Plane-based Object Categorization using Relational Learning

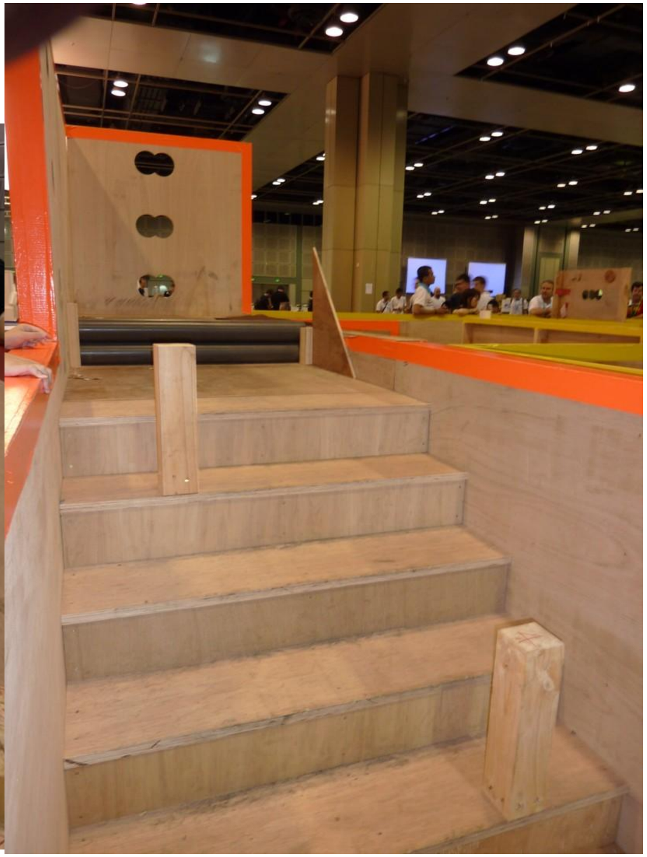
Reza Farid, Claude Sammut

Never Stand Still

School of Computer Science and Engineering, The University of New South Wales

Urban Search and Rescue

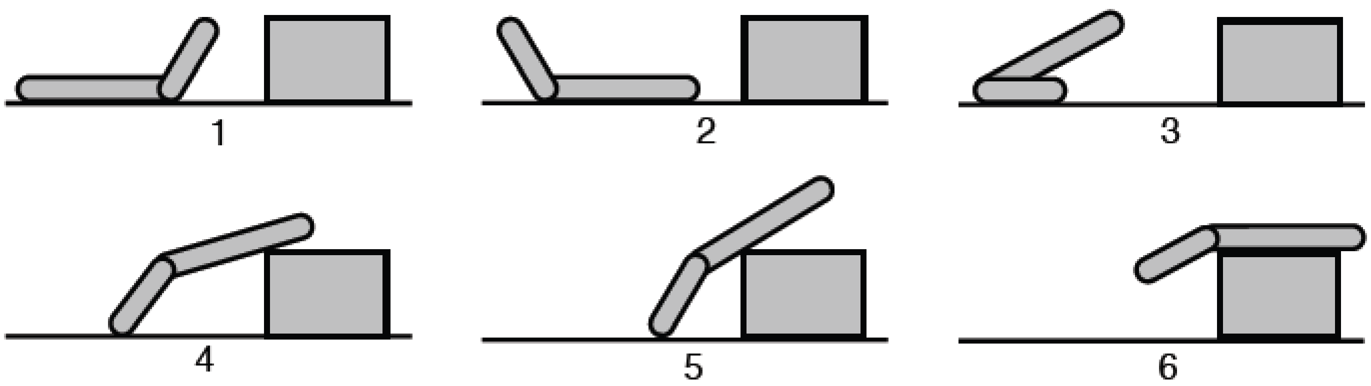
Rescue arena



