UNIVERSIDAD
DE MÁLAGA

AIM2014
Besançon
FRANCE
IEEE/ASME International Conference on Advanced Intelligent Mechatronics

Collapsible Cubes: Removing Overhangs from 3D Point Clouds to Build Local Navigable Elevation Maps

Antonio J. Reina, Jorge L. Martínez, Anthony Mandow,
Jesús Morales, and Alfonso García-Cerezo

Dpto. Ingeniería de Sistemas y Automática
Universidad de Málaga (Spain)

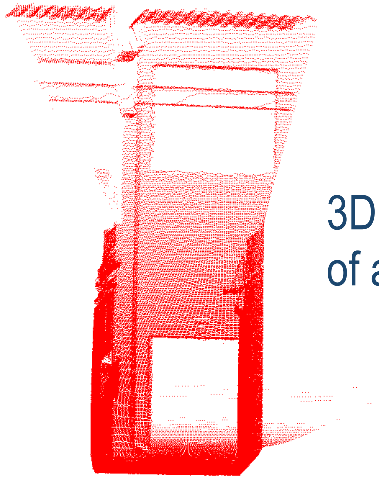
OUTLINE

1. OVERVIEW
2. CBCs DATA STRUCTURES
3. COLLAPSING CUBES METHOD
4. EXPERIMENTAL RESULTS
5. CONCLUSIONS

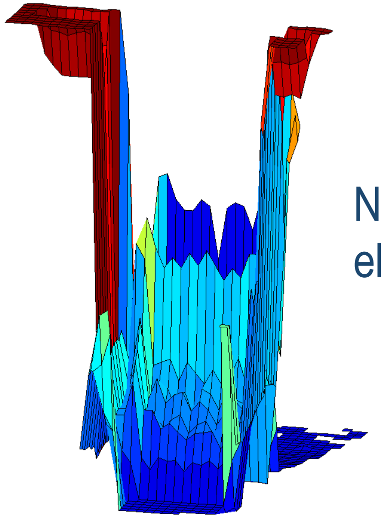
1.
OVERVIEW

Overview

- ***Elevation maps:***
  - *Compact 2½ D terrain surface model*
- ***3D point clouds***
  - *Overhangs produce unreliable maps*
- ***Goal***
  - *Identification and removal of overhangs from point clouds*
  - *Collapsing cubes instead of a point-based gap search*
  - *Use of coarse binary cubes (CBCs) data structures*



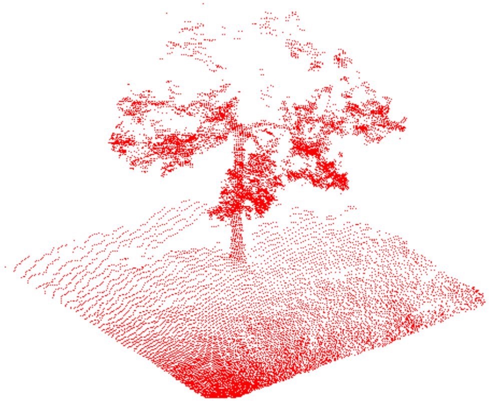
3D point cloud of an open door

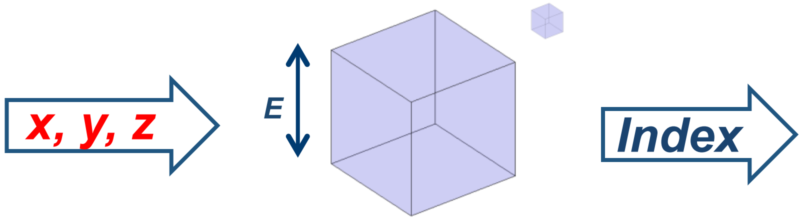


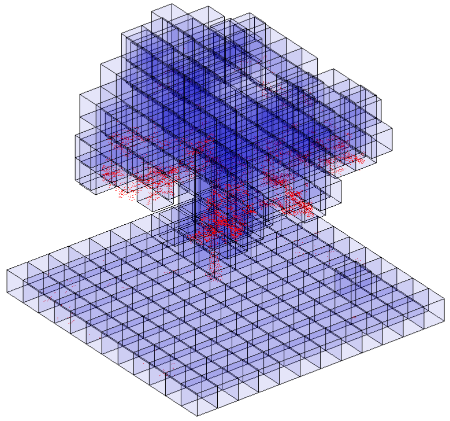
Non-navigable elevation map

2.
CBCs DATA STRUCTURES

Coarse Binary Cubes Data Structures (CBCs)

