Tesi di laurea Magistrale in Bioingegneria

Nicolò Genesio

Genova 30/10/2015

ISTITUTO ITALIANO DI TECNOLOGIA



# Mobility map computation for autonomous navigation using RGBD sensors

**RELATORE**: PROF. FABIO SOLARI
**CORRELATORI**: DOTT. MANUELA CHESSA
DOTT. LORENZO NATALE
DOTT. TARIQ ABUHASHIM

# Introduction



- Thesis' goal

# Introduction

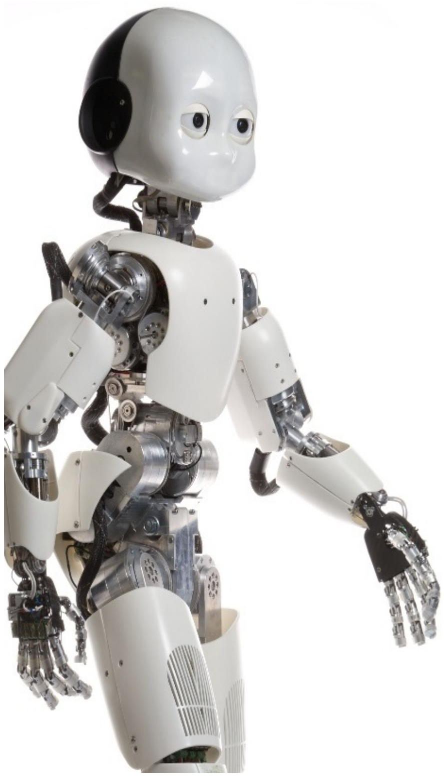


- Thesis' goal
  - Problem definition

# Introduction



- Thesis' goal
  - Problem definition



# Introduction

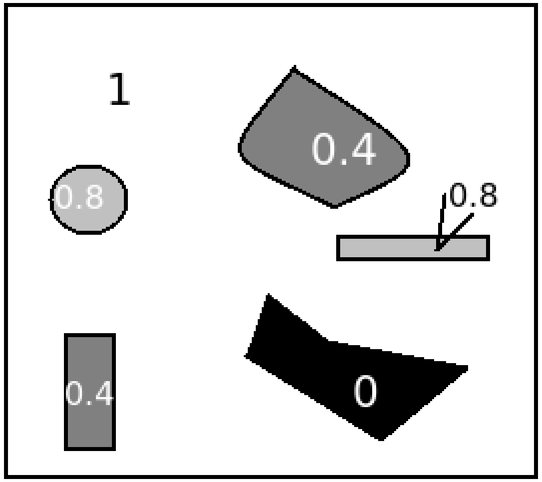


- Thesis' goal
  - Problem definition
  - What is a mobility map?

# Introduction



- Thesis' goal
  - Problem definition
  - What is a mobility map?



# Introduction



- Thesis' goal
  - Problem definition
  - What is a mobility map?
- System Setup

# Introduction



- Thesis’ goal
  - Problem definition
  - What is a mobility map?
- System Setup
  - Hardware

# Introduction



- Thesis' goal
  - Problem definition
  - What is a mobility map?
- System Setup
  - Hardware


# Introduction



- Thesis' goal
  - Problem definition
  - What is a mobility map?
- System Setup
  - Hardware
  - Software

# Introduction



- Thesis' goal
  - Problem definition
  - What is a mobility map?
- System Setup
  - Hardware
  - Software

# Biomedical motivations



- A 3D sensor within you

# Biomedical motivations

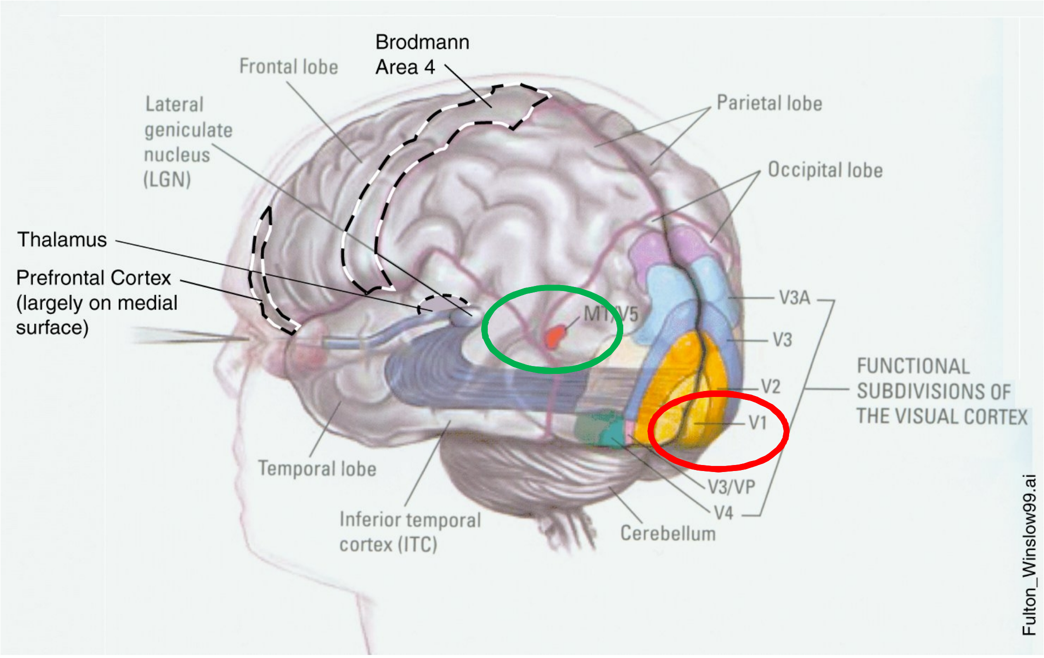


- A 3D sensor within you
  - Structure-from-motion(SFM) perception oriented to 3D appereance recovery→MT & V1 areas. [Buracas and Albright, *Vision Research* 1996]

# Biomedical motivations



- A 3D sensor within you
  - Structure-from-motion(SFM) perception oriented to 3D appereance recovery→MT & V1 areas. [Buracas and Albright, *Vision Research* 1996]



# Biomedical motivations



- A 3D sensor within you
  - Structure-from-motion(SFM) perception oriented to 3D appereance recovery→MT & V1 areas. [Buracas and Albright, *Vision Research* 1996]
- Biomedical applications

# Biomedical motivations



- A 3D sensor within you
  - Structure-from-motion(SFM) perception oriented to 3D appereance recovery→MT & V1 areas. [Buracas and Albright, *Vision Research* 1996]
- Biomedical applications
  - Wheelchair users assistance

# Biomedical motivations



- A 3D sensor within you
  - Structure-from-motion(SFM) perception oriented to 3D appereance recovery→MT & V1 areas. [Buracas and Albright, *Vision Research* 1996]
- Biomedical applications
  - Wheelchair users assistance
    - Impedance control [Kitagawa et al., *Intelligent Robots and Systems*, 2001]
    - Robotic wheelchair [Murarka et al., *Computer and Robot Vision*, 2006]

# Biomedical motivations



# 3D sensing



- Point Cloud

# 3D sensing



- Point Cloud
  - Irregular sampling
  - Implicit function not present
  - Range-dependent noise

# 3D sensing



- Sensors

# 3D sensing



- Sensors
  - LiDAR( *Light/Laser Detection and Ranging* ) sensors

# 3D sensing



- Sensors
  - LiDAR( *Light/Laser Detection and Ranging* ) sensors
  - Stereo camera systems

# 3D sensing

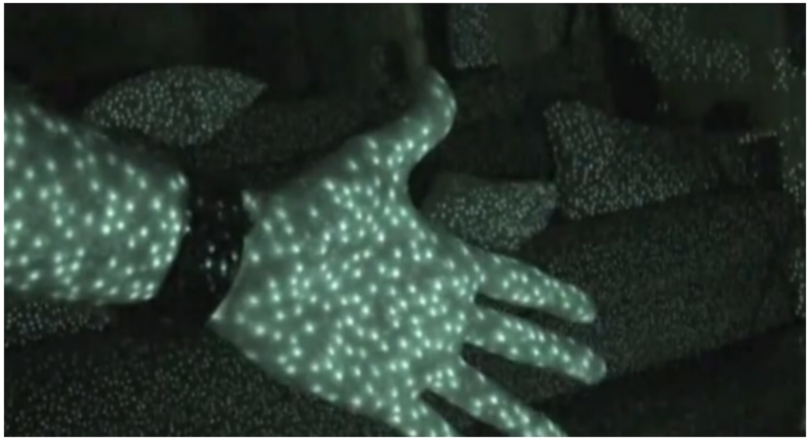


- Sensors
  - LiDAR( *Light/Laser Detection and Ranging* ) sensors
  - Stereo camera systems
  - RGBD sensors based on structured light

# 3D sensing

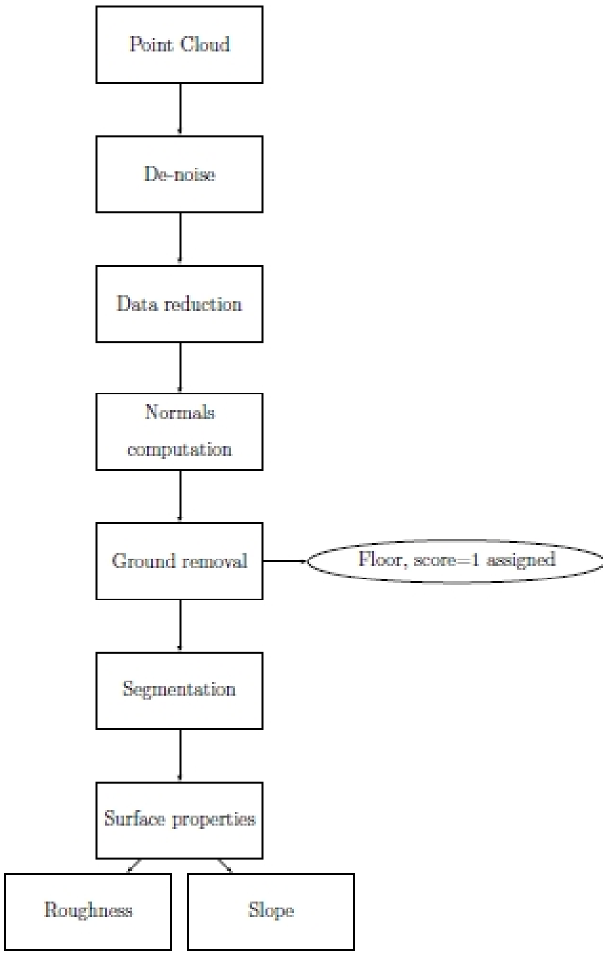


- Sensors
  - LiDAR( *Light/Laser Detection and Ranging* ) sensors
  - Stereo camera systems
  - RGBD sensors based on structured light



