Segmentação de Teto para a Localização de Robôs

Maurício Lopes e Ezequiel Simeoni

INTRODUÇÃO

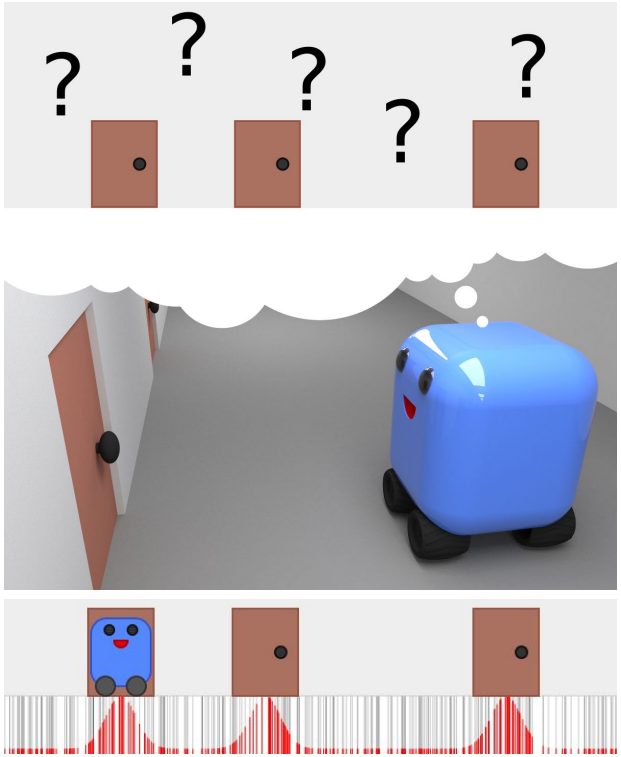
Robótica Móvel

Inteligência Artificial

Aprendizado de Máquina

Segmentação Semântica

Localização de Markov



RECURSOS

Câmera Raspberry Pi-V2

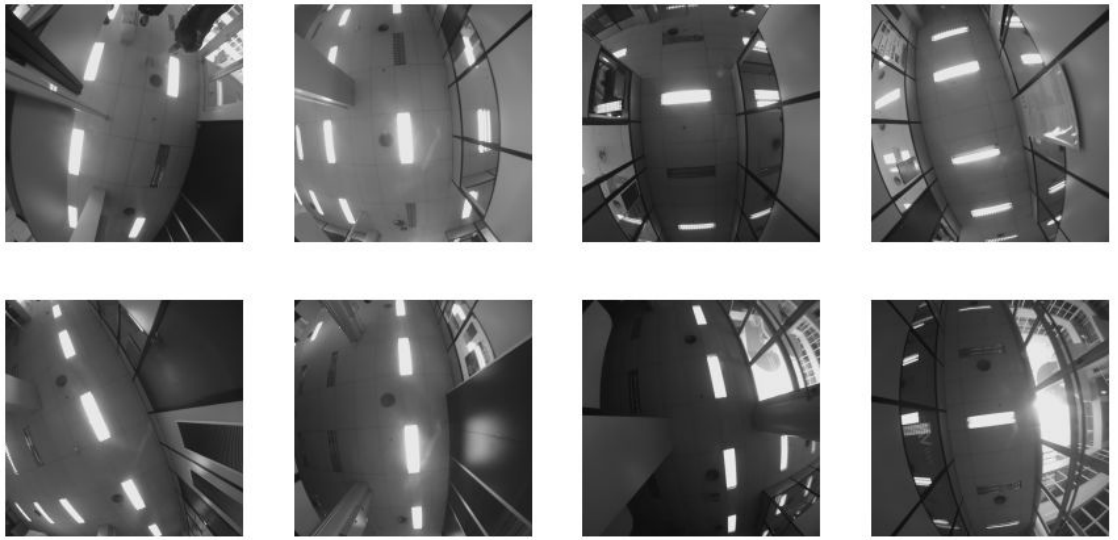