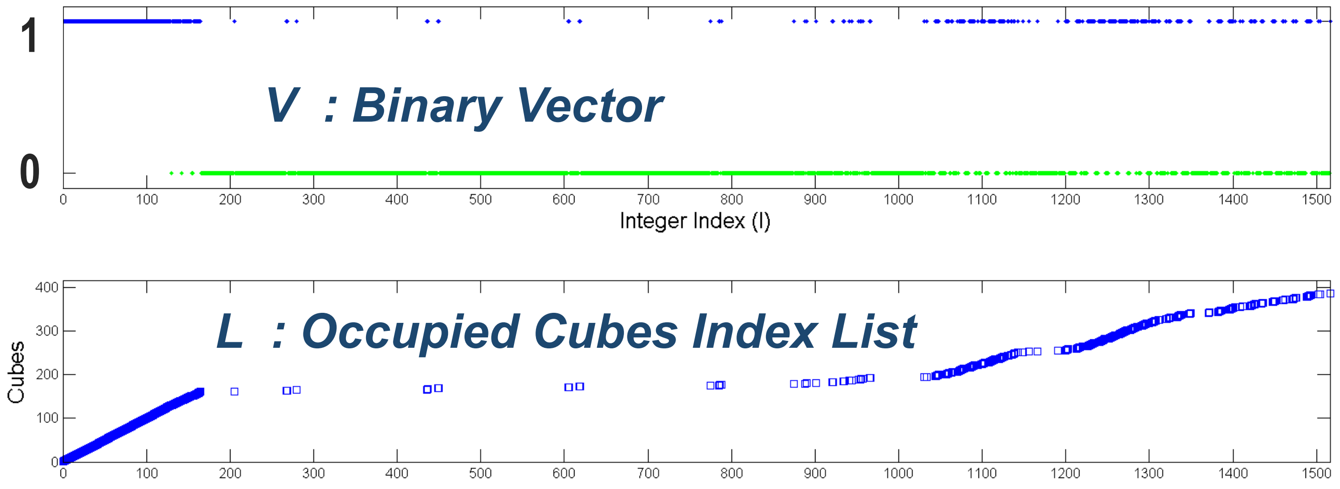






Each cube has a unique integer index I → (x,y,z,I)

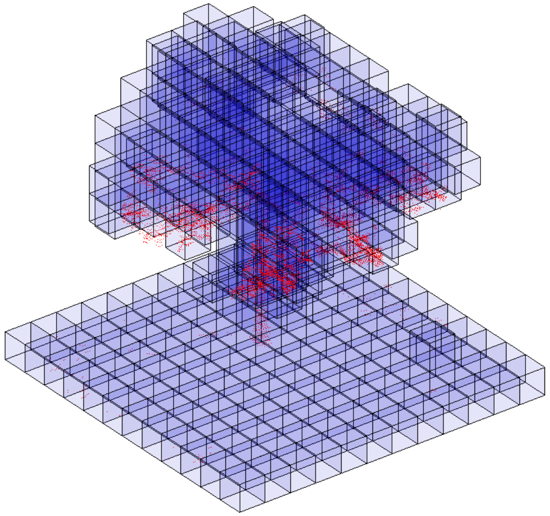
1D CBCs data structures



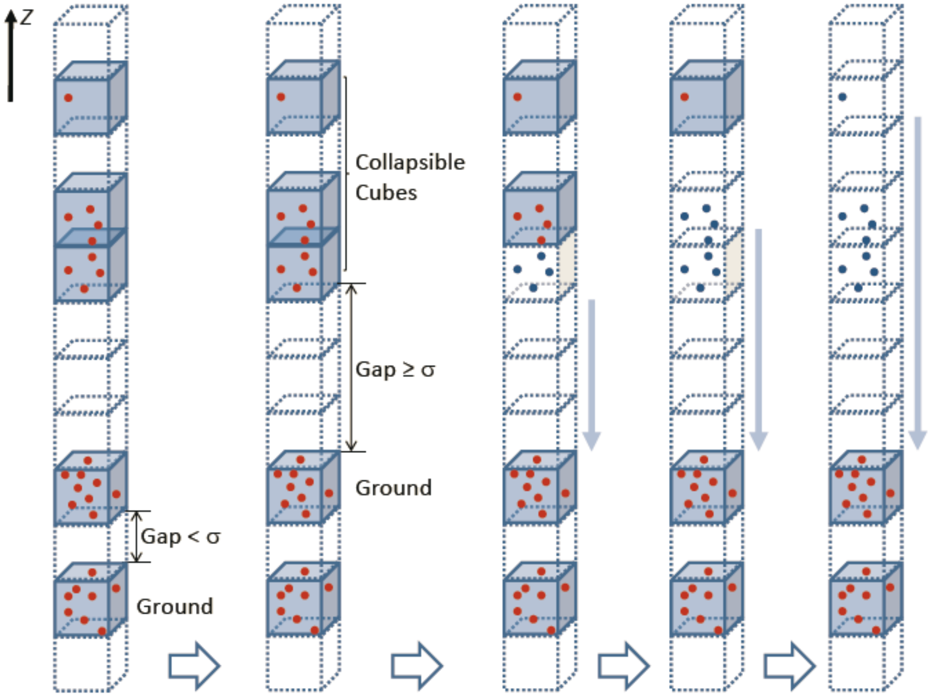
3.
COLLAPSING CUBES METHOD

Collapsing Cubes Method

- **PRINCIPLE: Classify 3D points as ground (including vertical obstacles) and overhangs**

