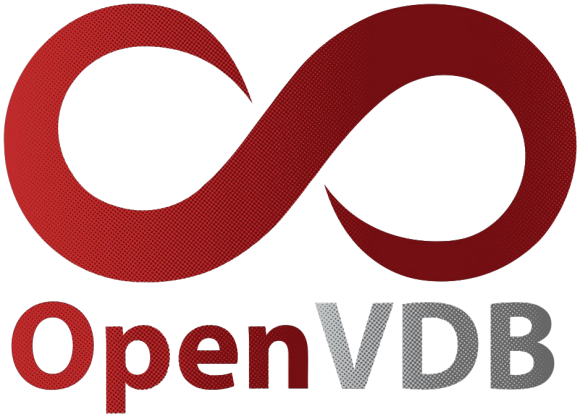
I. Introduction

Localization systems are an essential enabling component has not been reported to reach the precision requested by industrial applications [3], [15], [9]. Röwekämper et al. in [9]

3D Likelihood sensor model

Currently developing an experimental 3D likelihood model:

- Will enable MCL localization in sparse volumetric 3D maps of the environment.
- Based on VDB and OpenVDB



Conclusion

Conclusion

Visit us!

- https://github.com/Ekumen-OS/beluga, the repository.
- https://ekumen-os.github.io/beluga, the documentation.

EKUMEN

Survey!

Thank you!