Lente Lensoul LS-WD2

Robô Neato XV-12

DATASET

2000 imagens

7º andar FACIN



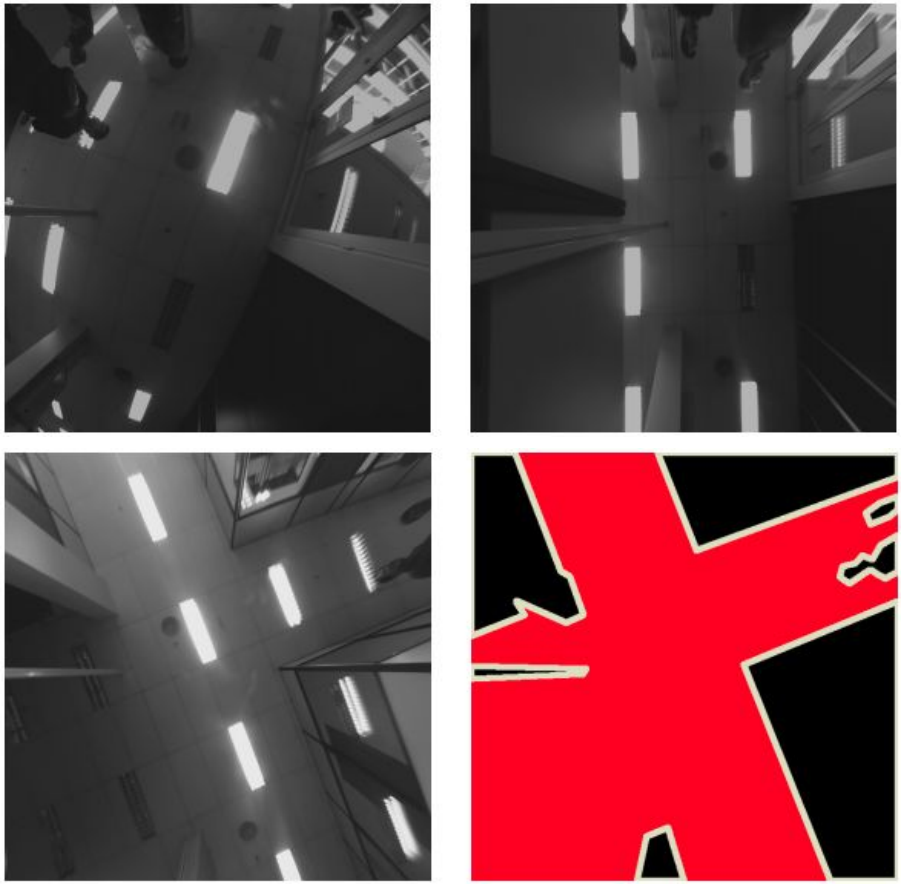
ANOTAÇÃO

126 imagens

Treino: 106

Validação: 10

Teste: 10



FCN-VGG19

VGG19 (ImageNet ILSVRC-2014)

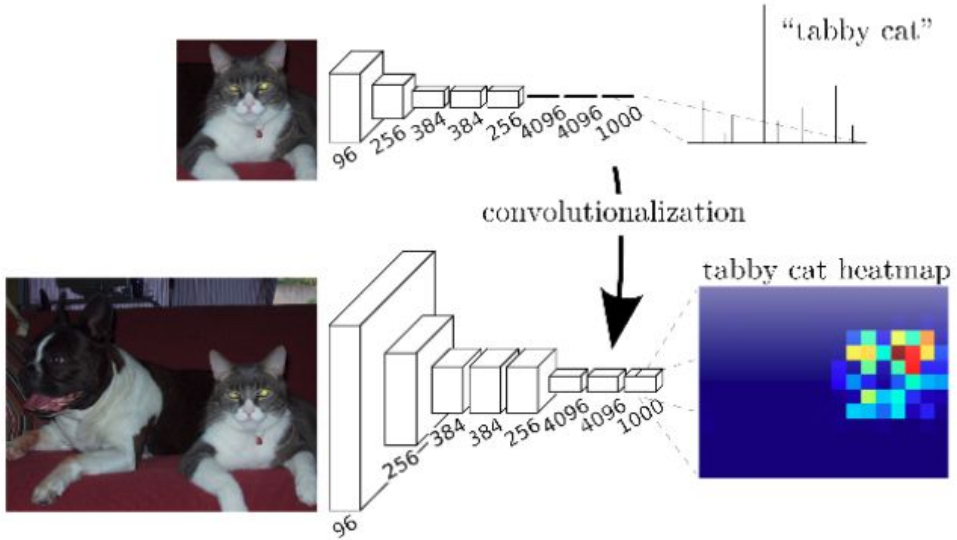
Convolução de stride fracionado ou Deconvolução