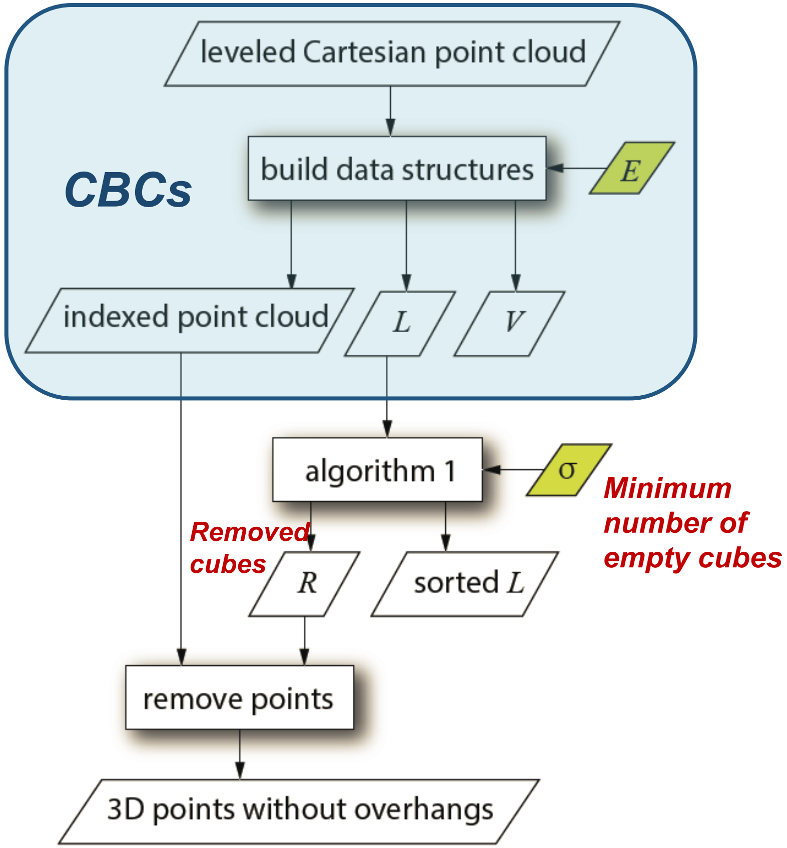


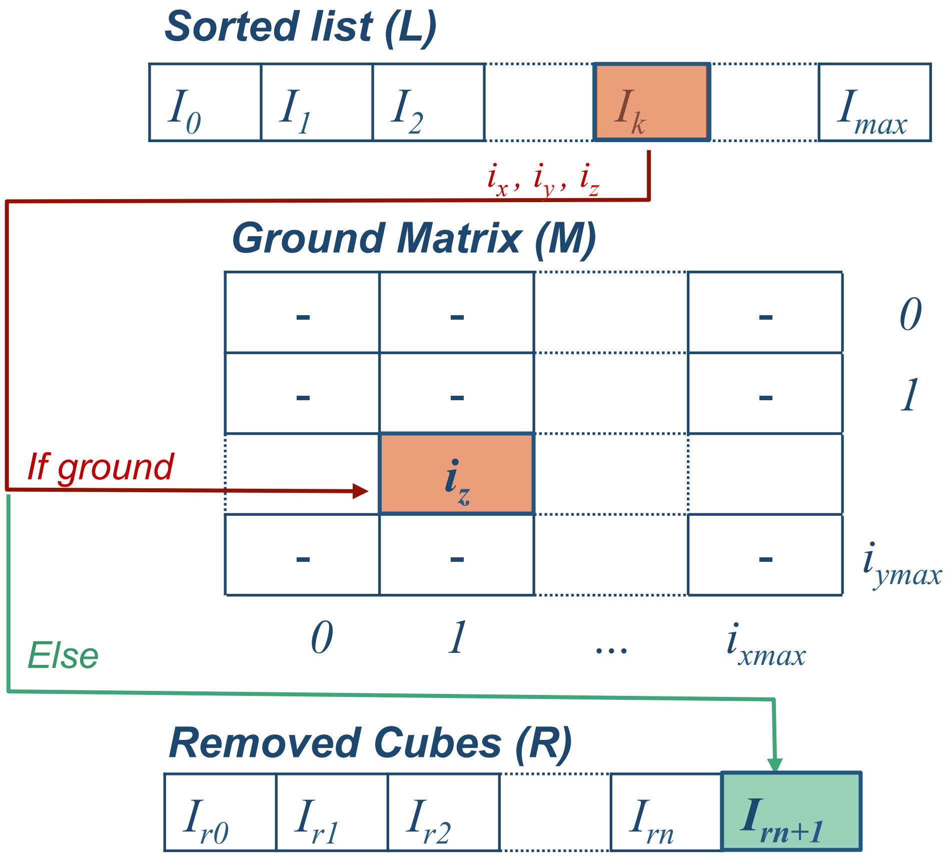
Sorting index list L implies visiting the lowest occupied cubes in the first place



Collapsing Cubes Method

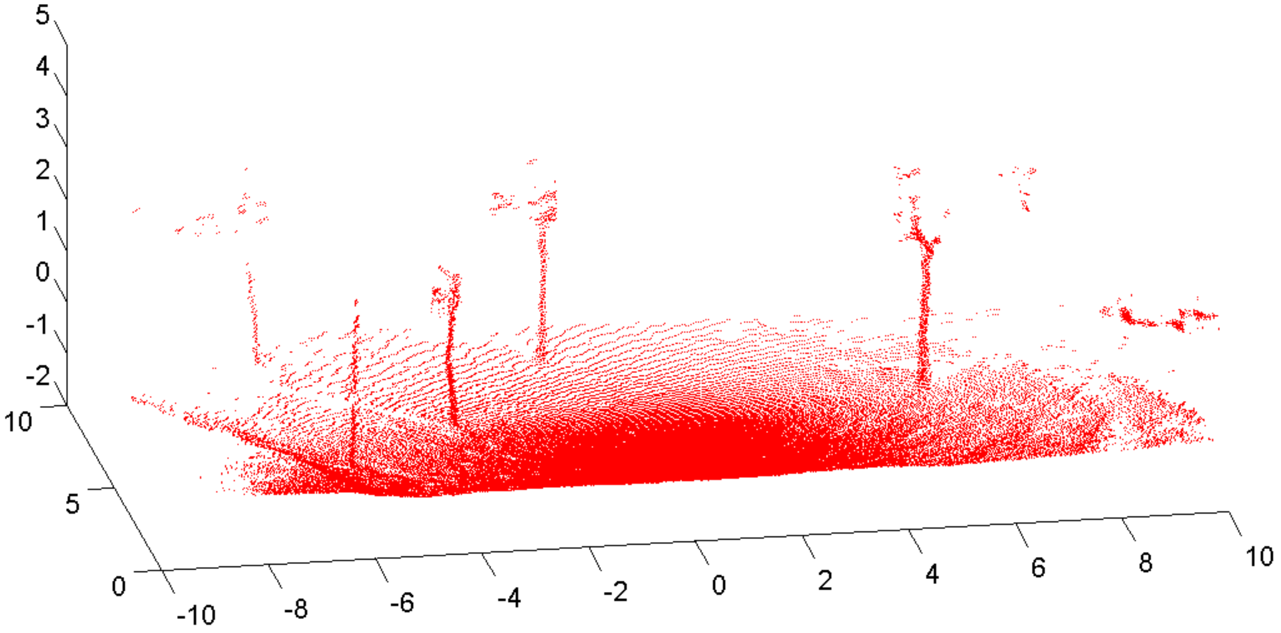
- **Implementation with CBCs data structures**