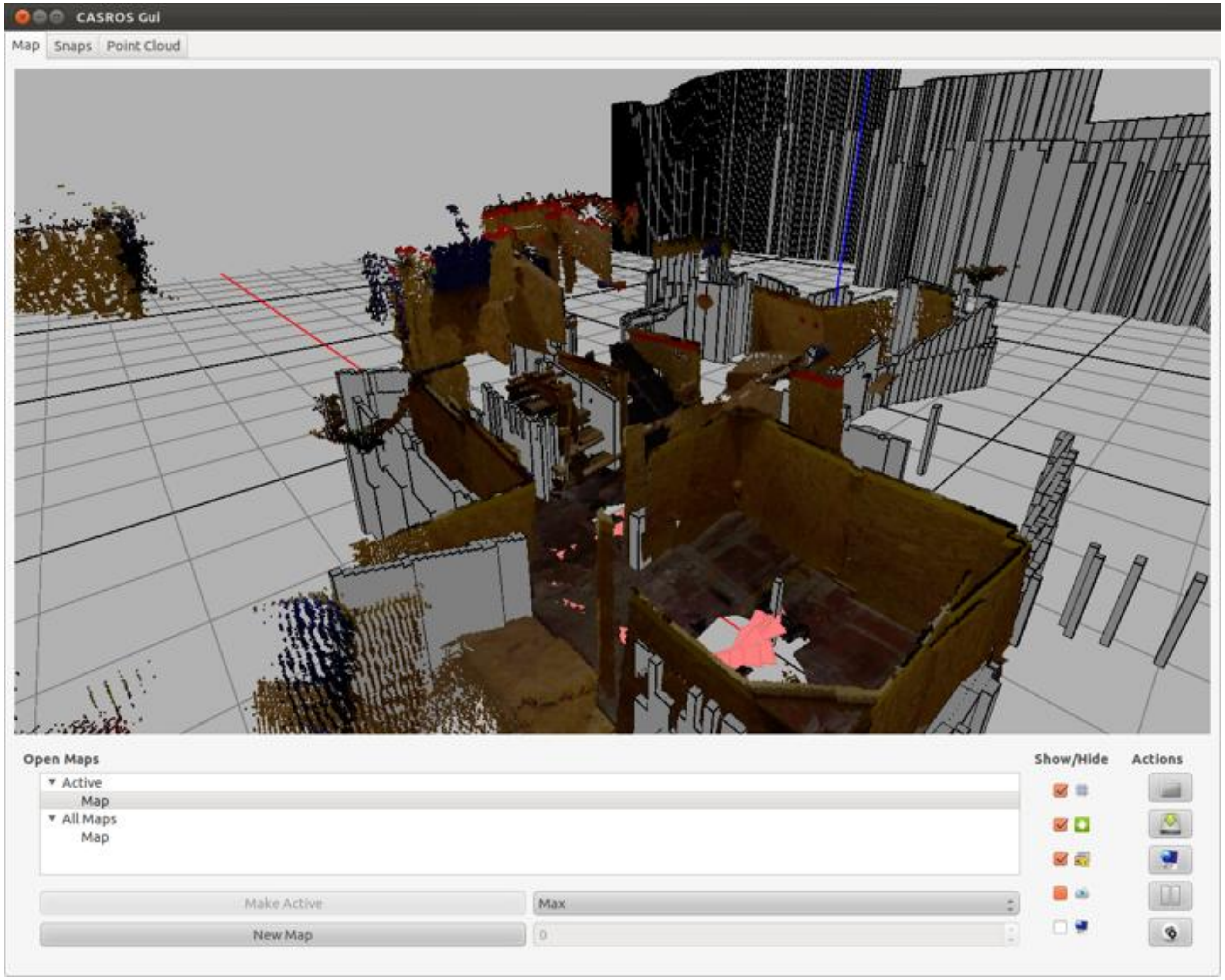
Rescue arena

Related work

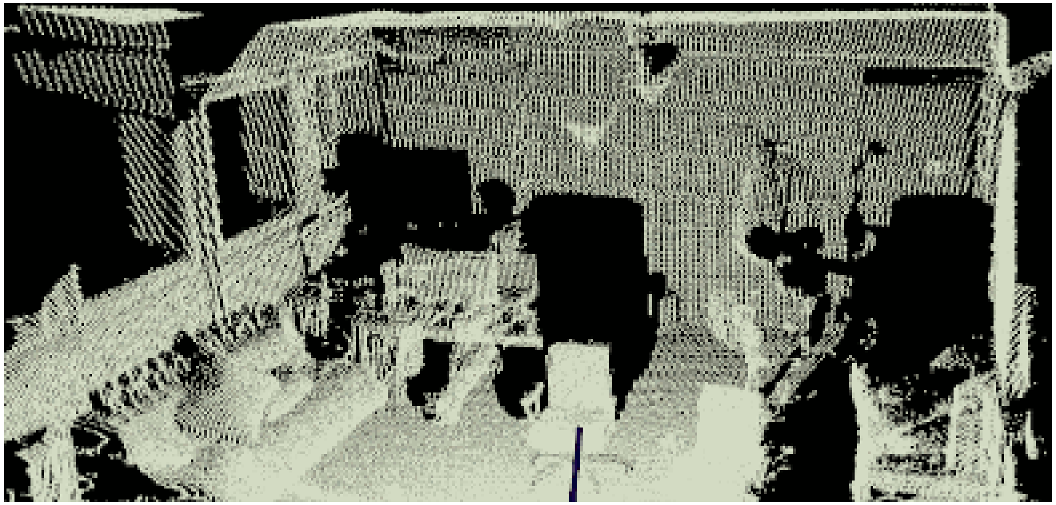
- **Shanahan 2002,2004**
  - Using a logic program as a relational representation for 3D objects in 2D line drawings
  - Using abduction for object recognition
- **Bo et.al 2011**
  - Local features