Saída: 256 x 256 x 2

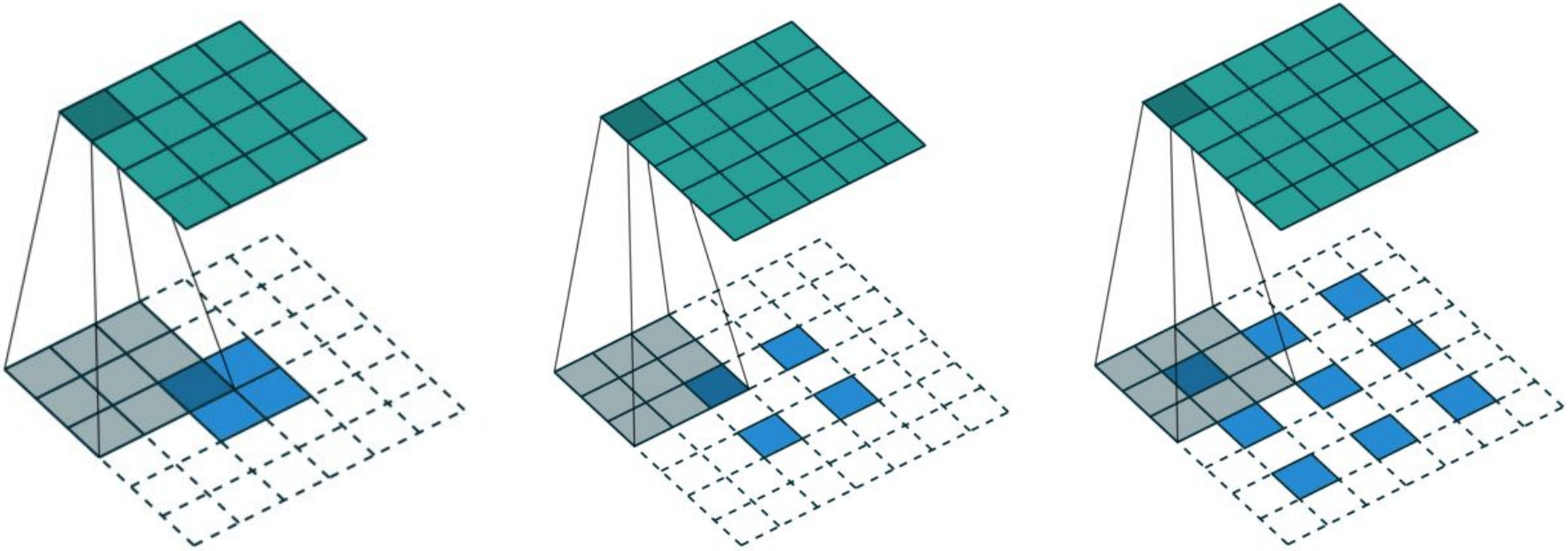
Classificador pixel a pixel

Caffe

Tensorflow



CONVOLUÇÃO TRANSPOSTA



FCN-VGG19

Github

TREINO E VALIDAÇÃO

70 épocas

51º época

```
****************** Epochs completed: 51******************
Step: 2960, Train_loss:0.0281338
Step: 2970, Train_loss:0.00893782
Step: 2980, Train_loss:0.0166022
Step: 2990, Train_loss:0.019159
Step: 3000, Train_loss:0.0220134
2017-12-06 21:39:54.279919 ---> Validation_loss: 0.223992
Step: 3010, Train_loss:0.0173228
```