UNIVERSIDAD DE MÁLAGA

Collapsed Cubes Animation

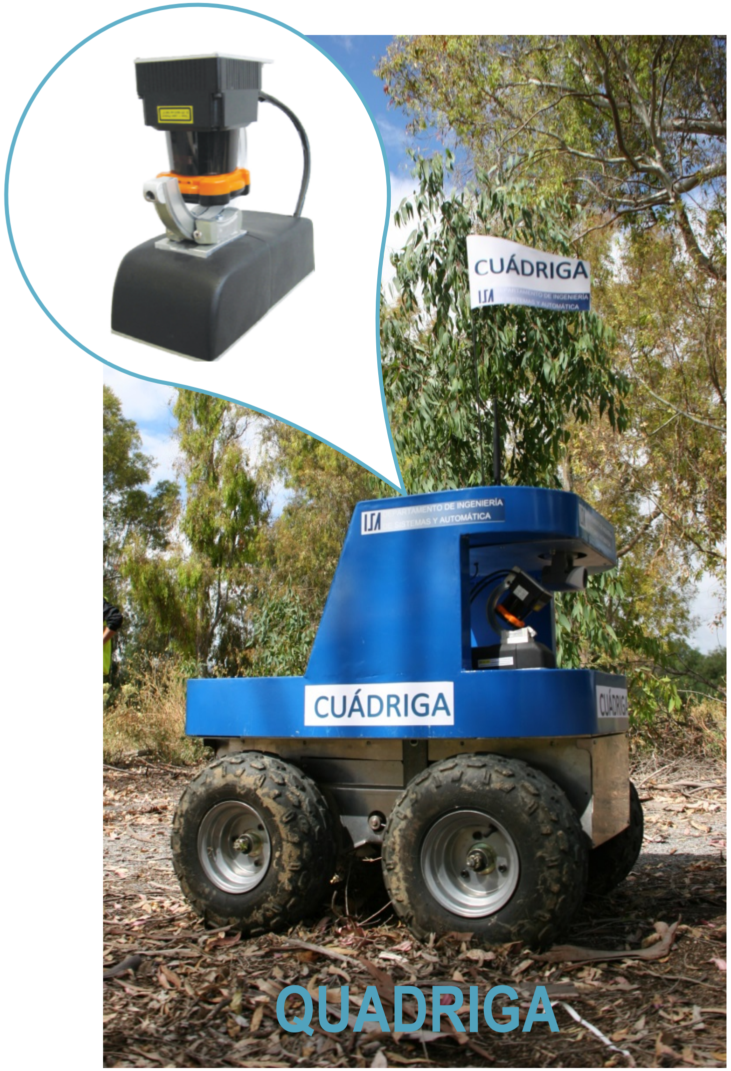


UNIVERSIDAD DE MÁLAGA

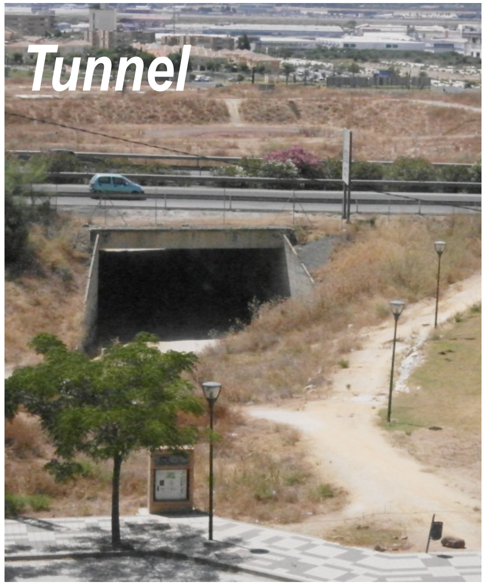
4.
EXPERIMENTAL RESULTS

UNIVERSIDAD DE MÁLAGA

Experimental Setup

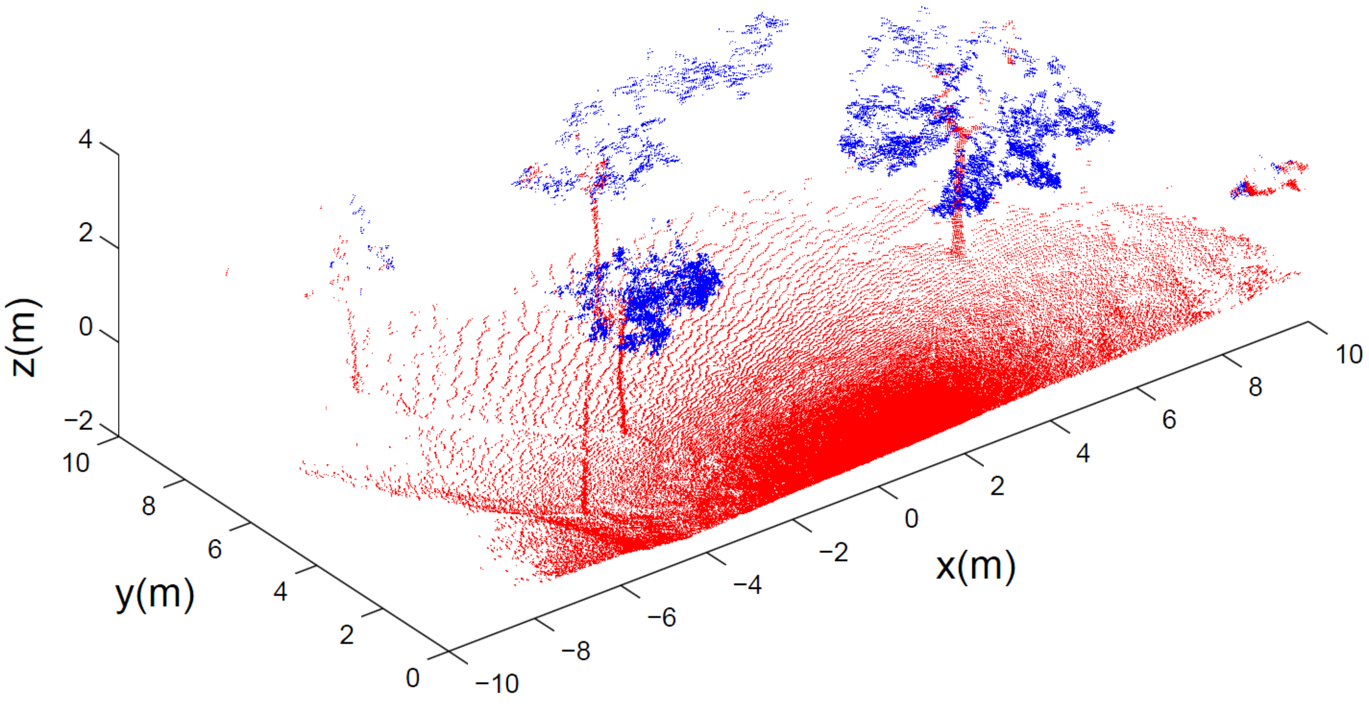


- **UnoLaser 3D Scanner:**
  - *30 m range*
  - *0.7 m above ground*
- **Elevation Maps**
  - *Subsampling resolution δ = 0.1m*
  - *20x10 (meter)*