Explore, map and find victims



How does the world look like?



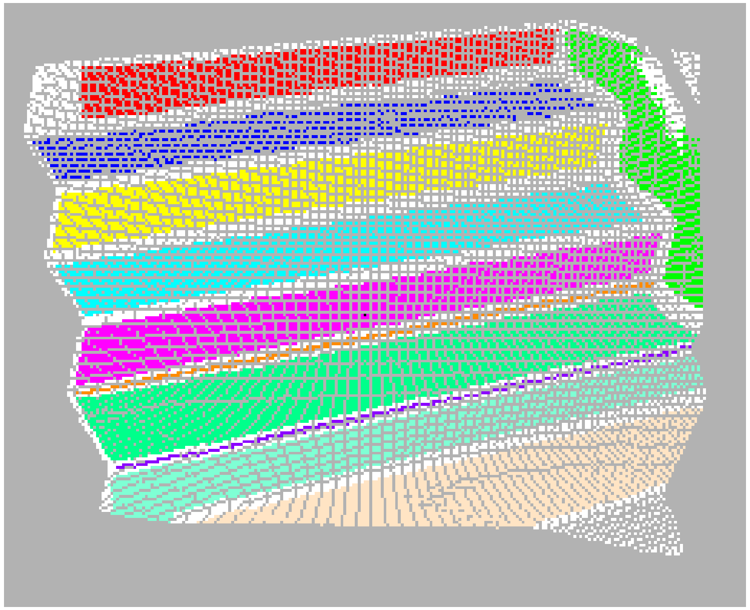
UNSW AUSTRALIA

Summary

- **Segmentation**
  - Fit planes to surfaces
  - ... may also use other geometric objects
- **Feature extraction**
  - Extract features of planes
  - Find relations between planes
- **Training**
  - Label set of planes as belonging to an example of an object class
- **Learning Evaluation**
  - 10-fold cross validation