RESULTADOS

(a) Resultado no ambiente de treino 1

(b) Resultado no ambiente de treino 2

(c) Resultado no ambiente de treino 3

(d) Resultado em novo ambiente

Filtro de Partículas

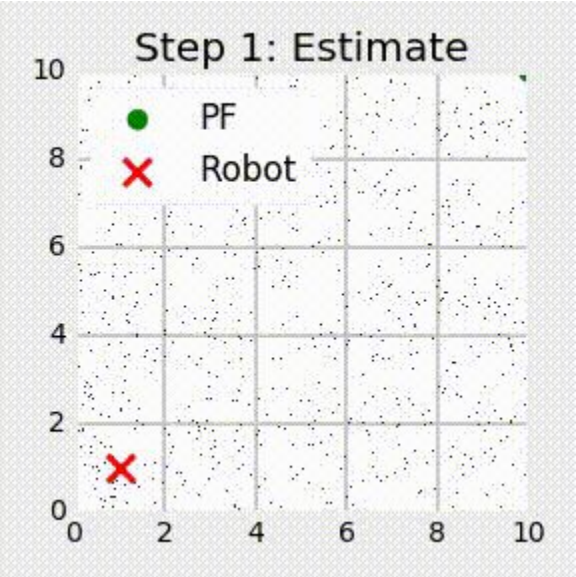
Bayesian Monte Carlo

Generic Particle Filter Algorithm

- **Randomly generate a bunch of particles**
  Particles can have position, heading, and/or whatever other state variable you need to estimate. Each has a weight (probability) indicating how likely it matches the actual state of the system. Initialize each with the same weight.
- **Predict next state of the particles**
  Move the particles based on how you predict the real system is behaving.
- **Update**
  Update the weighting of the particles based on the measurement. Particles that closely match the measurements are weighted higher than particles which don't match the measurements very well.
- **Resample**
  Discard highly improbable particle and replace them with copies of the more probable particles.

Filtro de Partículas

Estimate
Predict
Resample
Estimate
...



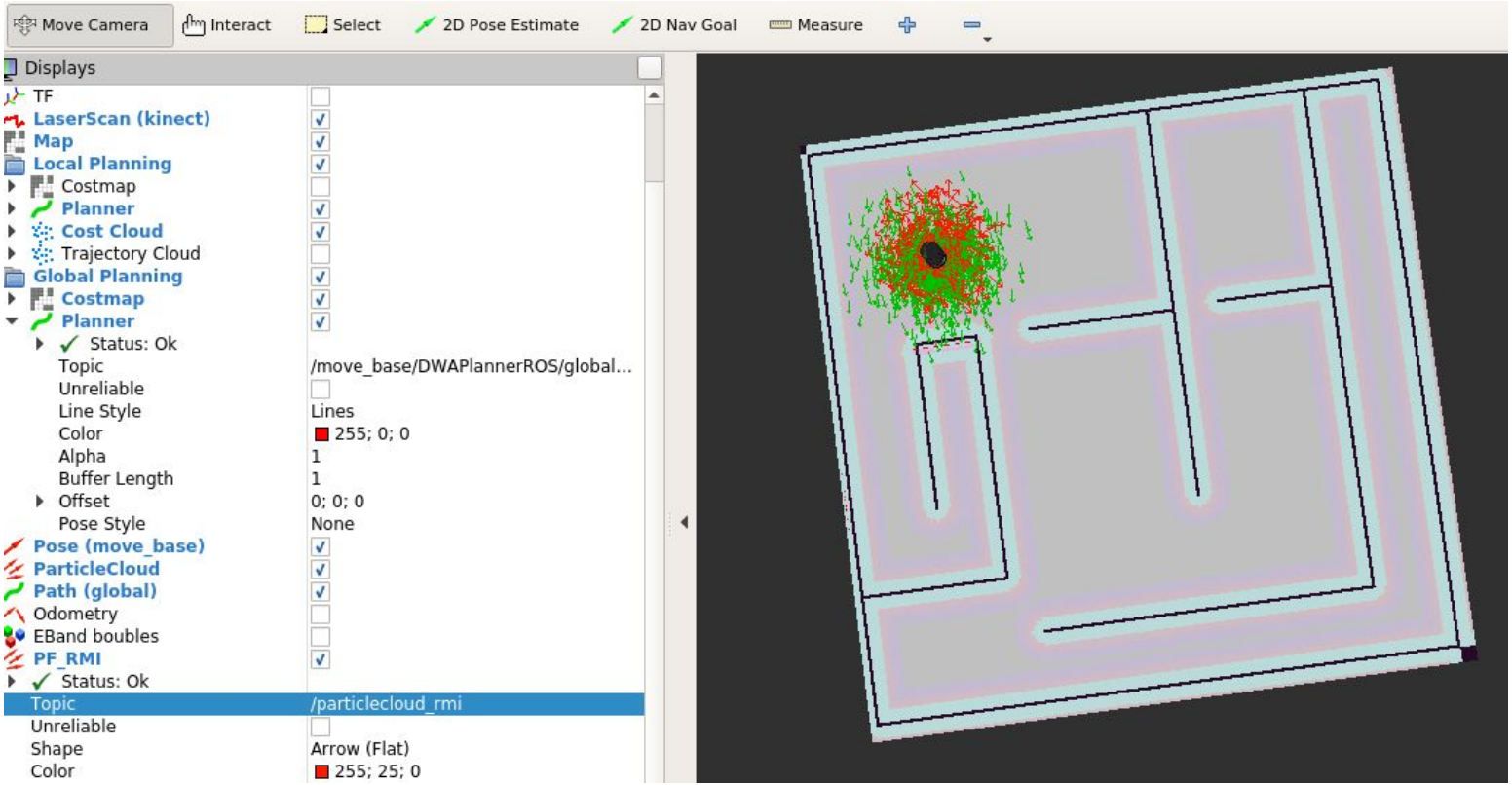
Filtro de Partículas - Código

```
def __init__(self):
    self.initialized = False
    rospy.init_node('RMI_pf')

    self.base_frame = "base_link"
    self.map_frame = "map"
    self.odom_frame = "odom"
    self.scan_topic = "scan"

    self.n_particles = 300
```