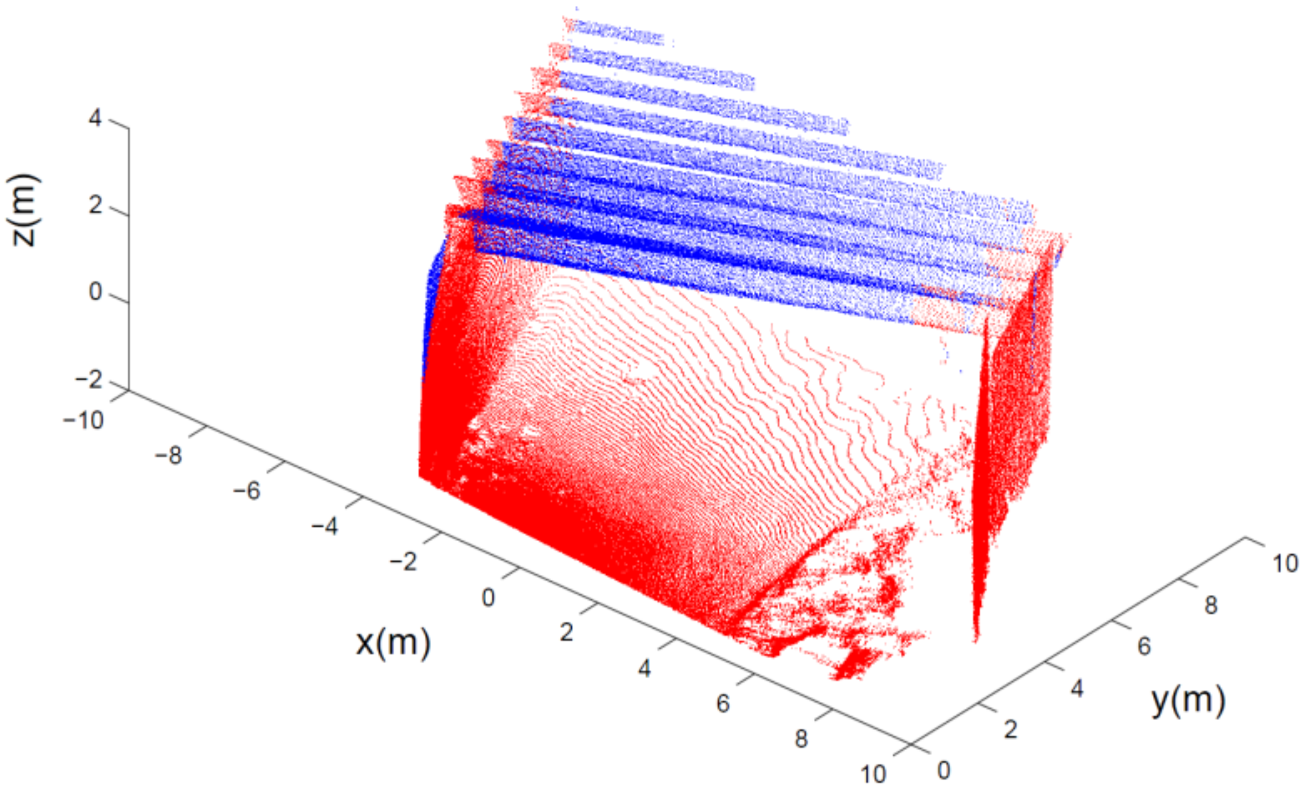
Results: Point Classification

1st Outdoor Environment: TREES



Results: Point Classification

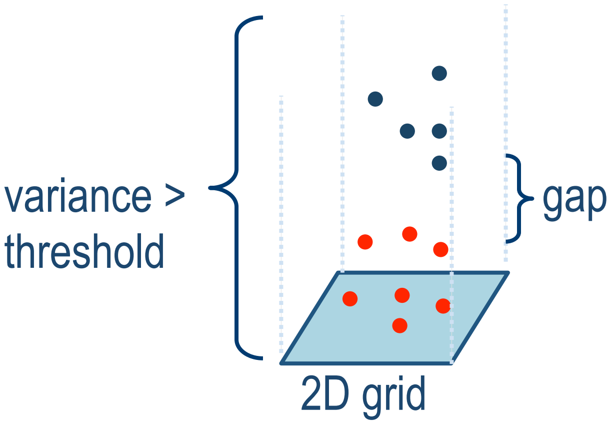
2do Outdoor Environment: TUNNEL



Results: Computational Times

Comparison between point-based and cube-based