# Algorithm overview

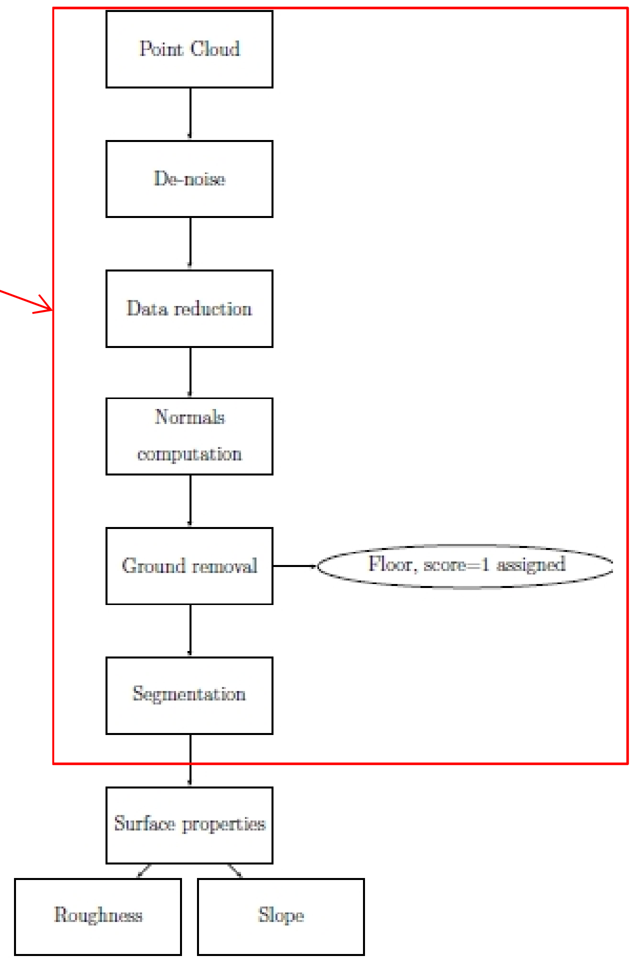




# Algorithm overview



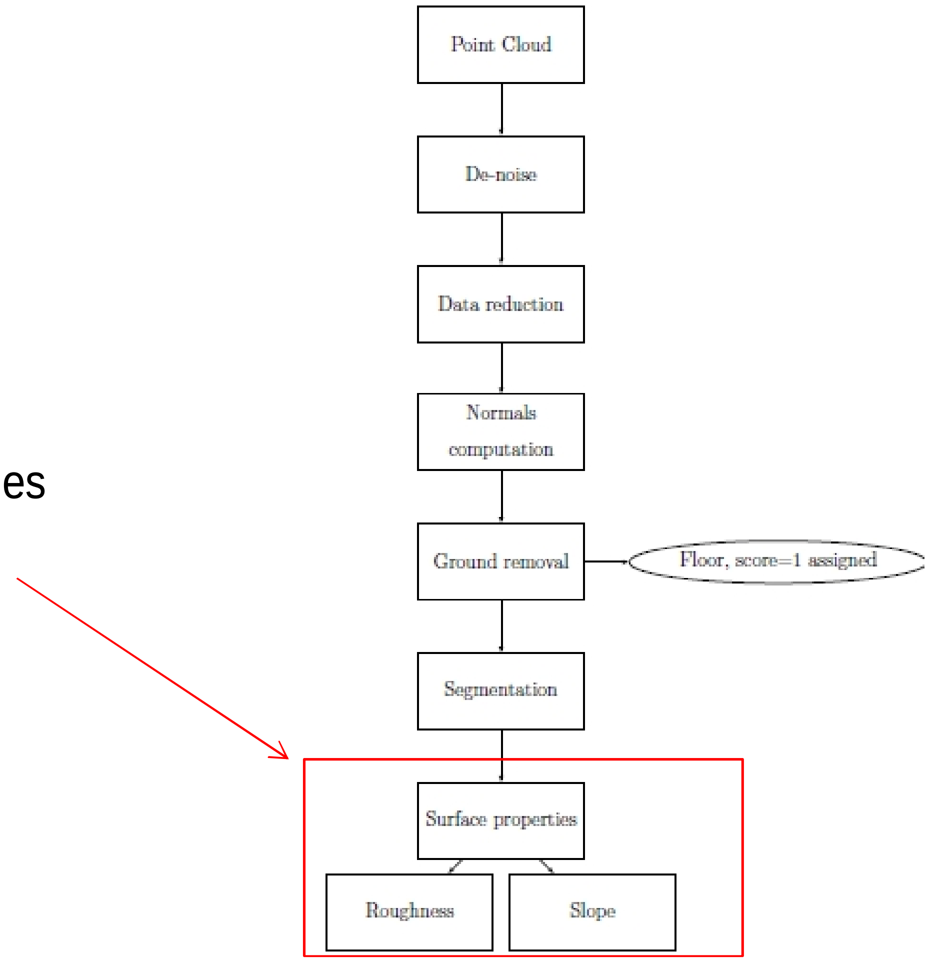
Segmentation algorithm



# Algorithm overview

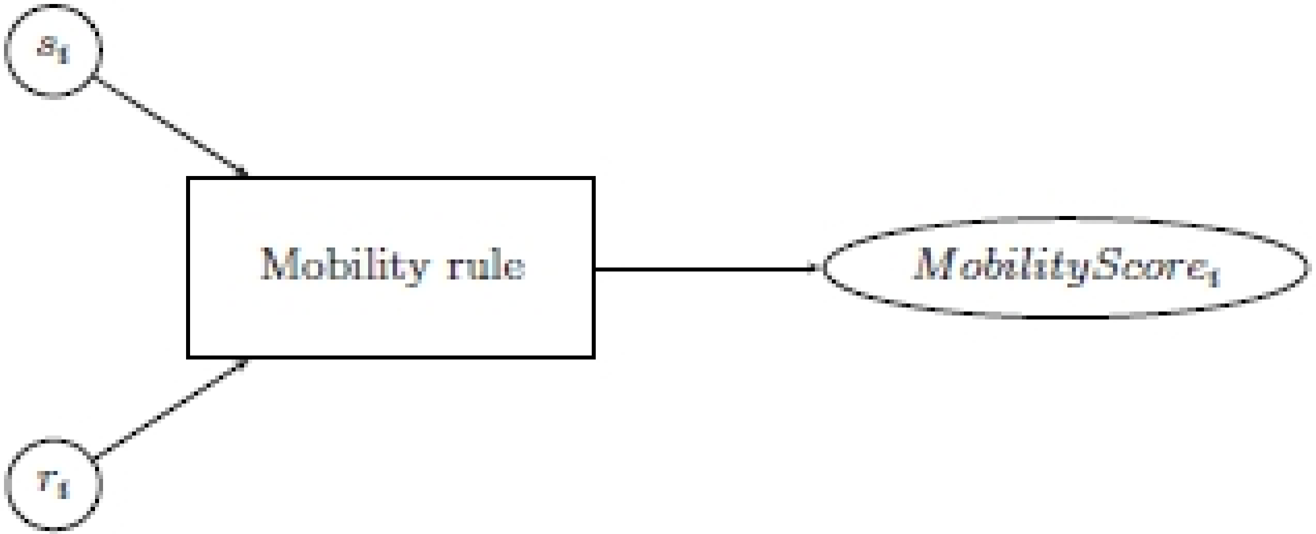


Surface properties estimation



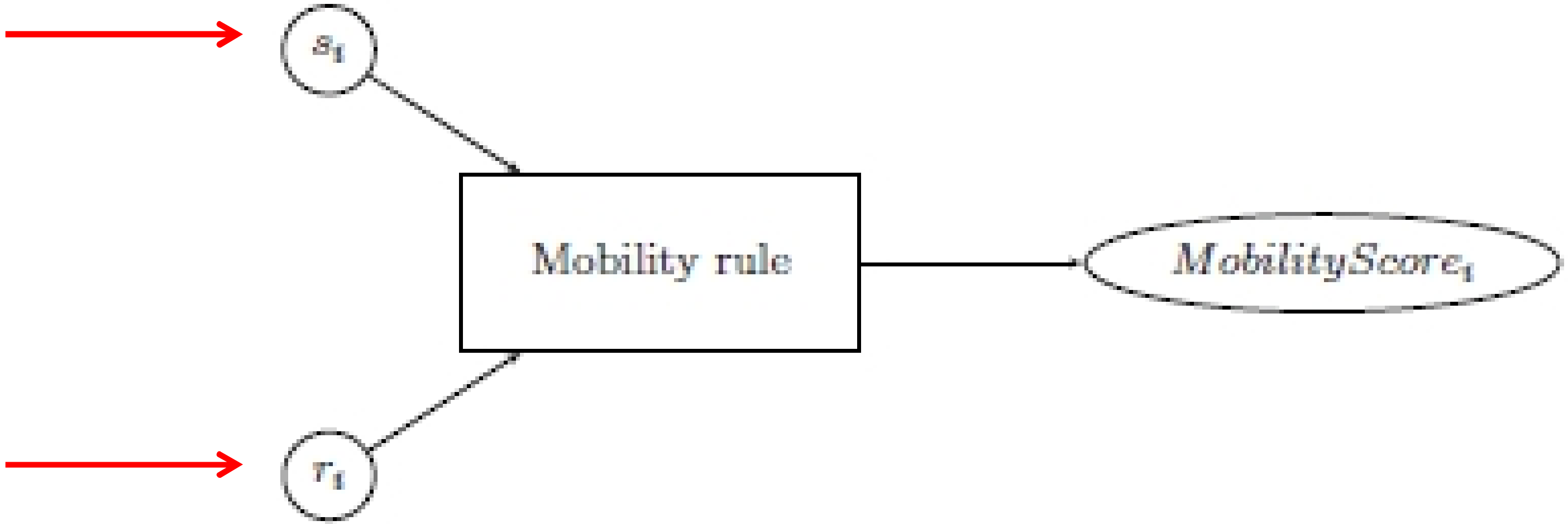
# Algorithm overview





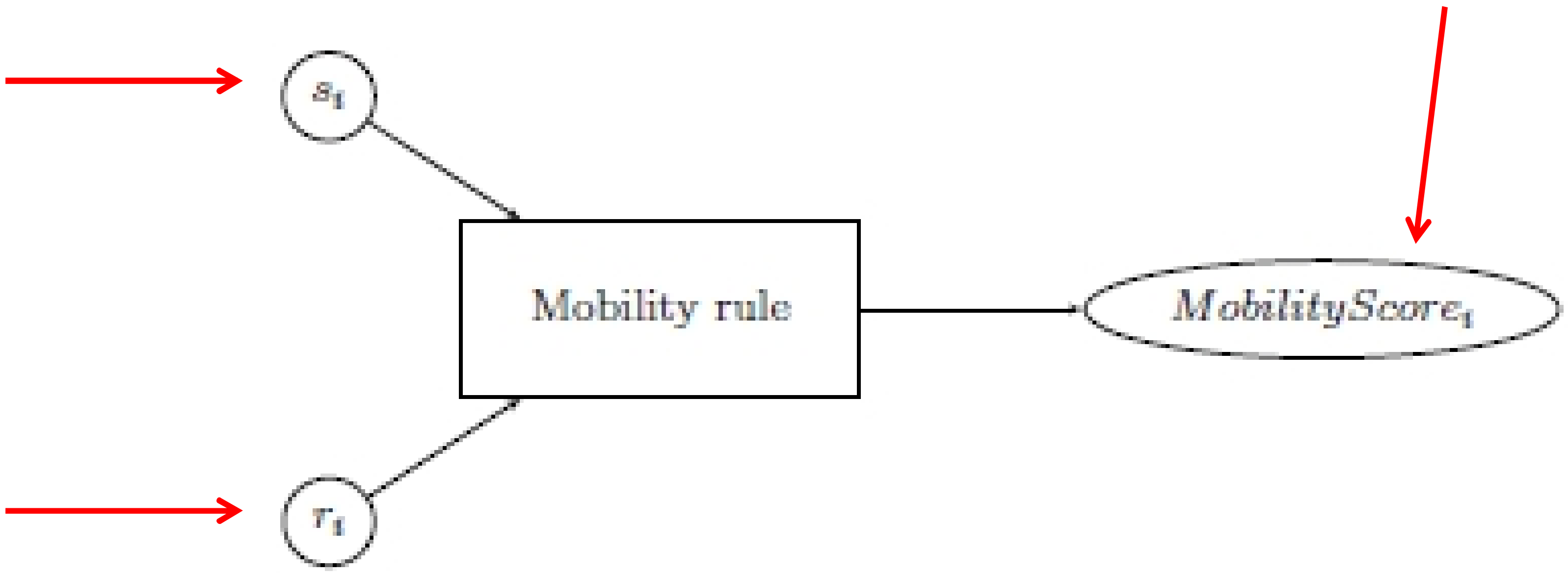
# Algorithm overview





# Algorithm overview





# Segmentation algorithm



- De-noise

# Segmentation algorithm



- De-noise
  - Statistical outlier removal:

$$P^* = \{p_q^* \in P | (\mu_k - \alpha\sigma_k) \leq \bar{d} \leq (\mu_k + \alpha\sigma_k)\}$$

# Segmentation algorithm



- De-noise
  - Statistical outlier removal:

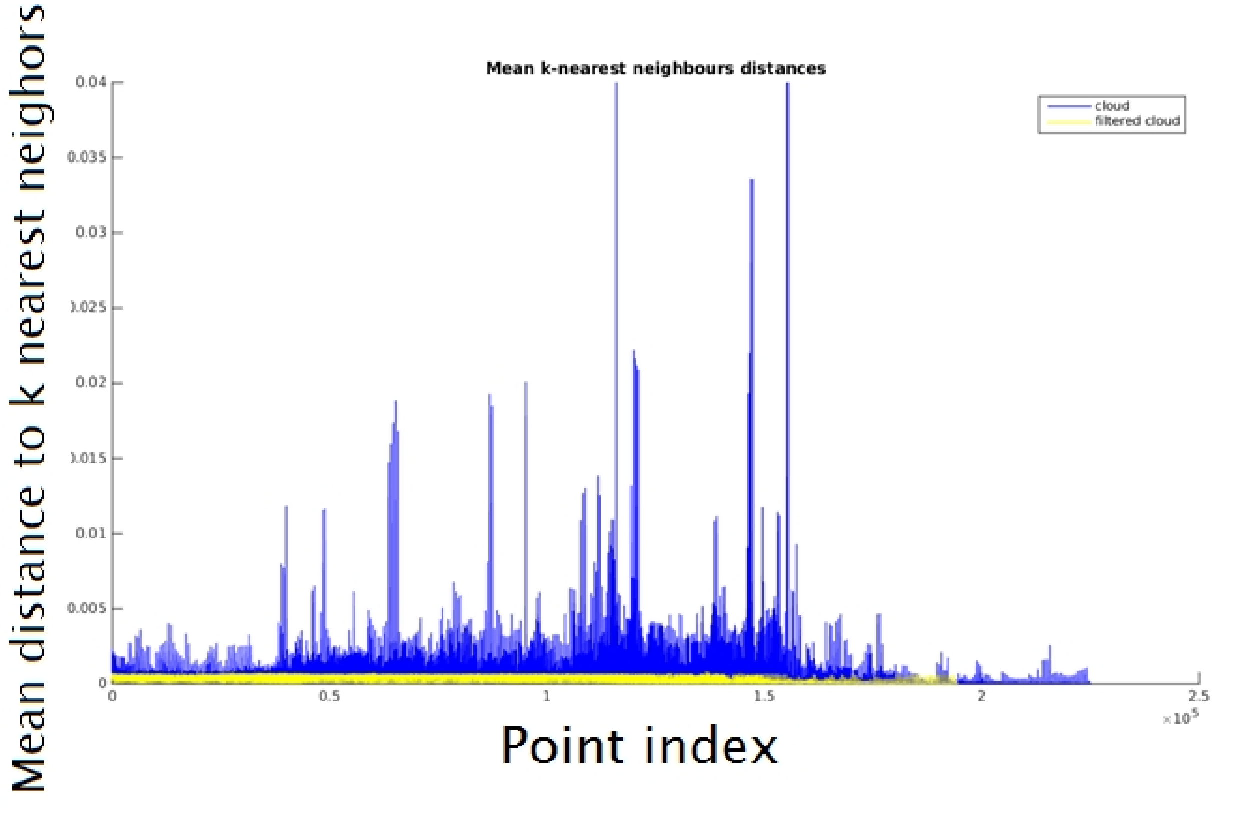
$$P^* = \{p_q^* \in P | (\mu_k - \alpha\sigma_k) \leq \bar{d} \leq (\mu_k + \alpha\sigma_k)\}$$

For each point $\mathbf{p_q}$ belonging to the cloud **P**, the mean distance $\bar{d}$ to its **k** closest neighbors is first computed.
Then, a distribution over the mean distance $\bar{d}$ space for the entire point cloud is assembled and its mean $\boldsymbol{\mu}_{\mathbf{k}}$ and standard deviation $\boldsymbol{\sigma}_{\mathbf{k}}$ are estimated**.**
The goal is to keep the points whose mean distance $\bar{d}$ to the closest **k** neighbors is similar to the one for the rest of the points.

# Segmentation algorithm



- De-noise
  - Statistical outlier removal:



# Segmentation algorithm



- De-noise
  - Statistical outlier removal
- Data reduction

# Segmentation algorithm

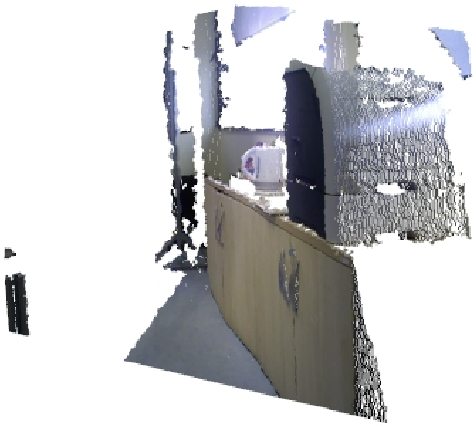


- De-noise
  - Statistical outlier removal
- Data reduction
  - Rejection of points farther than 1.5 mt from the sensor

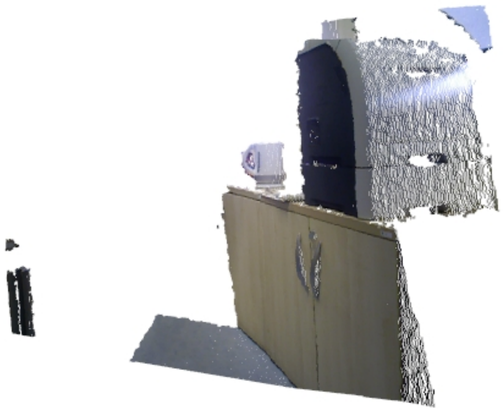
# Segmentation algorithm



- De-noise
  - Statistical outlier removal
- Data reduction
  - Rejection of points farther than 1.5 mt from the sensor



Before



After

# Segmentation algorithm

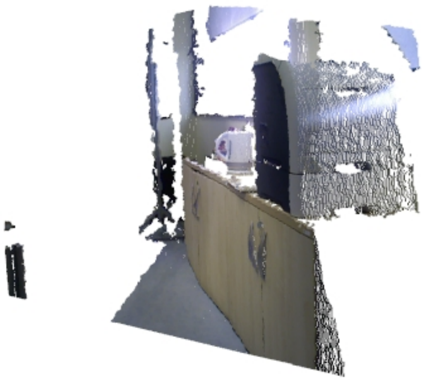


- De-noise
  - Statistical outlier removal
- Data reduction
  - Rejection of points farther than 1.5 mt from the sensor
  - Downsampling

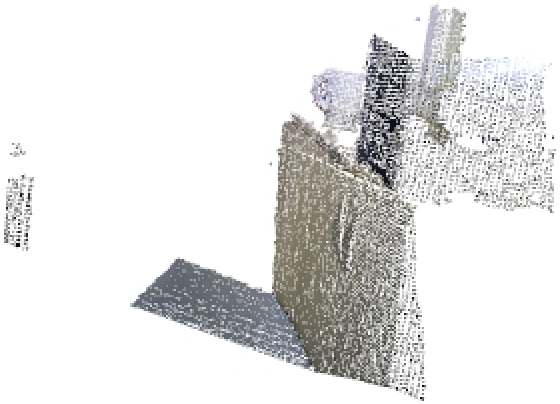
# Segmentation algorithm



- De-noise
  - Statistical outlier removal
- Data reduction
  - Rejection of points farther than 1.5 mt from the sensor
  - Downsampling



Before



After

# Segmentation algorithm



- De-noise
  - Statistical outlier removal
- Data reduction
  - Rejection of points farther than 1.5 mt from the sensor
  - Downsampling

| Reduction | Number of points | % of reduction |
|---|---|---|
| None | 307200 | – |
| Far points rejected | 181091 | 42% |
| Downsampling | 25706 | 91.7% |