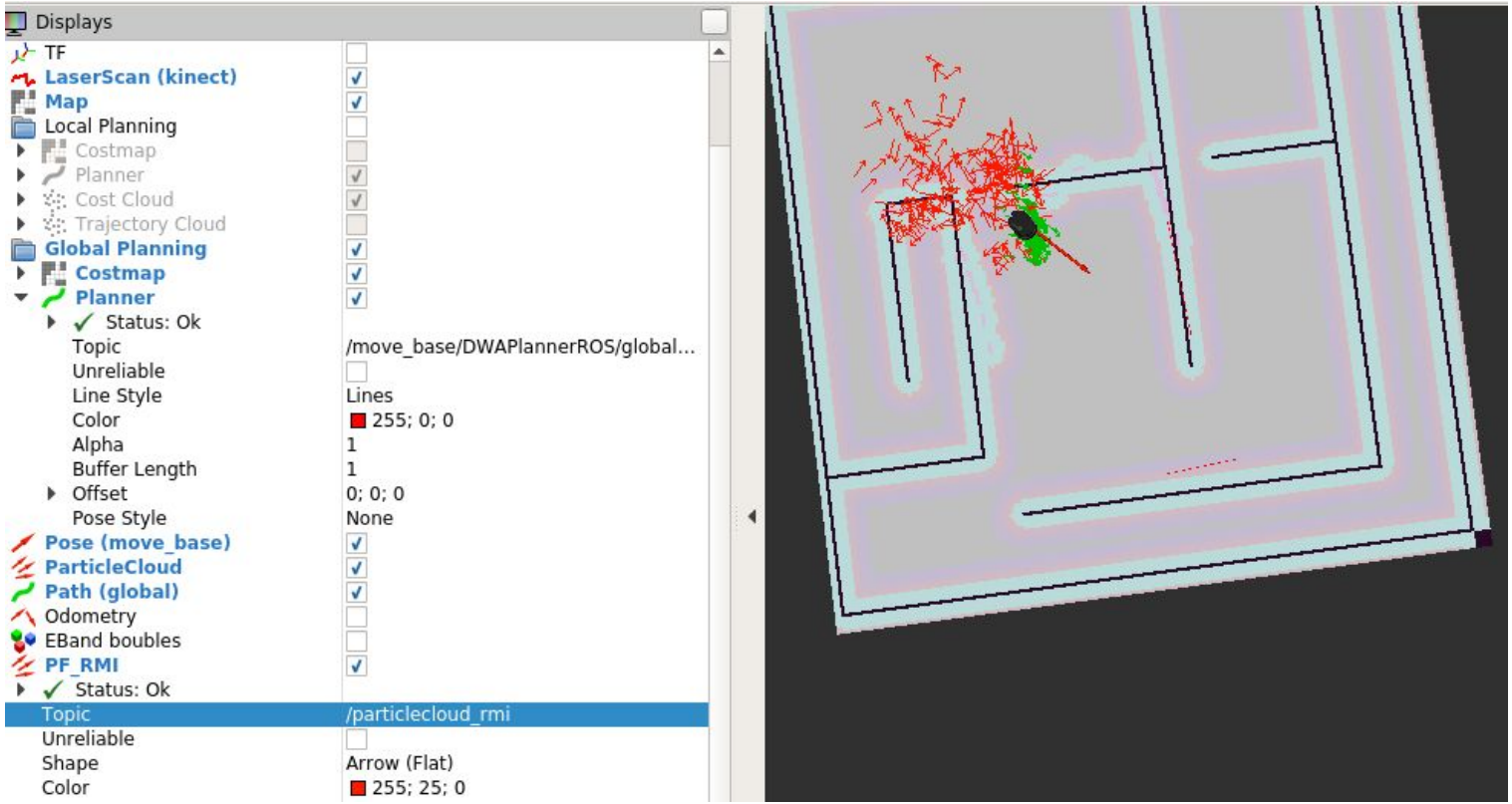
Filtro de Partículas - Código

```
self.pose_listener = rospy.Subscriber("initialpose",
    PoseWithCovarianceStamped,
    self.update_initial_pose)
# rospy.Subscriber(self.odom_frame, Odometry, self.odometryCb)
self.laser_subscriber = rospy.Subscriber(self.scan_topic, LaserScan, self.scan_received)

self.particle_pub = rospy.Publisher("particlecloud_rmi", PoseArray, queue_size=10)
self.marker_pub = rospy.Publisher("arraymarker", MarkerArray, queue_size=10)
# enable listening for and broadcasting coordinate transforms
self.tf_listener = TransformListener()
self.tf_broadcaster = TransformBroadcaster()
```