Point-based Method

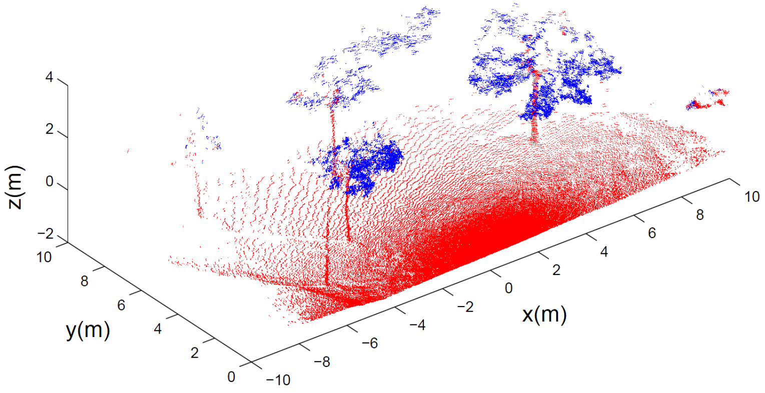


COMPUTATION TIMES

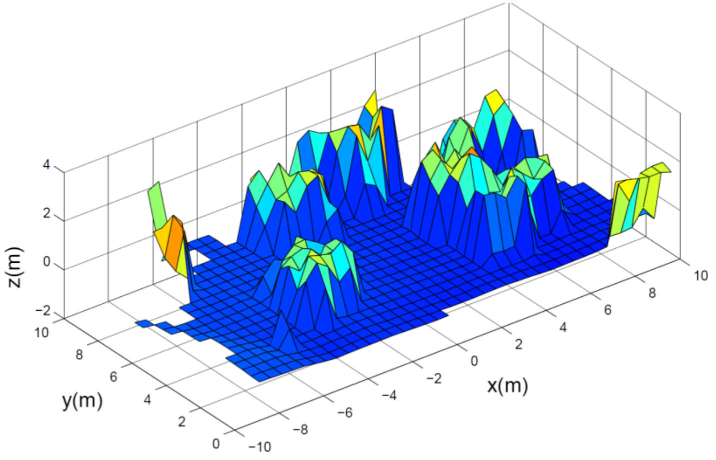
MatLab on a Intel Core i7

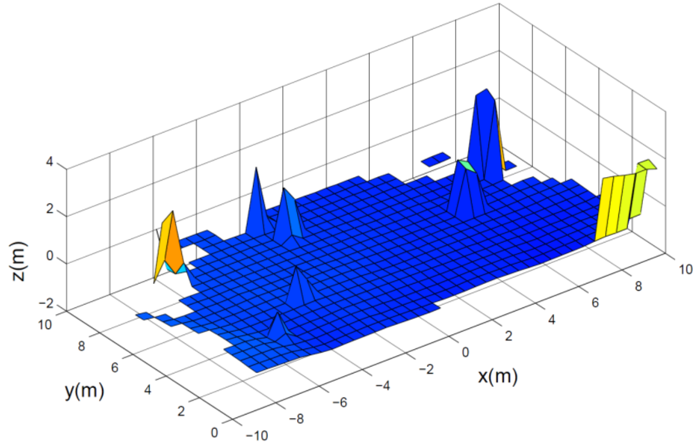
Method /Scene	TREES	TUNNEL
Cube-based	**0.21 s**	**0.37 s**
Point-based	**0.39 s**	**0.54 s**