# Segmentation algorithm



- Normal computation through PCA technique

# Segmentation algorithm



- Normal computation through PCA technique

$$C = \frac{1}{k}\sum_{i=1}^{k}(p_i - \bar{p}) \cdot (p_i - \bar{p})^T$$

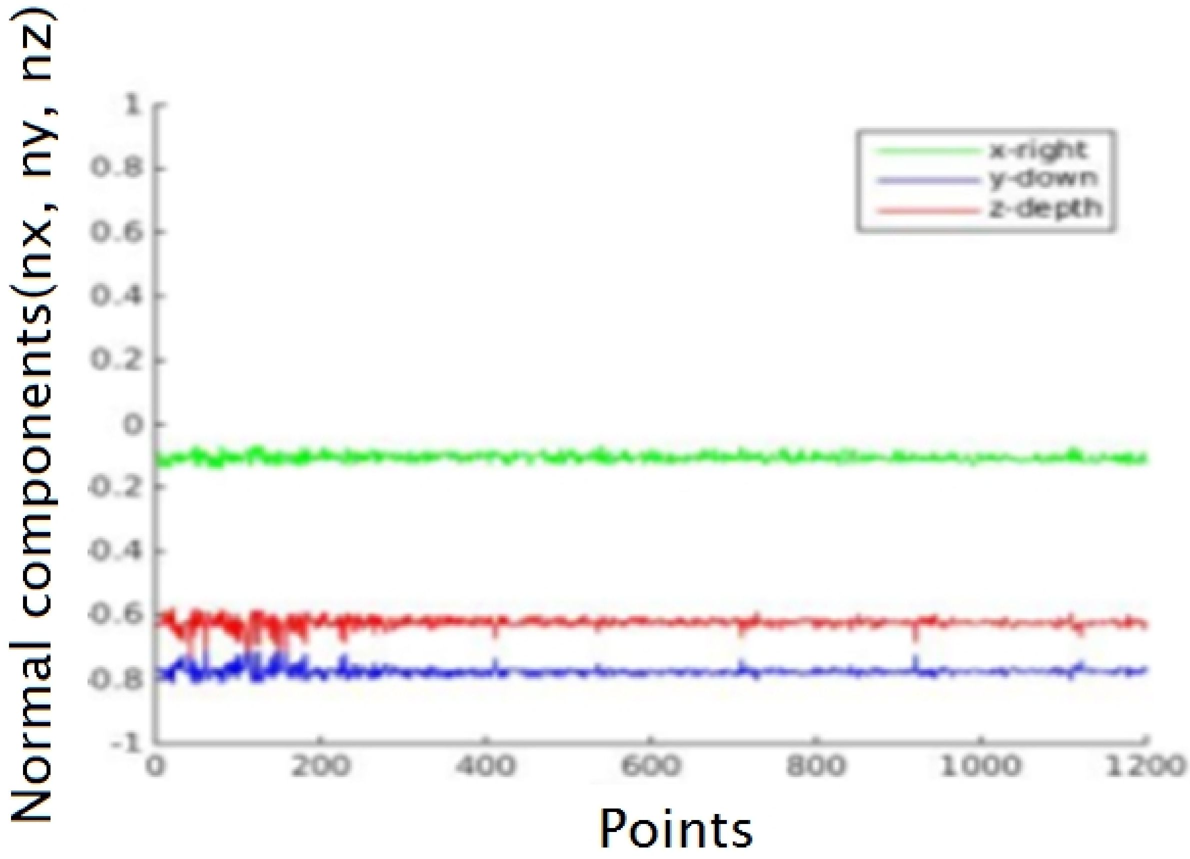
where

$$\bar{p} = \frac{1}{k}\sum_{i=1}^{k} p_i$$

# Segmentation algorithm



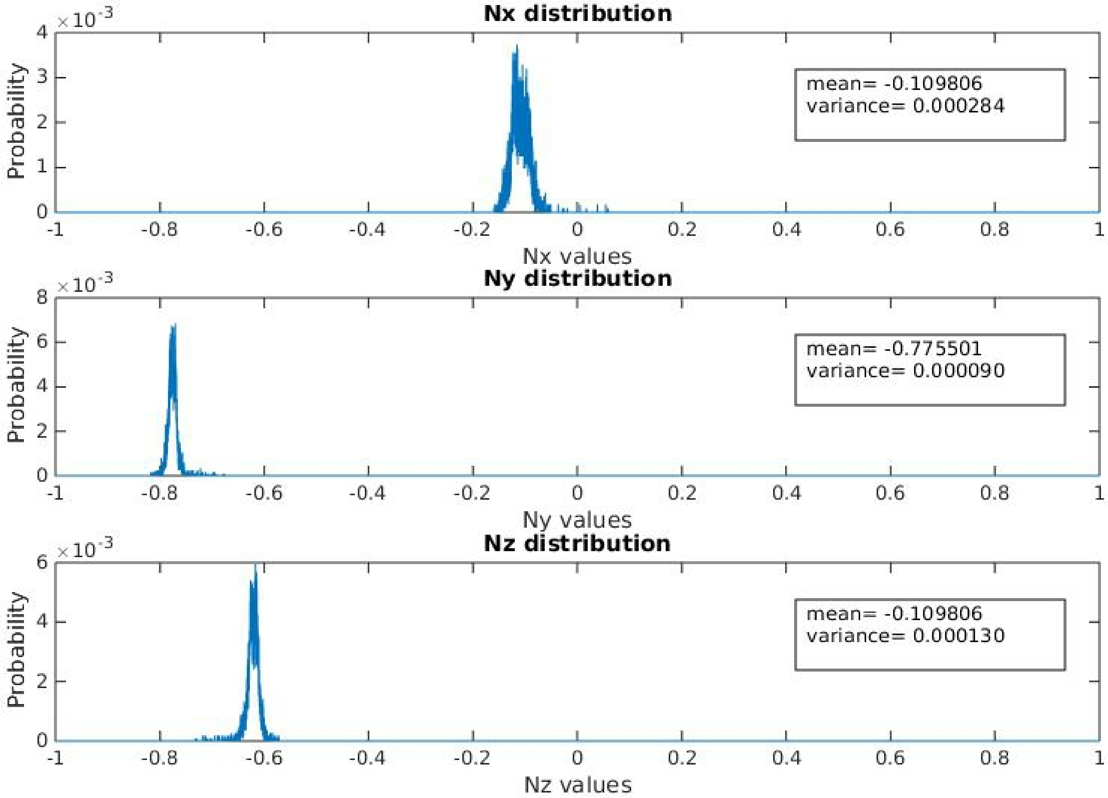
- Normal computation through PCA technique



# Segmentation algorithm



- Normal computation through PCA technique



# Segmentation algorithm



- Normal computation through PCA technique

| Cloud | Number of points | Computational time |
|---|---|---|
| Original | 307200 | 203634 ms |
| Downsampled | 7262 | 101 ms |

# Segmentation algorithm



- Normal computation through PCA technique
- Ground Remouval

# Segmentation algorithm

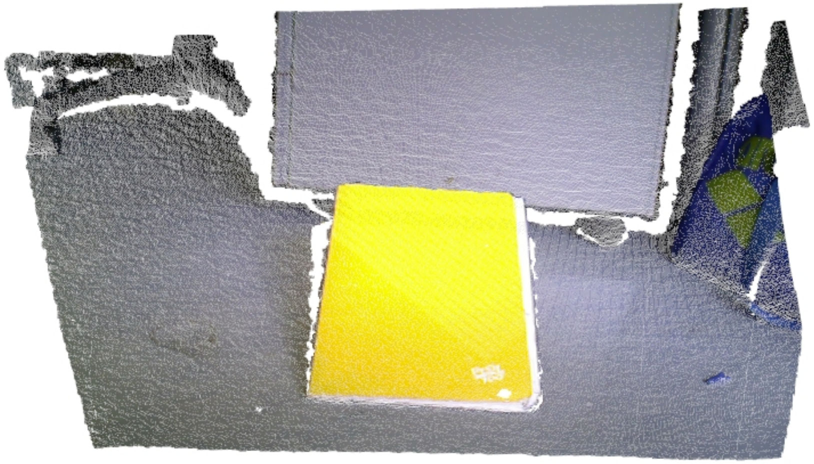


- Normal computation through PCA technique
- Ground Removual
  - RANSAC(Random Sample Consensous) –MLS(Mean Least Square) technique.

# Segmentation algorithm



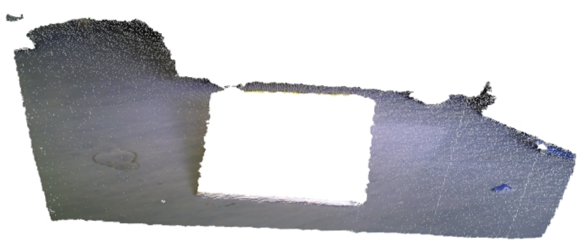
- Normal computation through PCA technique
- Ground Removal
  - RANSAC(Random Sample Consensous) –MLS(Mean Least Square) technique.

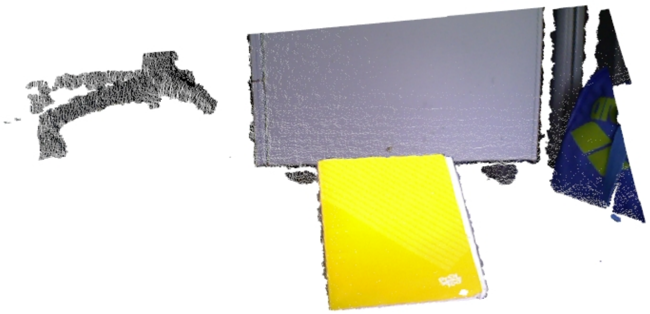


# Segmentation algorithm



- Normal computation through PCA technique
- Ground Remouval
  - RANSAC(Random Sample Consensous) –MLS(Mean Least Square) technique.





# Segmentation algorithm



- Segmentation via Region Growing

# Segmentation algorithm



- Segmentation via Region Growing
  - Distance constraint

# Segmentation algorithm



- Segmentation via Region Growing
  - Distance constraint
  - Angular difference between normals

# Segmentation algorithm



- Segmentation via Region Growing
  - Distance constraint
  - Angular difference between normals
  - Color  difference

# Segmentation algorithm

- Segmentation via Region Growing
  - Distance constraint
  - Angular difference between normals
  - Color difference

# Segmentation algorithm



- Segmentation via Region Growing
  - Distance constraint
  - Angular difference between normals
  - Color difference

$$R = \frac{255}{max_{nx} - min_{nx}} * (n_{ix} - min_{nx})$$

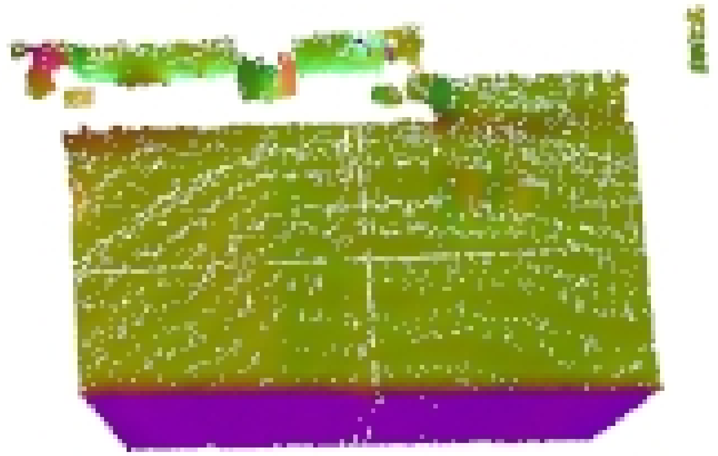
$$G = \frac{255}{max_{ny} - min_{ny}} * (n_{iy} - min_{ny})$$

$$B = \frac{255}{max_{nz} - min_{nz}} * (n_{iz} - min_{nz})$$

# Segmentation algorithm



- Segmentation via Region Growing
  - Distance constraint
  - Angular difference between normals
  - Color difference



# Segmentation algorithm

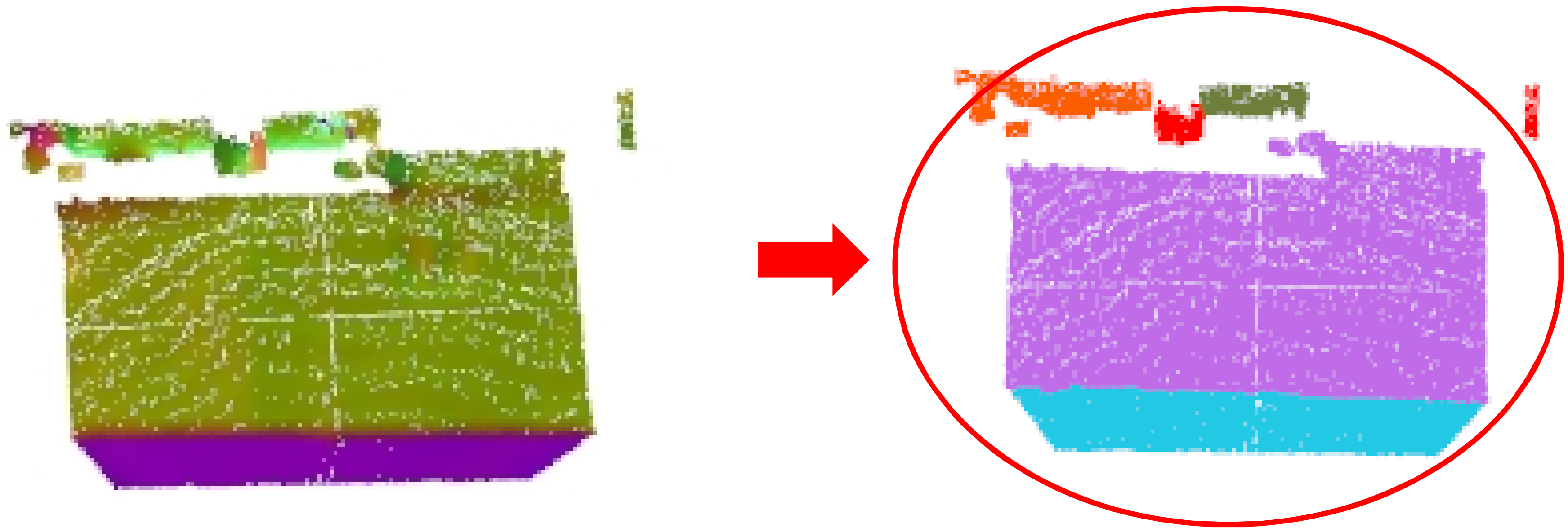


- Segmentation via Region Growing
  - Distance constraint
  - Angular difference between normals
  - Color difference

# Segmentation algorithm

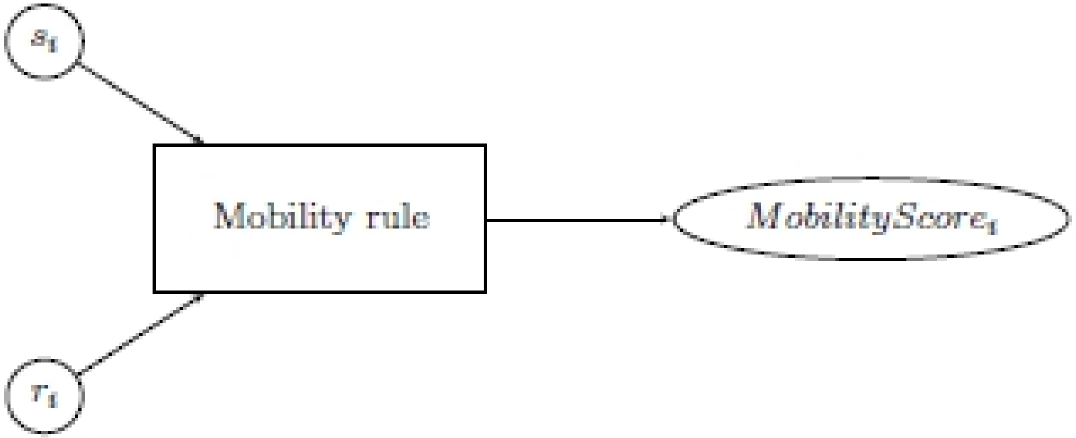


- Segmentation via Region Growing
  - Distance constraint
  - Angular difference between normals
  - Color difference