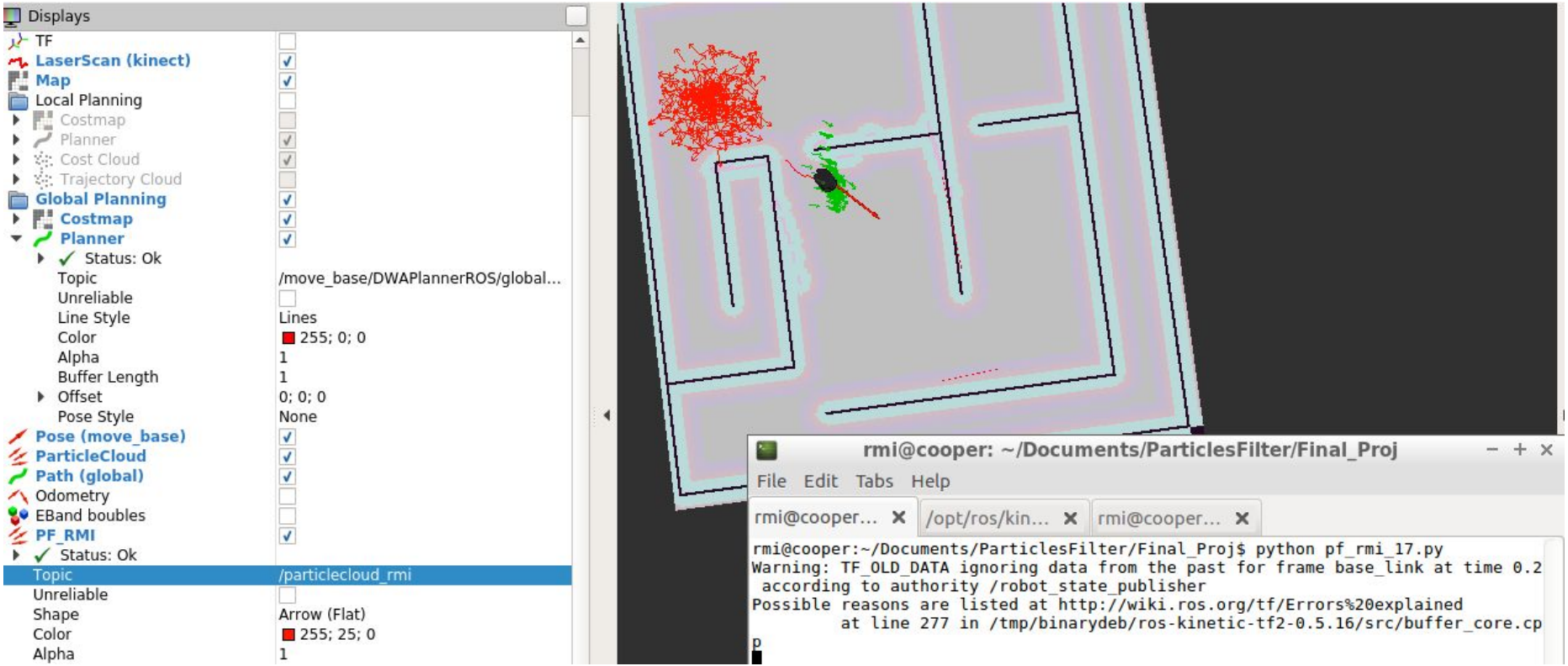
Filtro de Partículas - Resultado



Filtro de Partículas - Resultado



Filtro de Partículas - Problema



CONCLUSÃO E PROJETOS FUTUROS

Novo Dataset

Arquitetura mais simples

Filtro de partículas mais otimizado
e com um resample mais eficiente

DIFICULDADES

Implementação FCN.TensorFlow

Implementação Filtro de Partículas

- Ruído ODOM
- Resample