Results: Appication to Build Elevation Maps

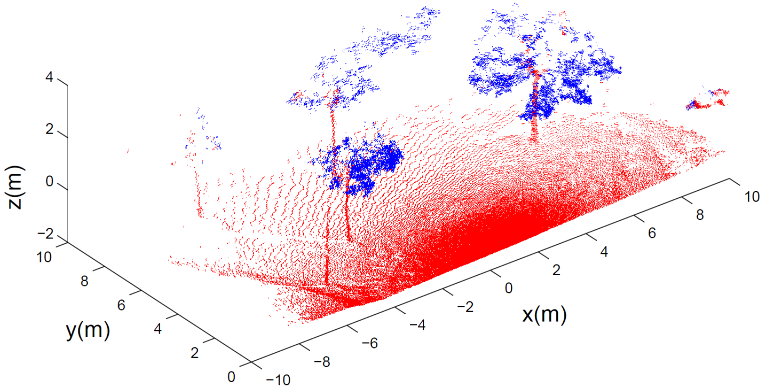


- **Standard Elevation Map**

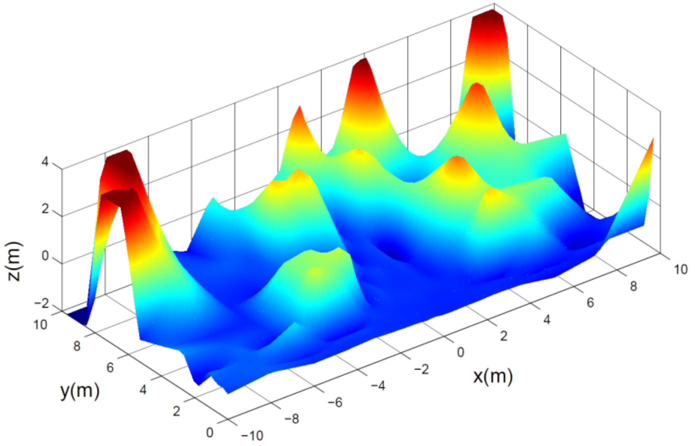


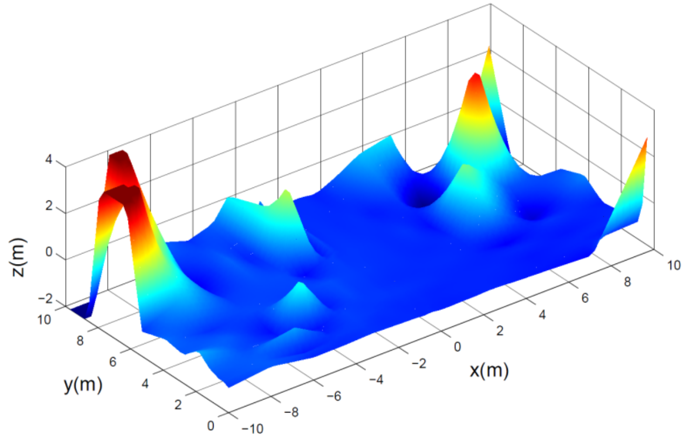


Results: Appication to Build Elevation Maps

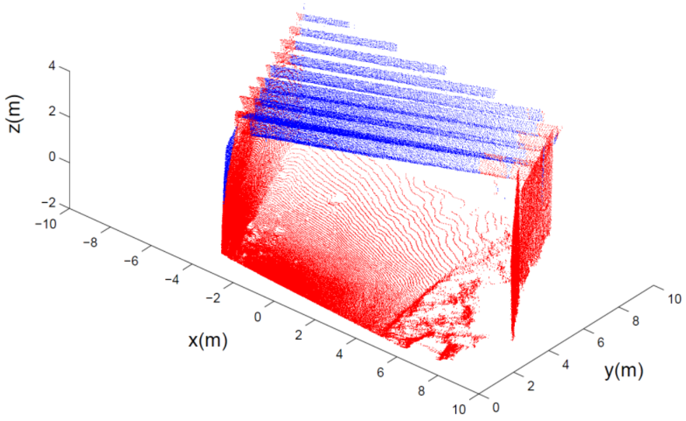


- **Fuzzy Elevation Map**

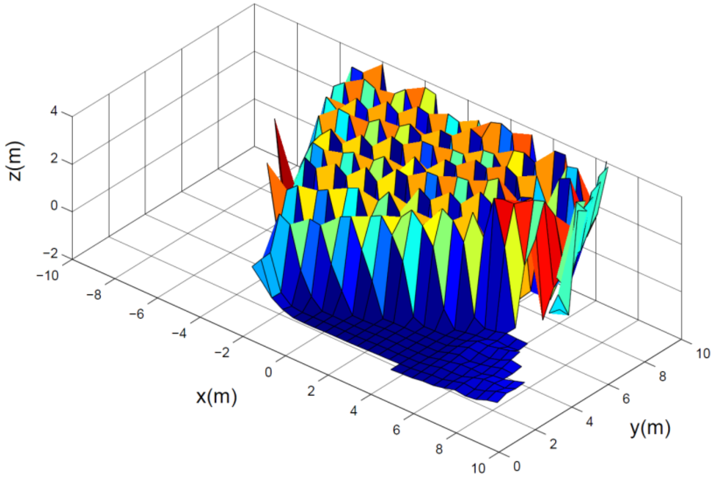


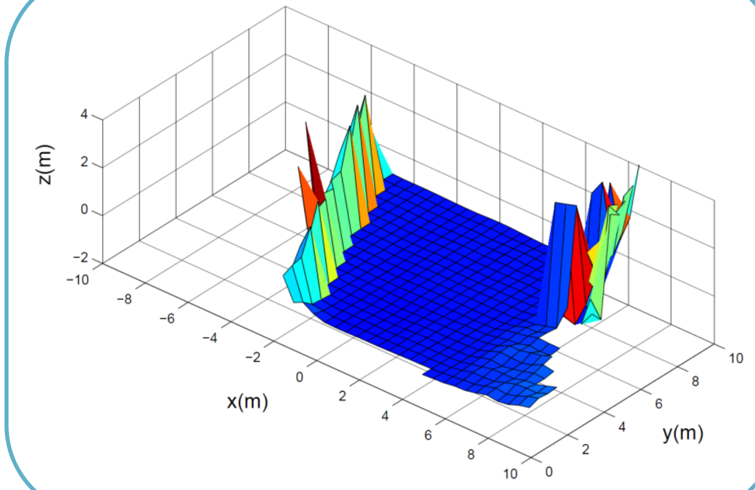


Results: Appication to Build Elevation Maps

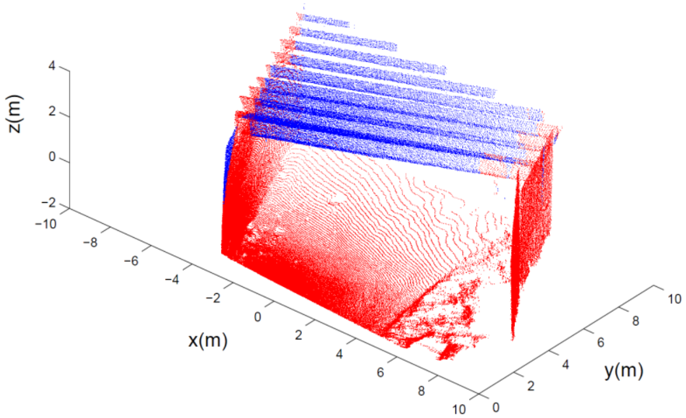


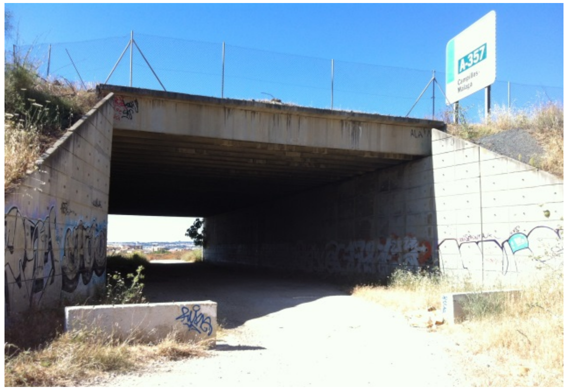
- **Standard Elevation Map**



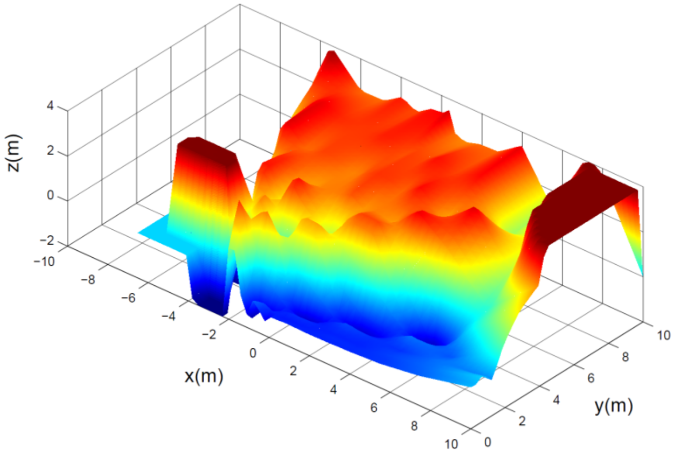


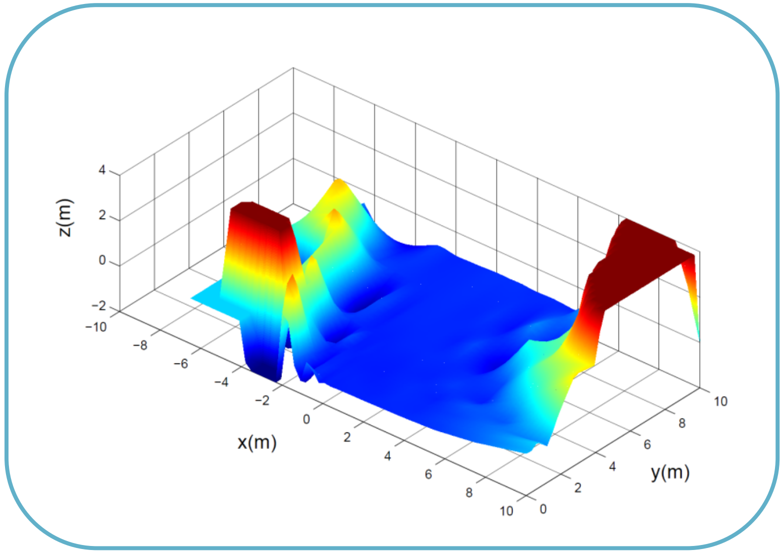
Results: Appication to Build Elevation Maps





- **Fuzzy Elevation Map**





5.
CONCLUSIONS

Conclusions

- **Simple processing of leveled 3D point cloud that identifies and removes overhang points**
  - *Efficient data structures from coarse binary cubes*
  - *An occupied cube is collapsible when a gap is detected*
  - *For navigation task this gap depends on the mobile robot height*
- **Improvement in computational times with respect to point-based solution**
  - *Verified in different outdoor environments*