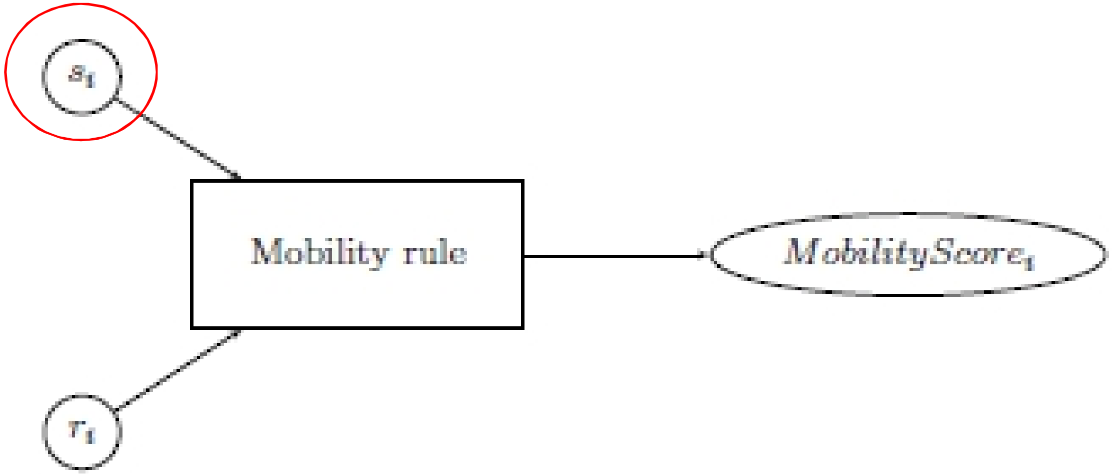
# Building the Mobility map





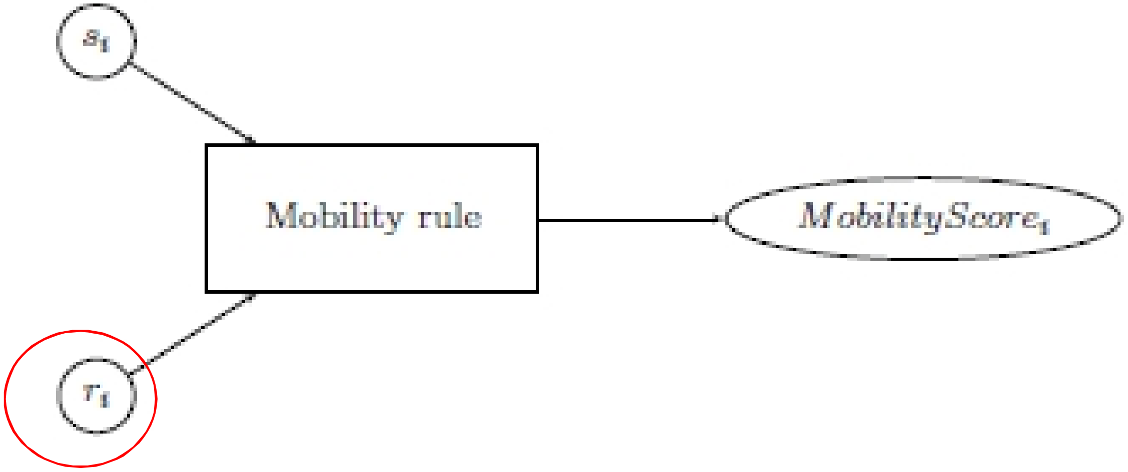
# Building the Mobility map





# Building the Mobility map





# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique

# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation

# Building the Mobility map

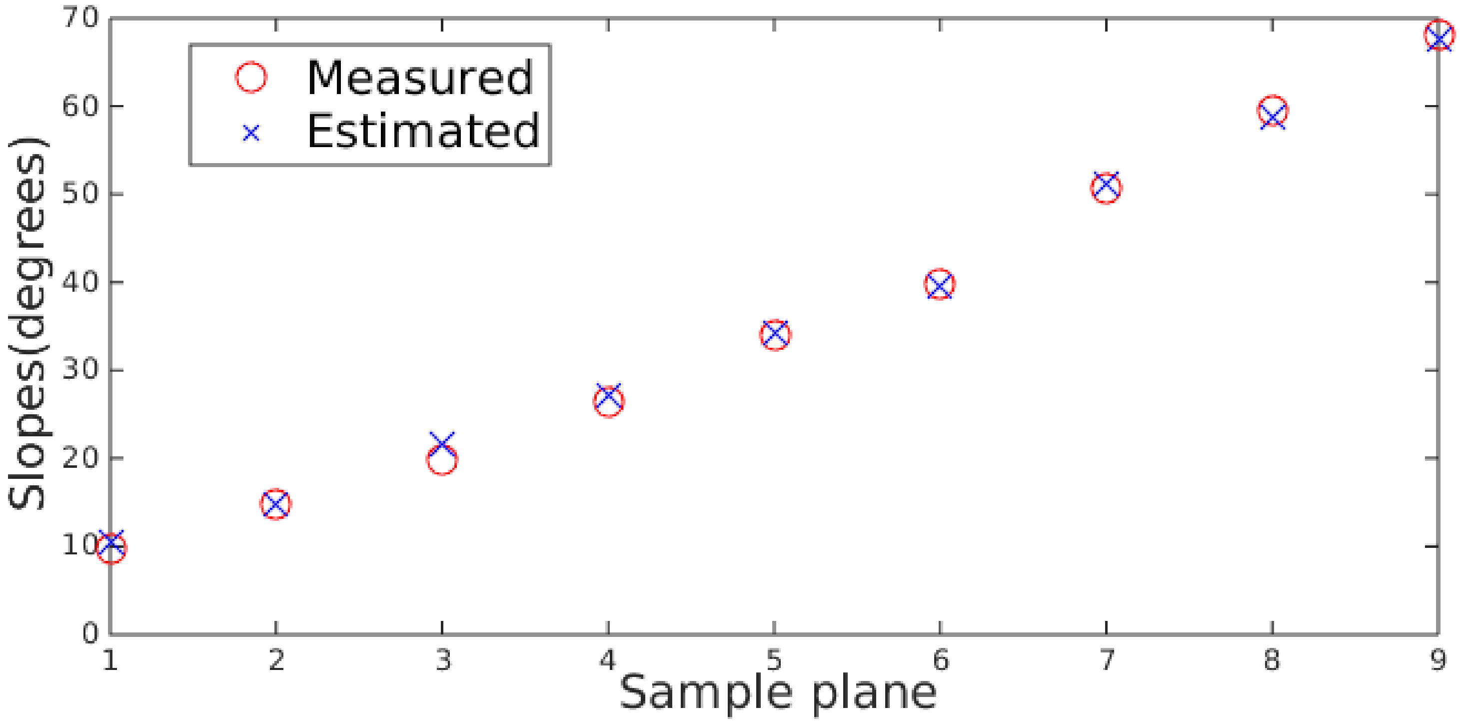


- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation

$$\theta = arcos(\frac{\vec{n_i} \cdot \vec{n_f}}{||\vec{n_i}|| \cdot ||\vec{n_f}||}) * \frac{180}{\pi}$$

# Building the Mobility map





# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation
- Roughness Estimation

# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation
- Roughness Estimation

$$r = \frac{A}{A'}$$

# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation
- Roughness Estimation

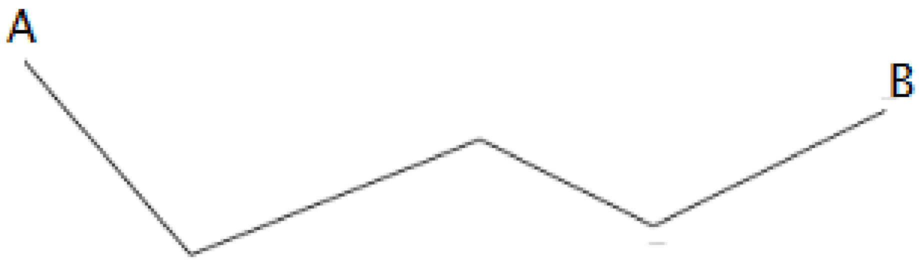
$$r = \frac{A}{A'}$$

r=1 ->smooth
r>>1->rough
r<1->impossible

# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation
- Roughness Estimation

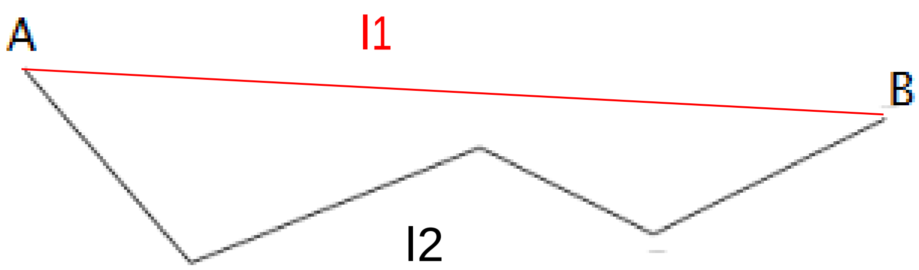



r=1 ->smooth
r>>1->rough
r<1->impossible

# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation
- Roughness Estimation

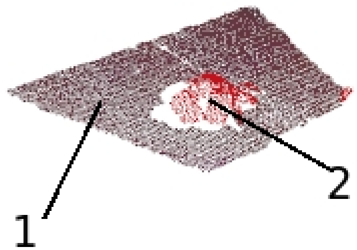



r=1 ->smooth
r>>1->rough
r<1->impossible

r= l2/l1

# Building the Mobility map

- Roughness estimation
  - Sample Cloud 1



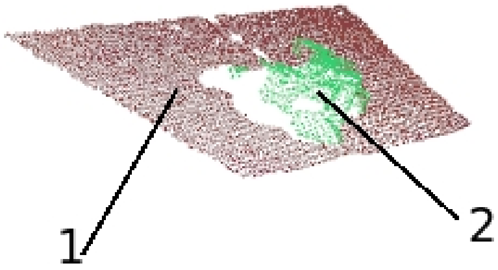

$r_1 = 1.03$

$r_2 = 1.582$

# Building the Mobility map



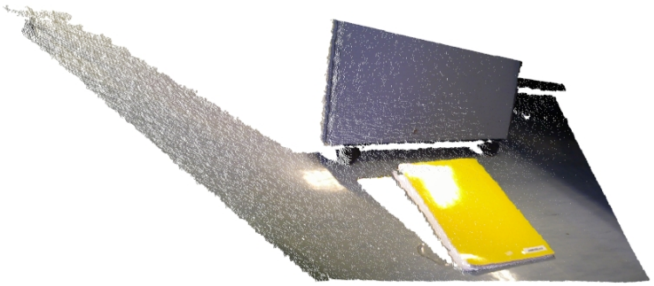
- Roughness estimation
  - Sample Cloud 2

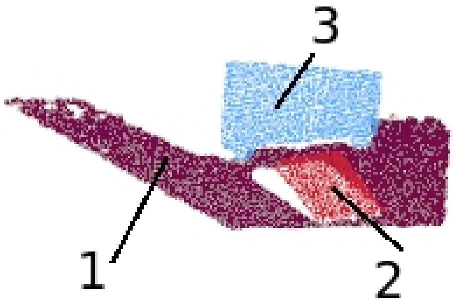



$r_1 = 1.04$

$r_2 = 1.70$

- Roughness estimation
  - Sample Cloud 3






$r_1 = 1.03$

$r_2 = 1.09$

$r_3 = 1.17$

# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation
- Roughness Estimation
- Mobility map computation

# Building the Mobility map



- Best Plane fitting using the RANSAC-MLS technique
- Slope estimation
- Roughness Estimation
- Mobility map computation
  - Five classes of traversability: 0, 0.25, 0.5, 0.75, 1

# Building the Mobility map

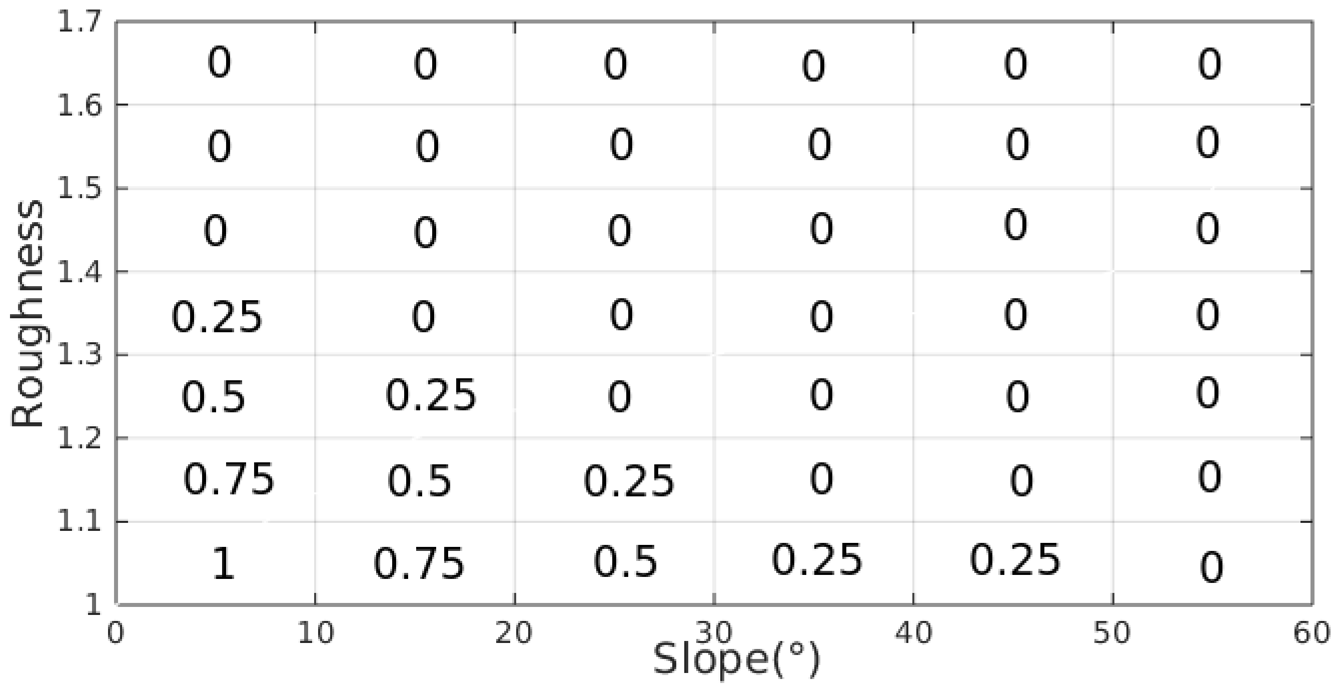


- Mobility map computation
  - Mobility rule

# Building the Mobility map



- Mobility map computation
  - Mobility rule



# Experimental evaluation



- 3 Workspaces analyzed: free corridor, corridor with obstacles, stairs.

# Experimental evaluation



- 3 Workspaces analyzed: free corridor, corridor with obstacles, stairs.

# Experimental evaluation

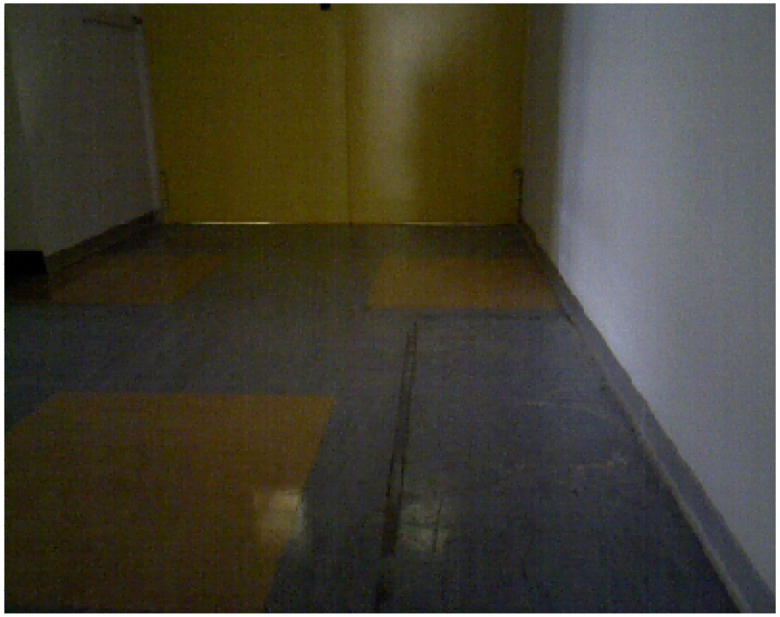


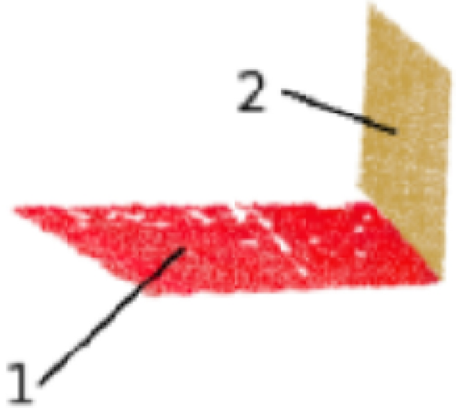
- Free corridor

# Experimental evaluation



- Free corridor



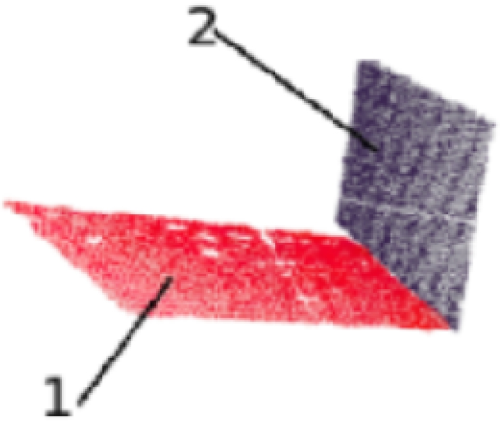



Slope_1=0.02° Roughness_1=1.11
Slope_2=89.4° Roughness_2=1.11

# Experimental evaluation



- Free corridor

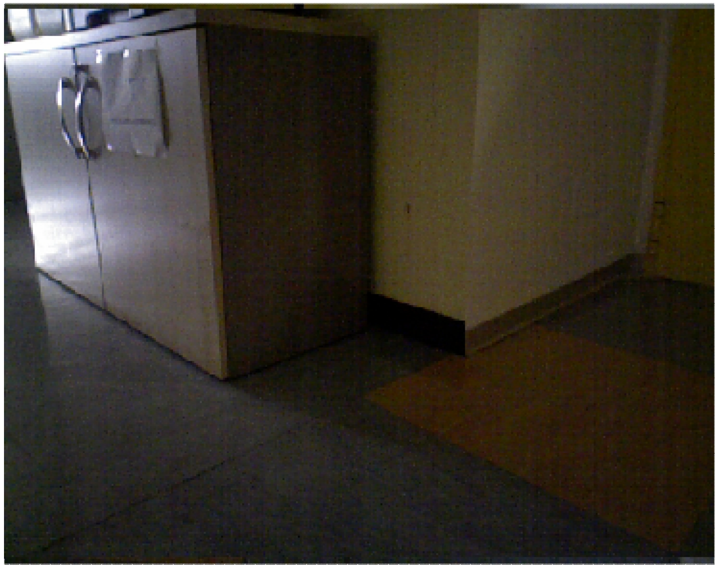


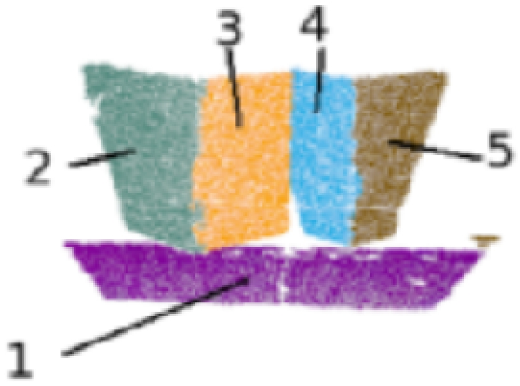

Slope_1=0°  Roughness_1=1.13
Slope_2=89.3°  Roughness_2=1.02

# Experimental evaluation



- Free corridor



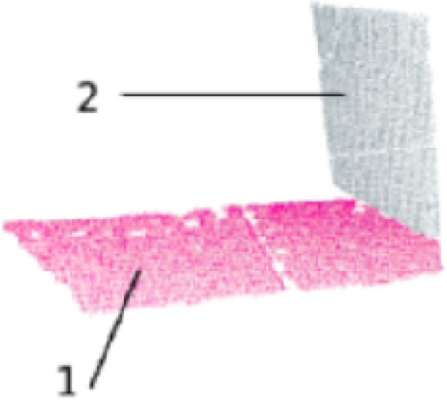



| | |
|---|---|
| Slope_1=0° | Roughness_1=1.11 |
| Slope_2=88.6° | Roughness_2=1.09 |
| Slope_3=89.2° | Roughness_3=1.13 |
| Slope_4=89.8° | Roughness_4=1.05 |
| Slope_5=89.9° | Roughness_5=1.01 |

# Experimental evaluation



- Free corridor

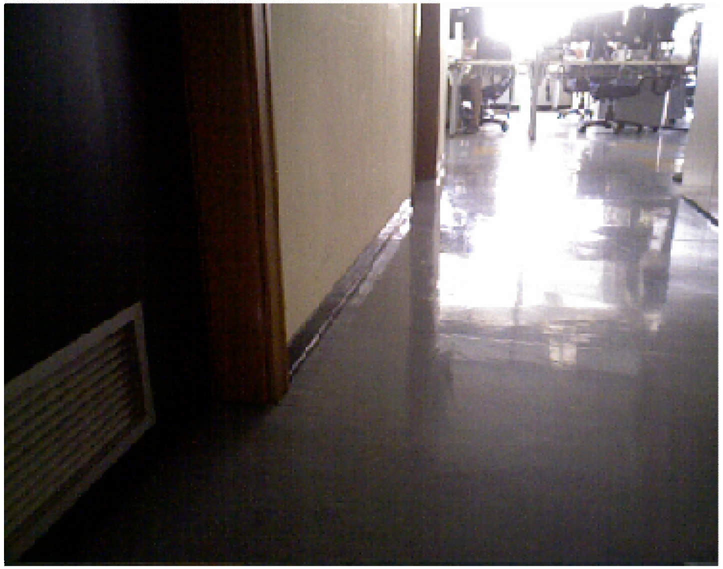


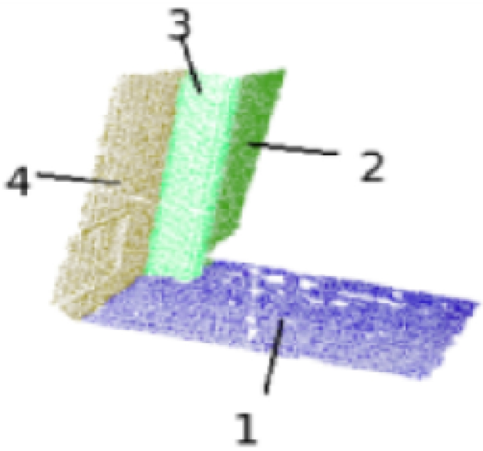

Slope_1=0°    Roughness_1=1.12
Slope_2=89.4°    Roughness_2=1.11

# Experimental evaluation



- Free corridor



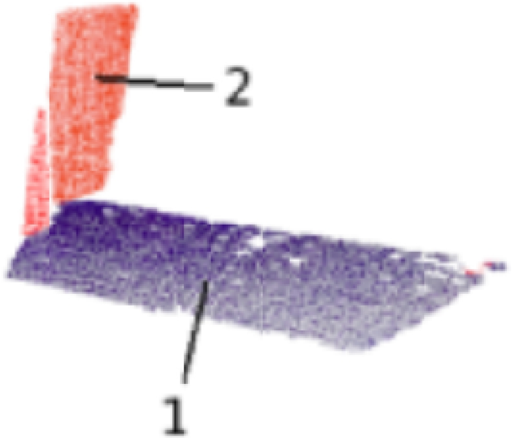



| | |
|---|---|
| Slope_1=0° | Roughness_1=1.07 |
| Slope_2=89.8° | Roughness_2=1.14 |
| Slope_3=87.9° | Roughness_3=1.38 |
| Slope_4=89.7° | Roughness_4=1.03 |