- **Employed to build reliable standard and fuzzy elevation maps**
- **Work in progress**
  - *Navigation with the Quadriga mobile robot based on local planned paths from the FEMs*

UNIVERSIDAD DE MÁLAGA

Thank you! Merci!

www.isa.uma.es/ajreina

Integer Index Cubes

$$i_x = \text{round}\left(\frac{x - x_{min}}{E}\right)$$

$$i_y = \text{round}\left(\frac{y - y_{min}}{E}\right)$$

$$i_z = \text{round}\left(\frac{z - z_{min}}{E}\right)$$

$$i_{xmax} = \text{round}\left(\frac{x_{max} - x_{min}}{E}\right) + 1,$$

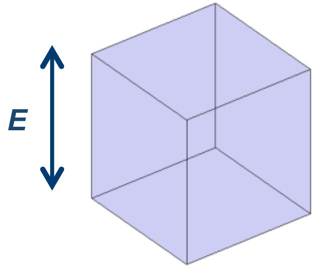
$$i_{ymax} = \text{round}\left(\frac{y_{max} - y_{min}}{E}\right) + 1,$$

$$I = i_x + i_y\, i_{xmax} + i_z\, i_{xmax}\, i_{ymax},$$

$$i_x = \text{remainder}\left(\frac{I}{i_{xmax}}\right),$$

$$i_y = \text{remainder}\left(\frac{(I - i_x)/i_{xmax}}{i_{ymax}}\right),$$

$$i_z = \frac{I - i_x - i_y\, i_{xmax}}{i_{xmax}\, i_{ymax}}.$$



Results: Computational Times

COMPUTATION TIMES

MatLab on a Intel Core i7

Method /Scene	TREES	TUNNEL
Cube-based	**0,21 s**	**0,37 s**
Point-based	**0,39 s**	**0,54 s**
Occupied Cubes	**1,407 (0.5 m)**	**2,910 (0.5 m)**
Scan Points	**130,580**	**235,641**