# Experimental evaluation



- Free corridor

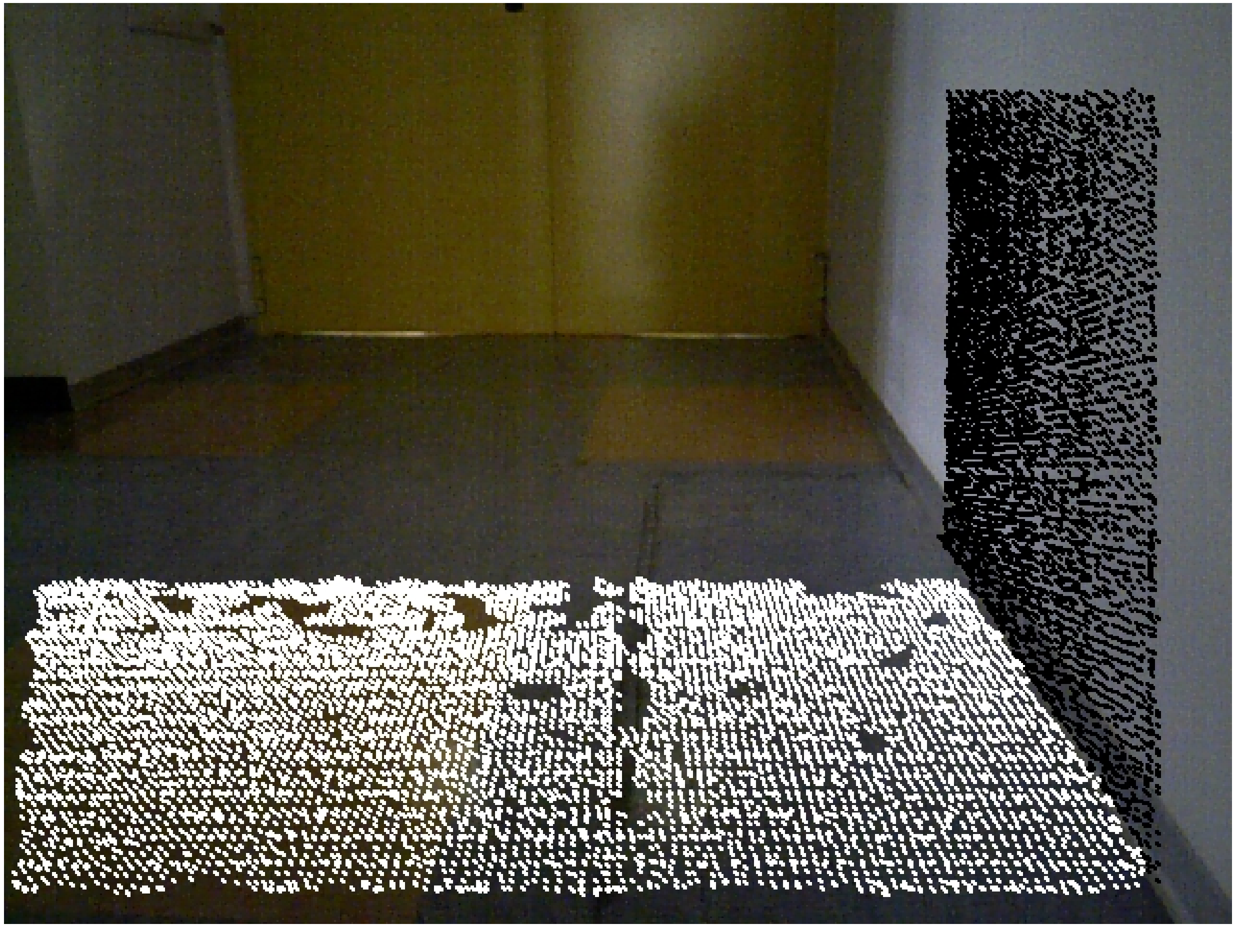



Slope_1=0°  Roughness_1=1.08
Slope_2=89.66°  Roughness_2=1.04

# Experimental evaluation



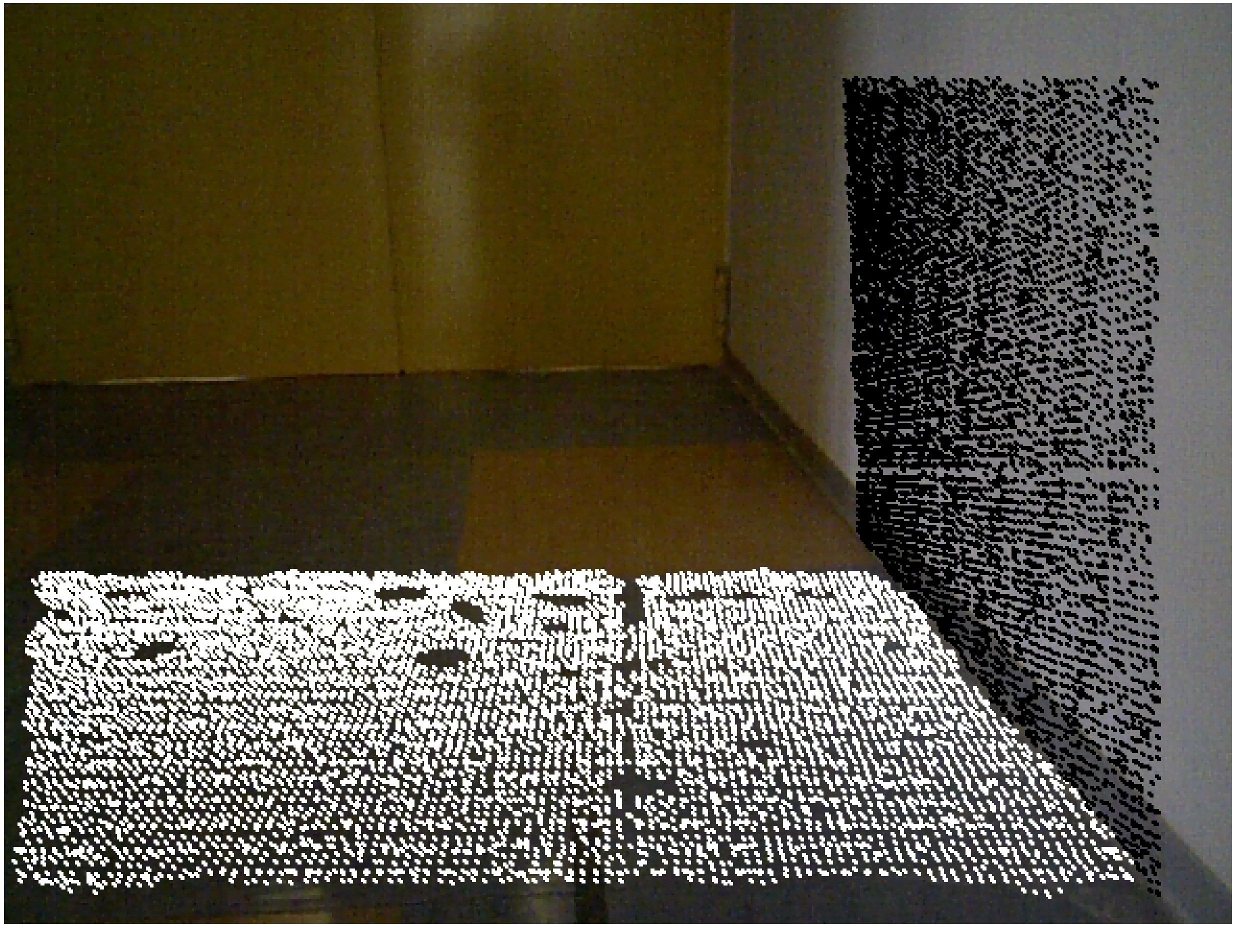
- Free corridor



# Experimental evaluation



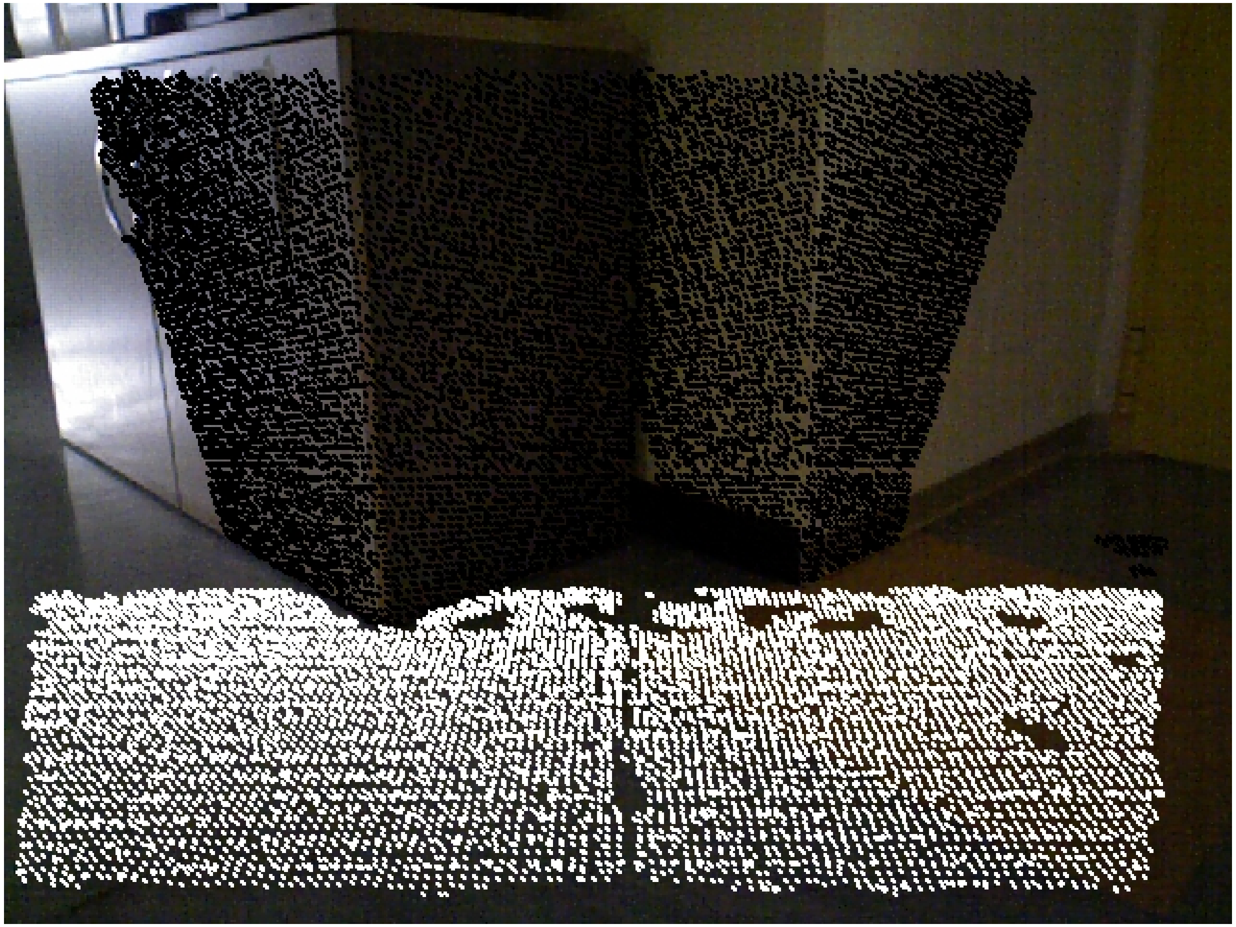
- Free corridor



# Experimental evaluation



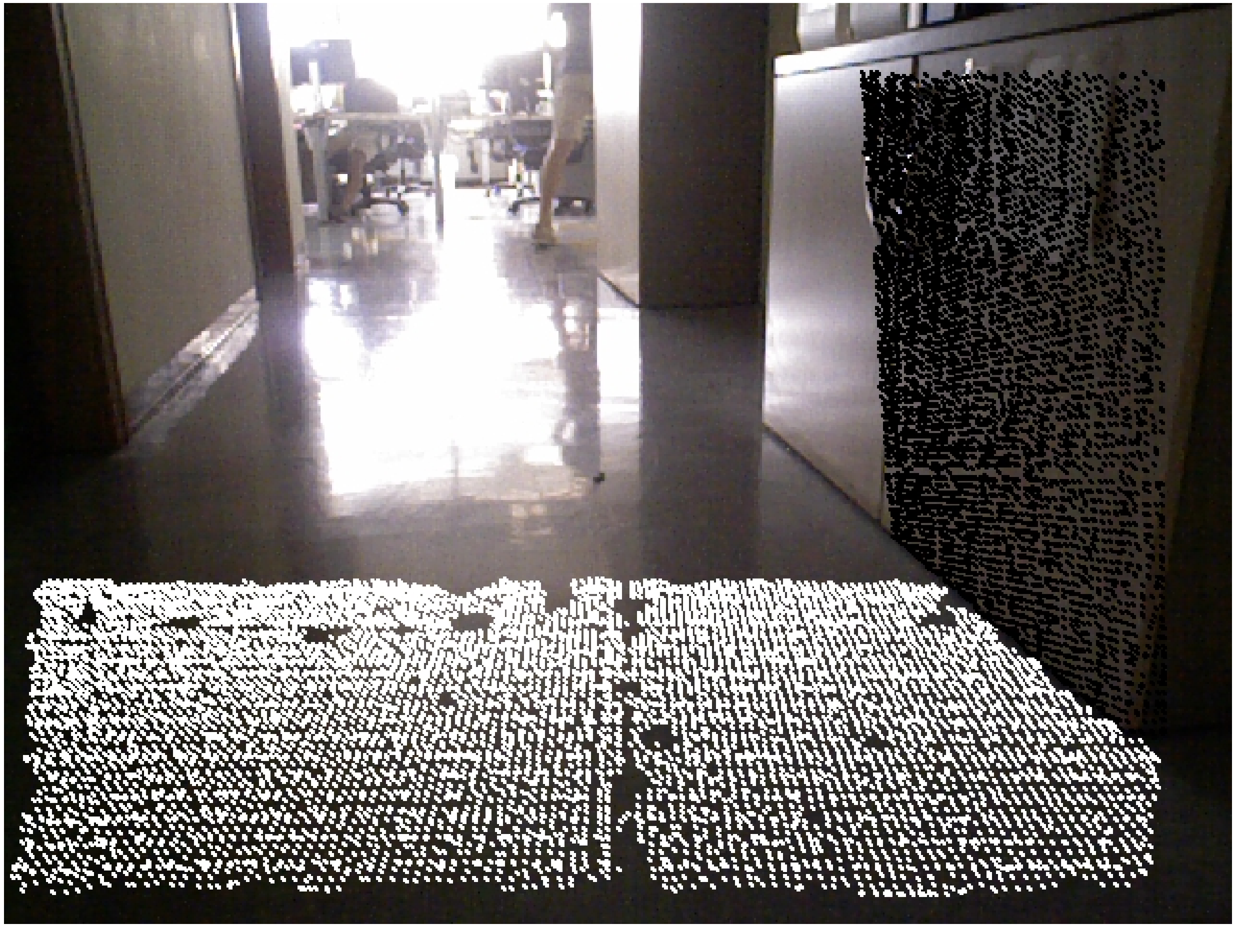
- Free corridor



# Experimental evaluation



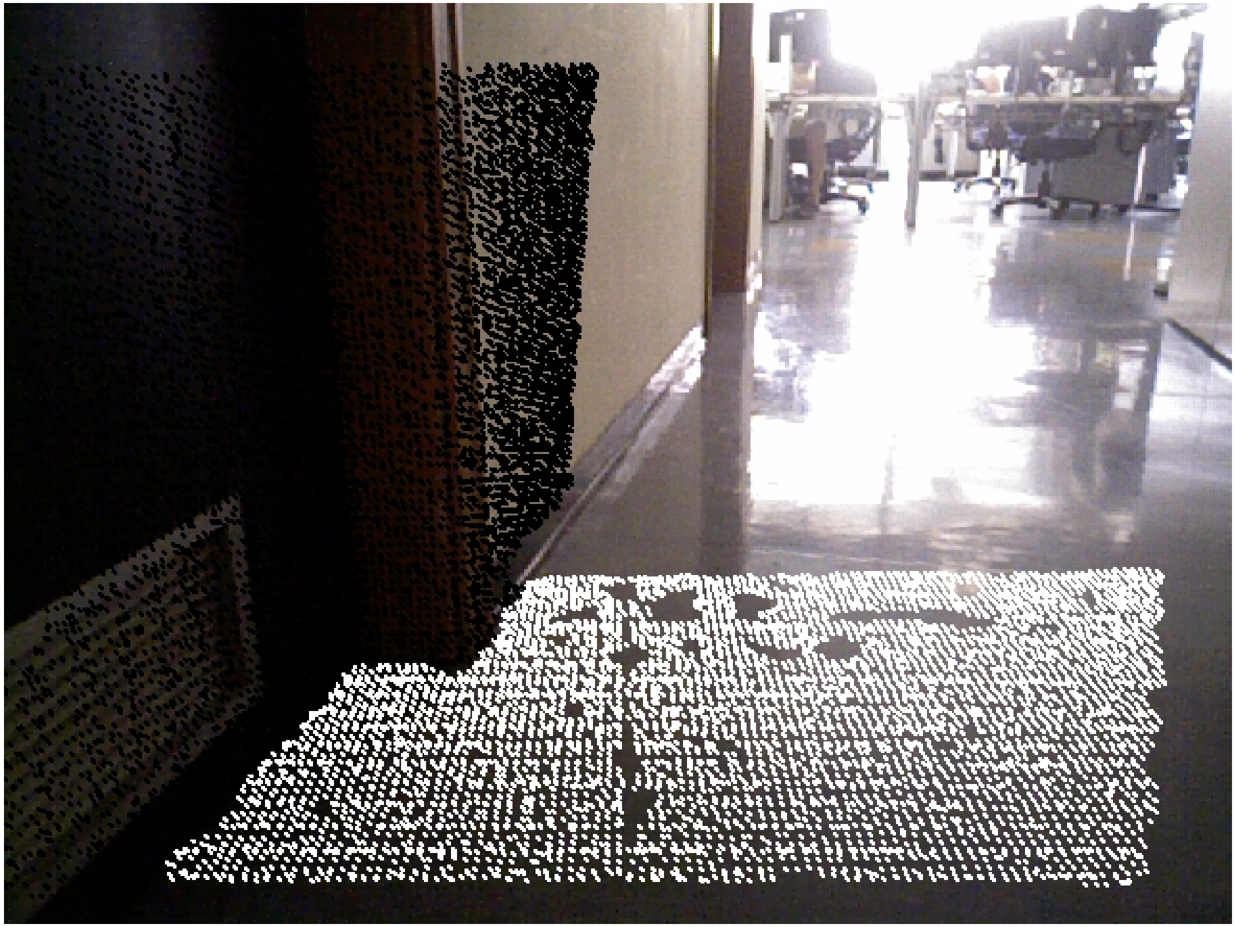
- Free corridor



# Experimental evaluation



- Free corridor



# Experimental evaluation



- Free corridor

# Experimental evaluation

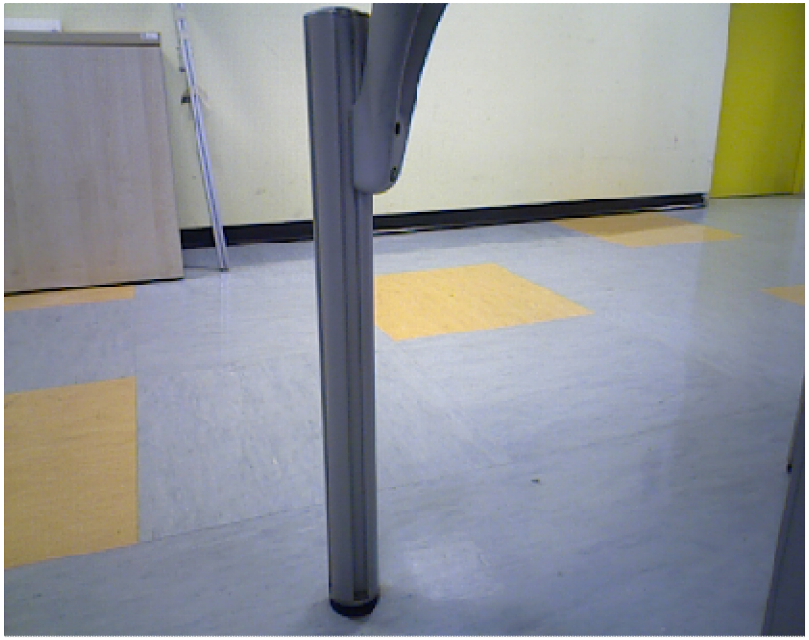


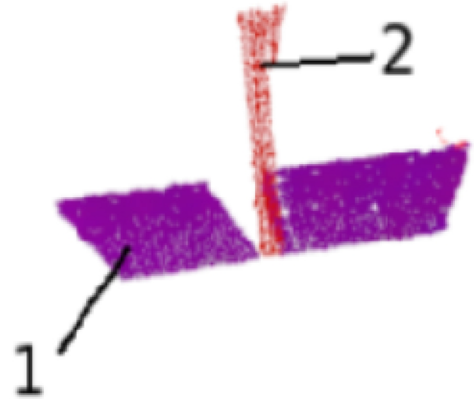
- Corridor with obstacles

# Experimental evaluation



- Corridor with obstacles

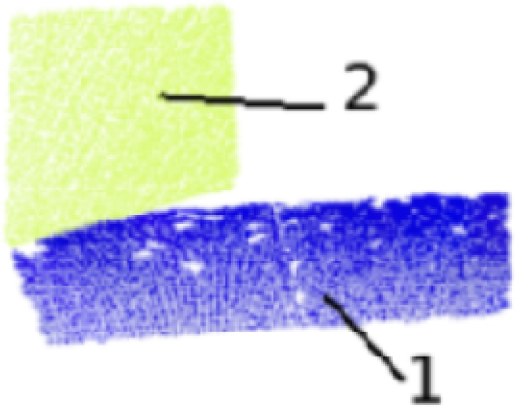





Slope_1=0°  Roughness_1=1.08
Slope_2=88.6°  Roughness_2=1.14

# Experimental evaluation

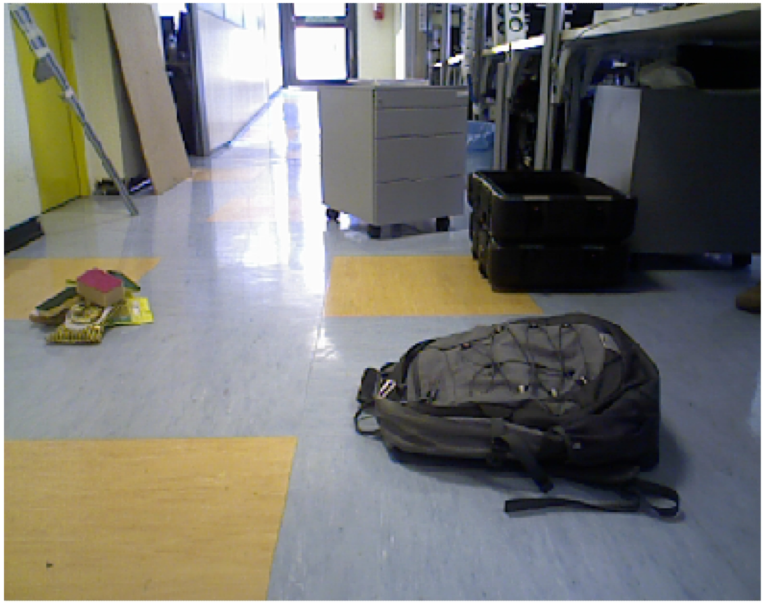
- Corridor with obstacles

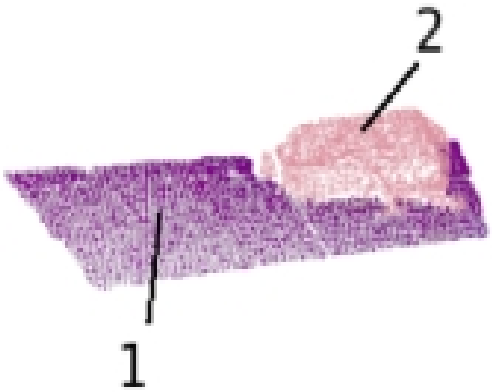



Slope_1=0°    Roughness_1=1.08
Slope_2=88.7°    Roughness_2=1.07

# Experimental evaluation

- Corridor with obstacles



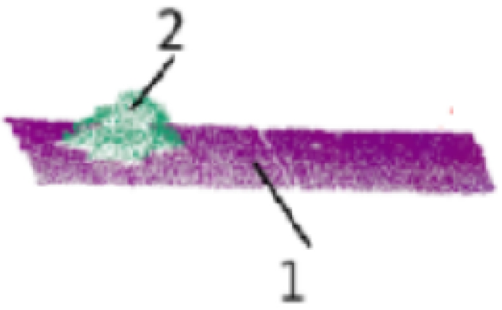



Slope_1=0°  Roughness_1=1.17
Slope_2=23.7°  Roughness_2=1.56

# Experimental evaluation



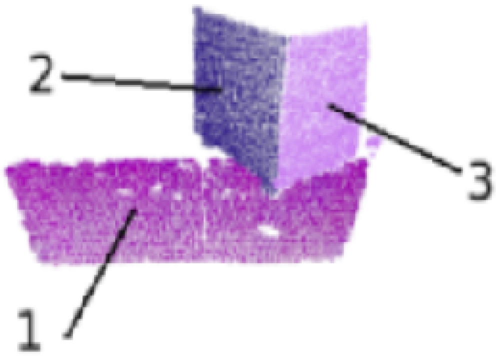
- Corridor with obstacles




Slope_1=0°　　Roughness_1=1.09
Slope_2=21.5°　　Roughness_2=2.01

# Experimental evaluation

- Corridor with obstacles

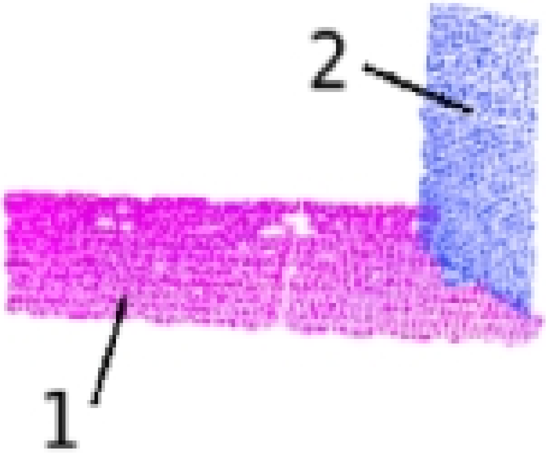



Slope_1=0°  Roughness_1=1.12
Slope_2=89.5°  Roughness_2=1.16
Slope_3=88.3°  Roughness_3=1.2

# Experimental evaluation



- Corridor with obstacles

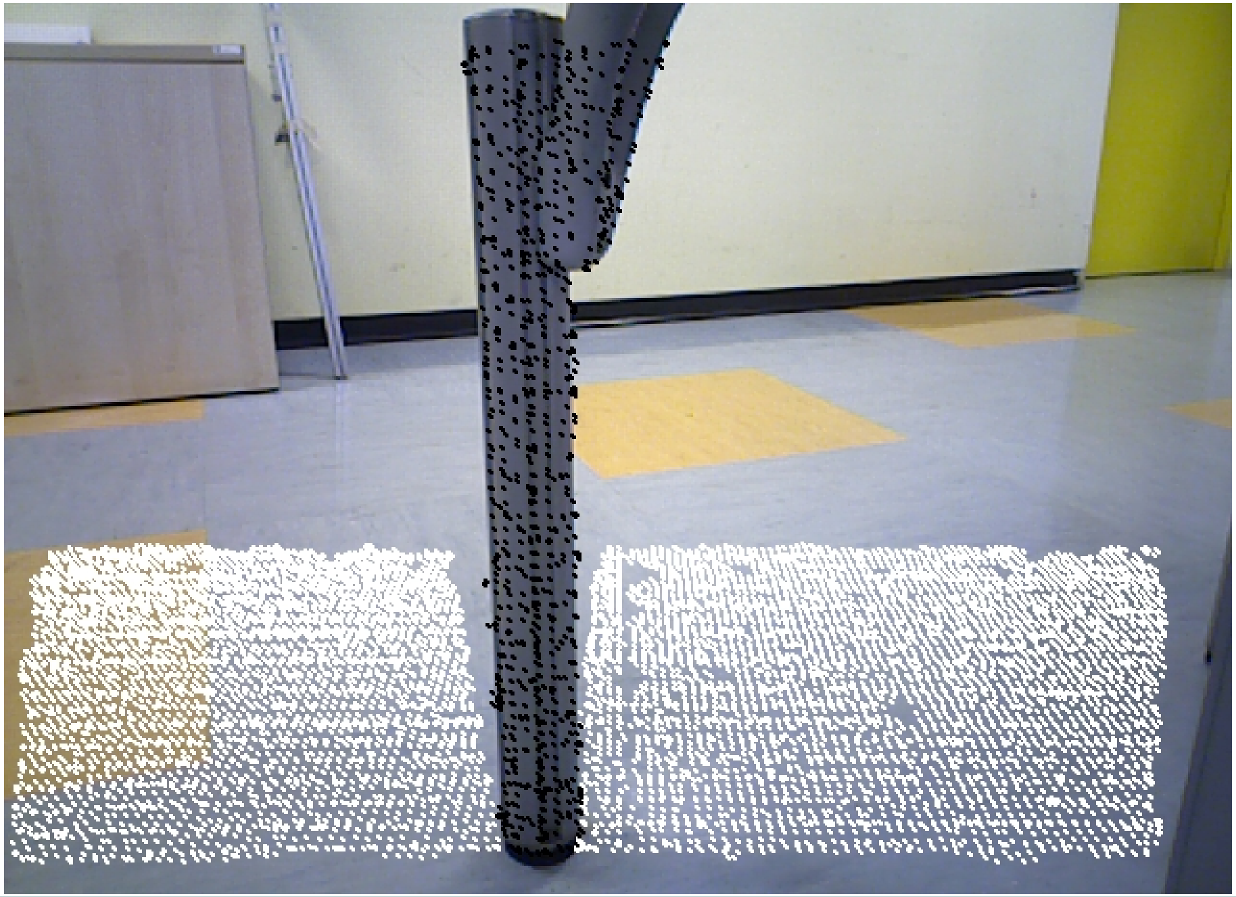



Slope_1=0° Roughness_1=1.15
Slope_2=89.7° Roughness_2=1.01

# Experimental evaluation



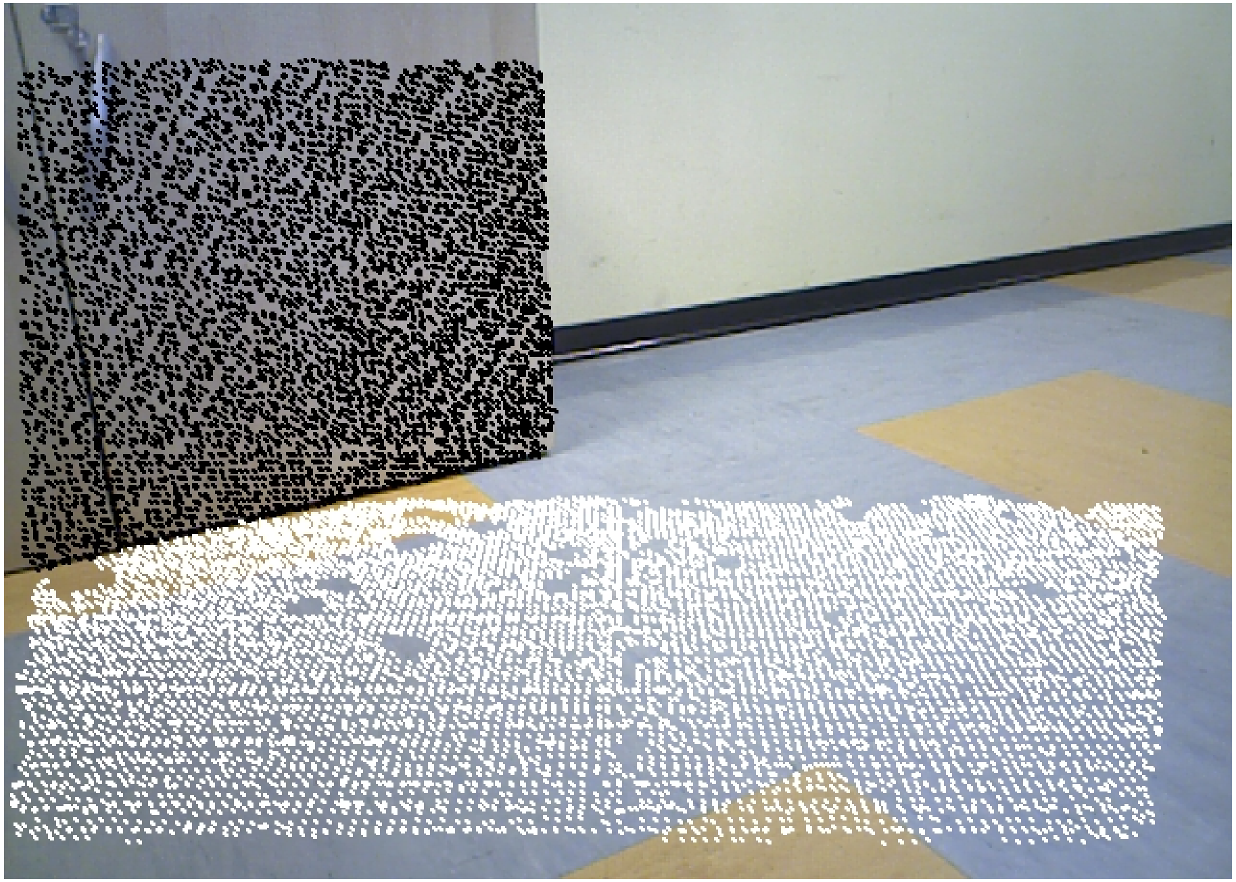
- Corridor with obstacles



# Experimental evaluation



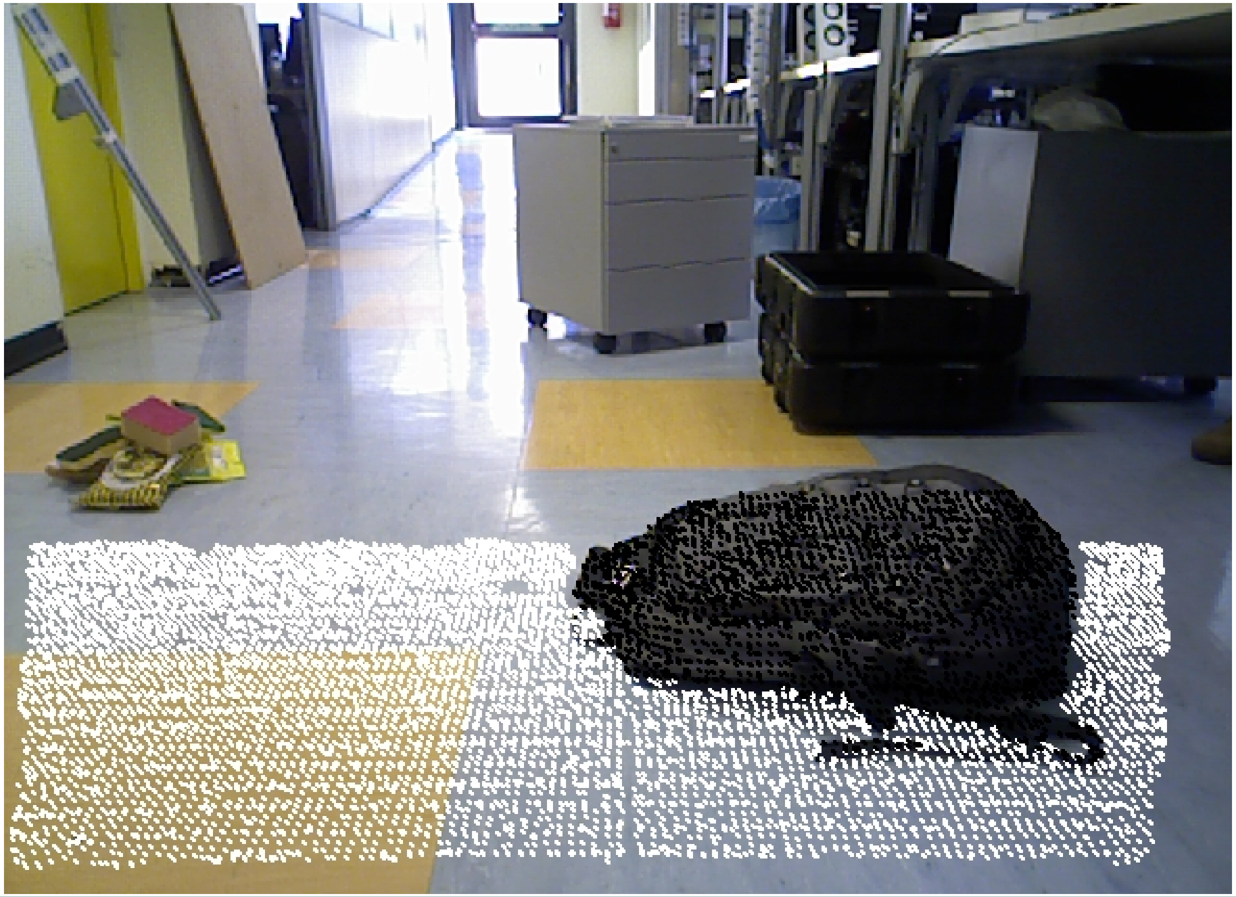
- Corridor with obstacles



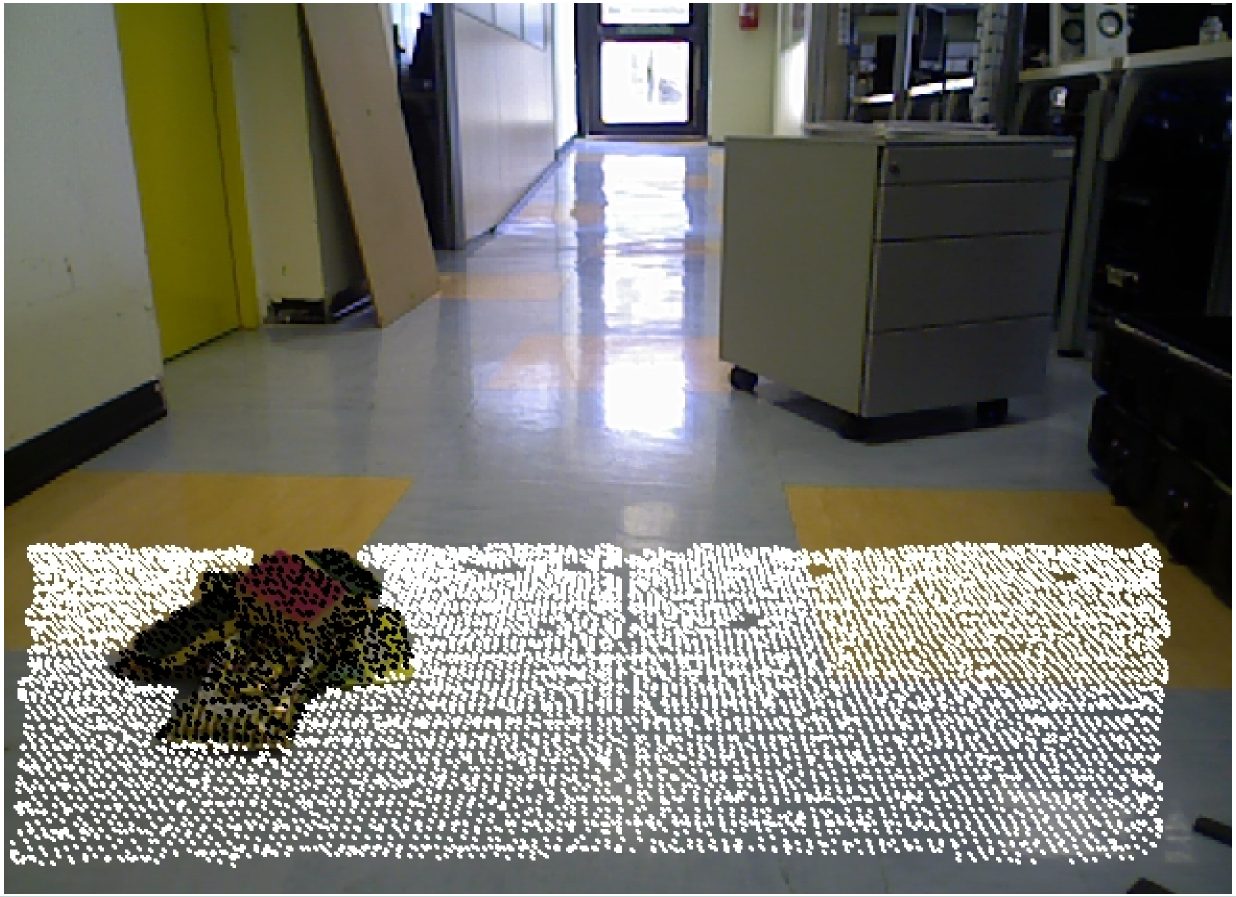
# Experimental evaluation



- Corridor with obstacles



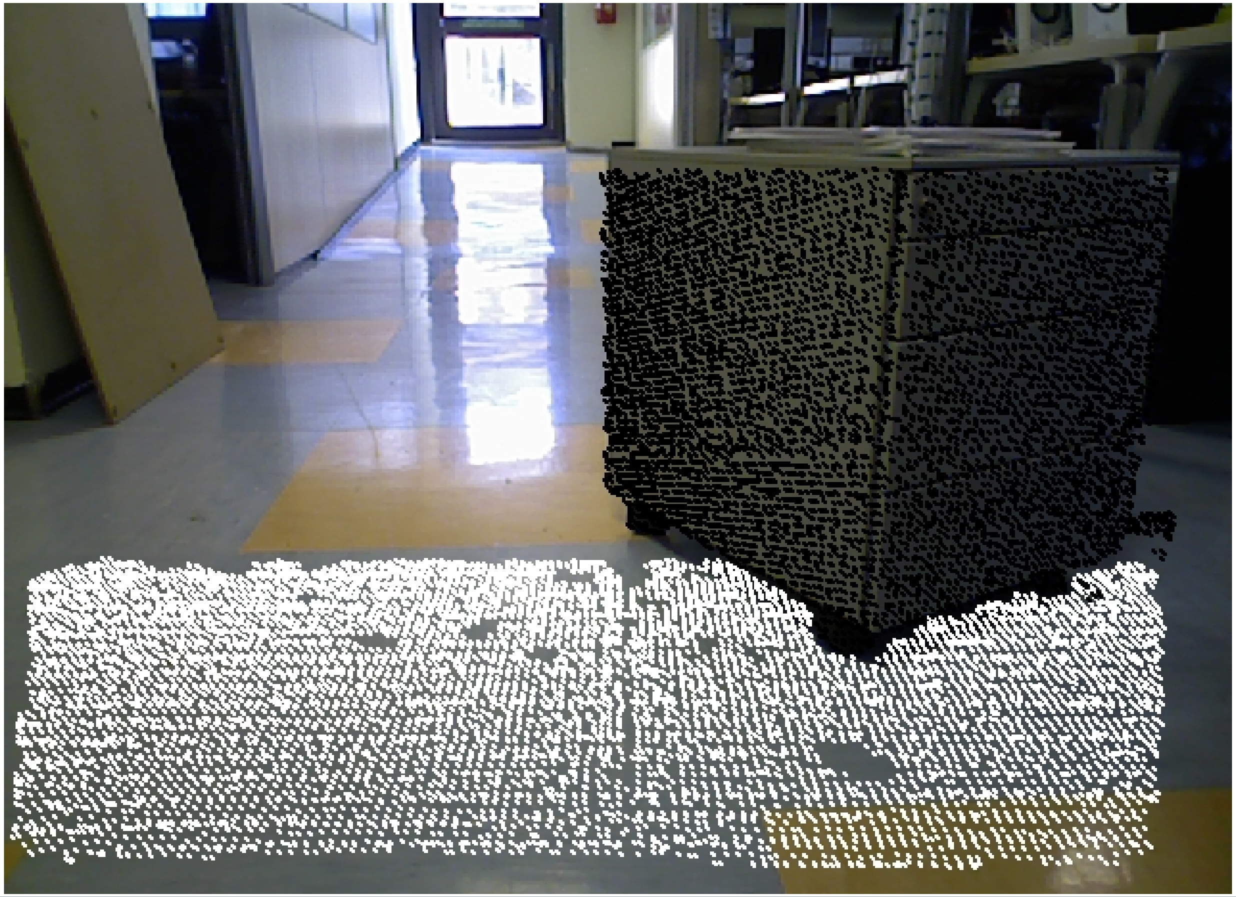
- Corridor with obstacles



# Experimental evaluation



- Corridor with obstacles



# Experimental evaluation



- Corridor with obstacles

# Experimental evaluation

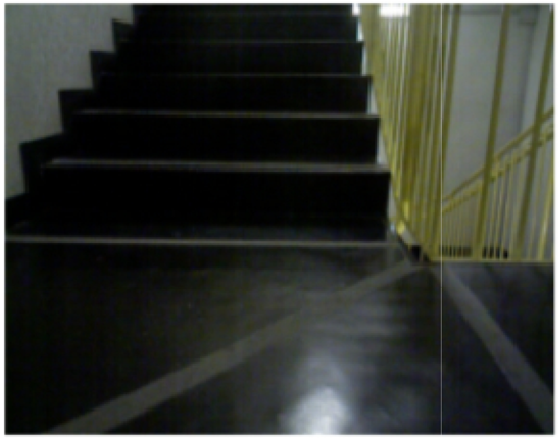


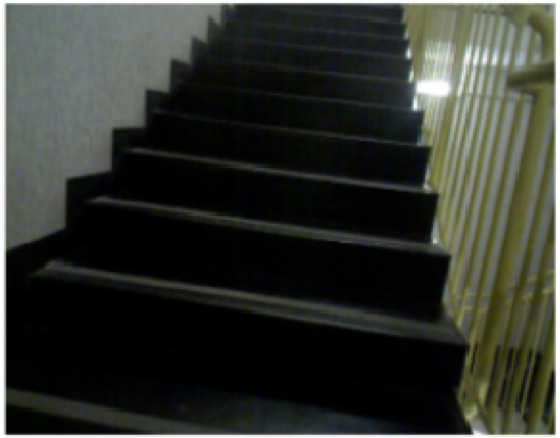
- Stairs

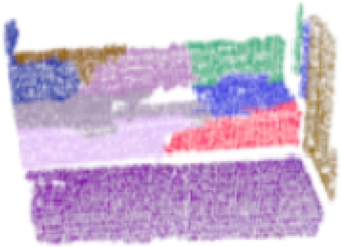
# Experimental evaluation

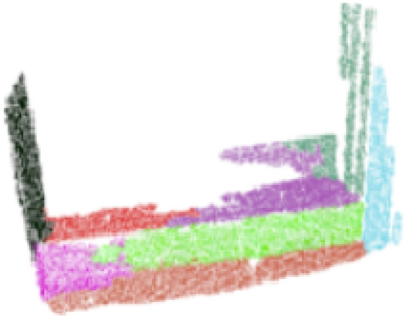


- Stairs





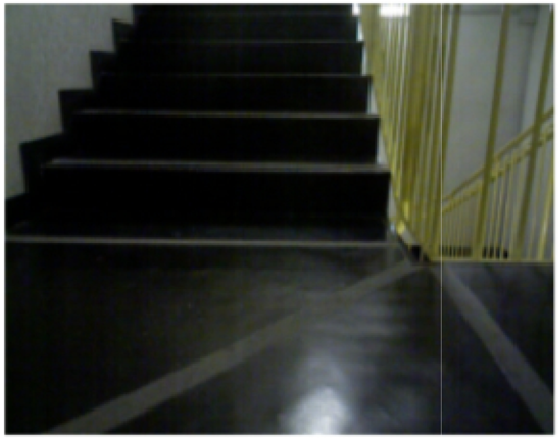


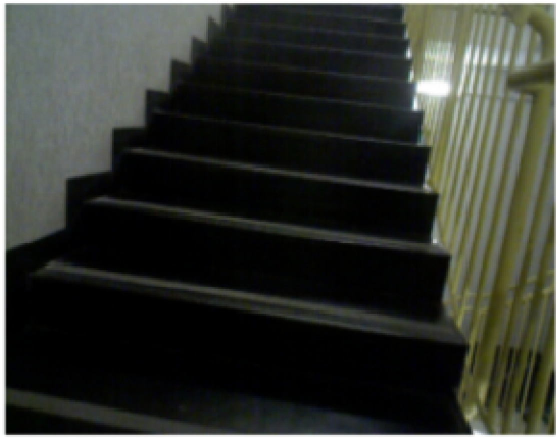


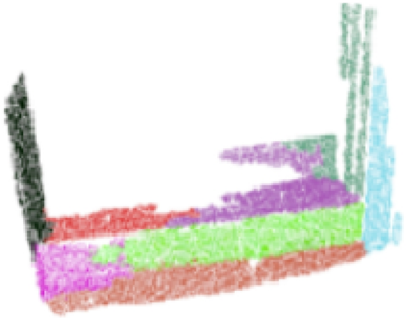
# Experimental evaluation



- Stairs



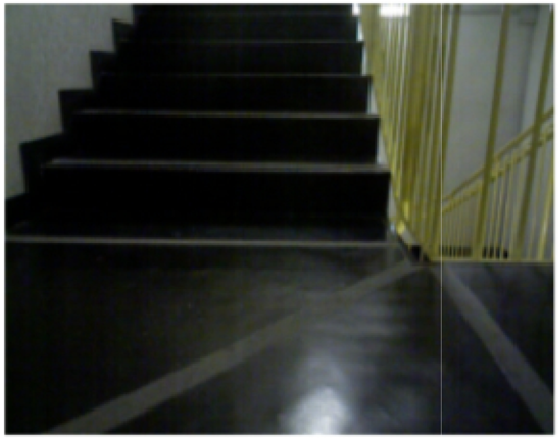


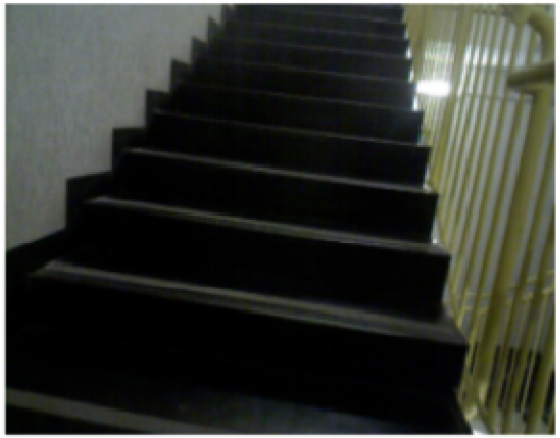


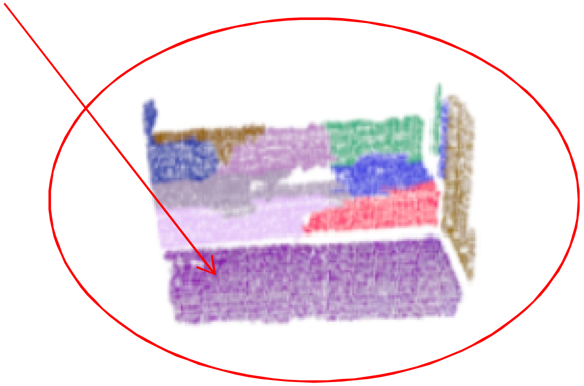
# Experimental evaluation

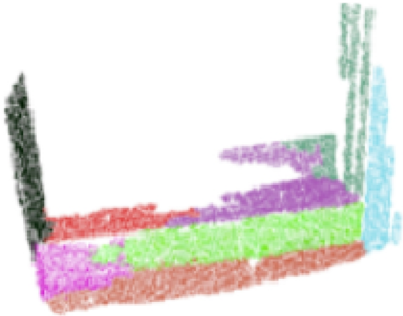


- Stairs





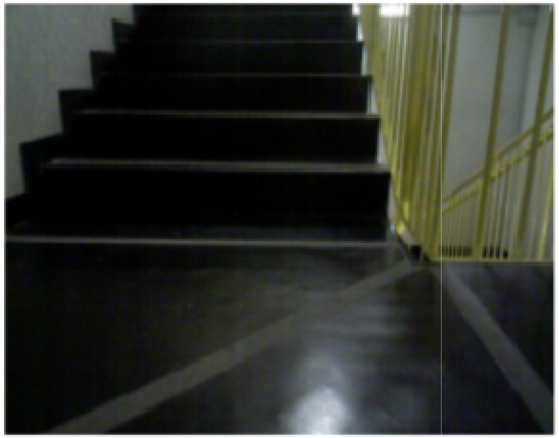


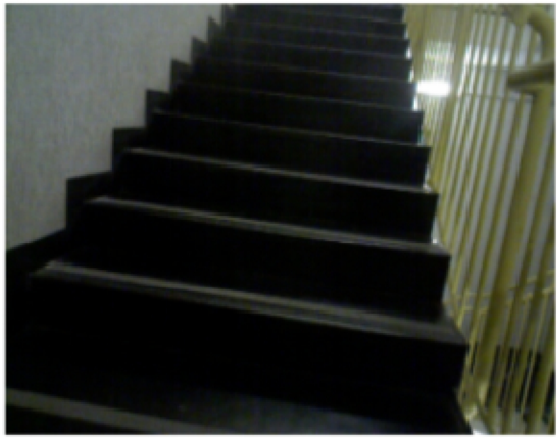


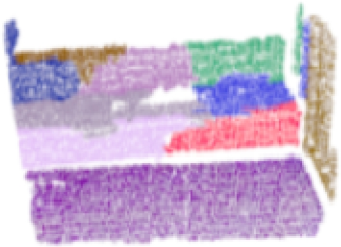
# Experimental evaluation

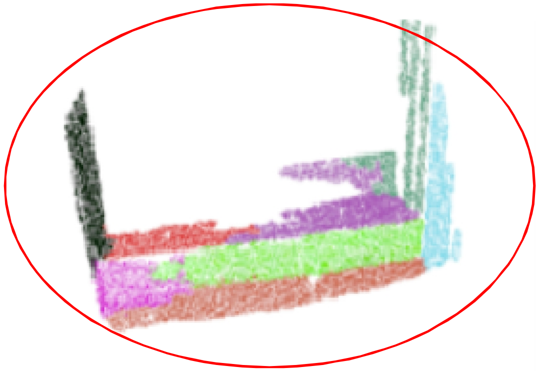


- Stairs









# Conclusions



- We have implemented a stable system able to build the mobility map of the scene

# Conclusions



- We have implemented a stable system able to build the mobility map of the scene
- The system was successfully evaluated in a real world indoor environment

# Conclusions



- We have implemented a stable system able to build the mobility map of the scene
- The system was successfully evaluated in a real world indoor environment
- Issues remarked & limitations

# Conclusions



- We have implemented a stable system able to build the mobility map of the scene
- The system was successfully evaluated in a real world indoor environment
- Issues remarked & limitations
- Future works

# Conclusions



- We have implemented a stable system able to build the mobility map of the scene
- The system was successfully evaluated in a real world indoor environment
- Issues remarked & limitations
- Future works
  - Removing the assumption on the ground, and treatment of workspaces where the floor is not present or occluded by objects.
  - Noise removal and feature estimation "at the same time".

# Conclusions



- We have implemented a stable system able to build the mobility map of the scene
- The system was successfully evaluated in a real world indoor environment
- Issues remarked & limitations
- Future works
  - Removing the assumption on the ground, and treatment of workspaces where the floor is not present or occluded by objects.
  - Noise removal and feature estimation “at the same time”.
  - Improving the segmentation using other features as for example texture or more in general appereance.
  - Tests on iCub

Thanks for the attention!
Have a nice day!

Nicolò