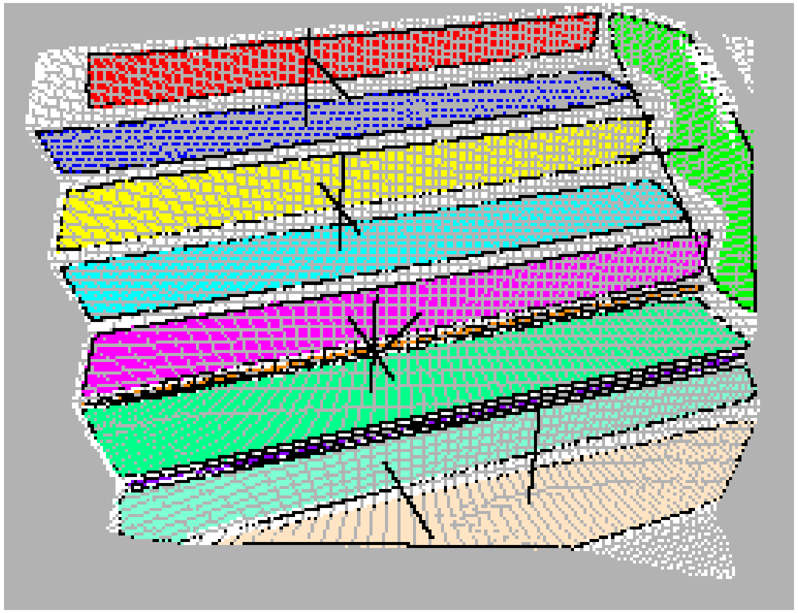
Segmentation

- Point cloud segmentation
- Using Planes as Primitives



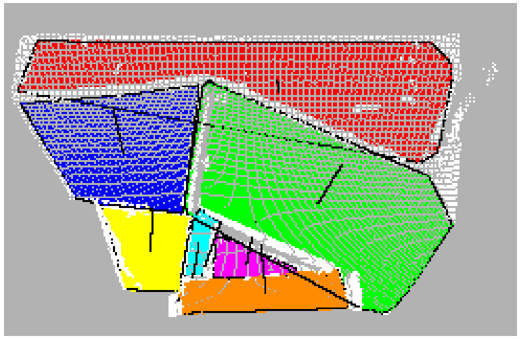
Segmentation

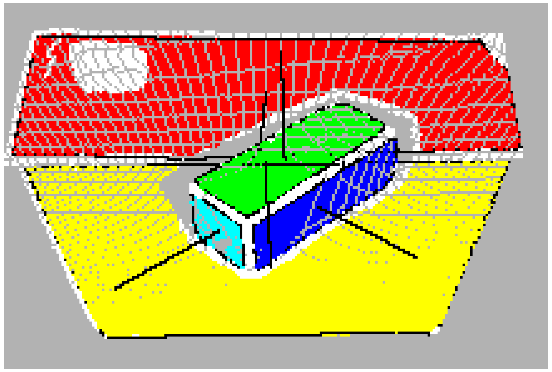
- Point cloud segmentation
- Using Planes as Primitives
- Represent each region’s boundary by a convex hull
- Using Plane’s normal vector for orientation



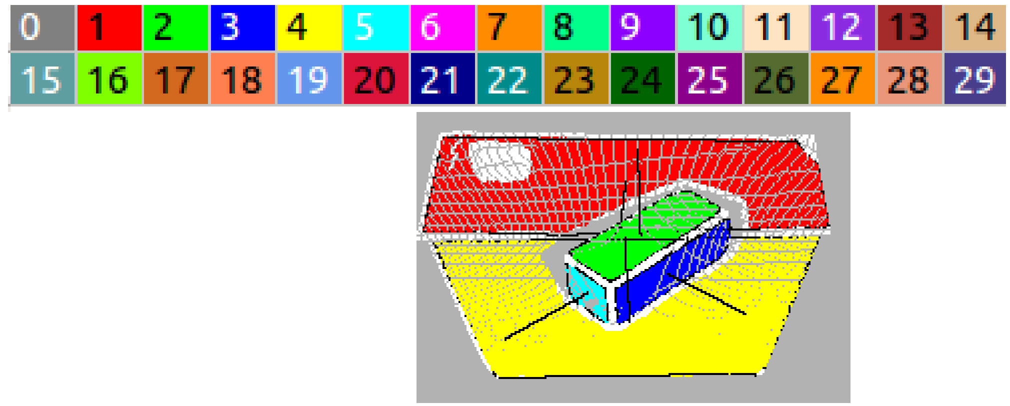
Segmentation

- More example



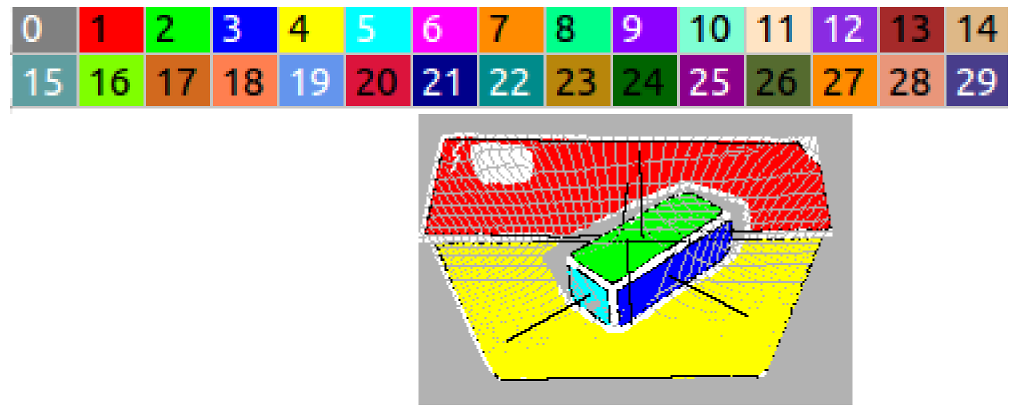


Feature extraction



- Two sets of features
  - Properties of individual planes
  - Relationships between pairs of planes
- Represented as PROLOG predicates

Feature extraction



plane(pl1). *plane(pl2).*

plane(pl3). *plane(pl4).* *plane(pl5).*

distributed_along(pl1,axisX). *distributed_along(pl2,axisX).*

distributed_along(pl3,axisX). *distributed_along(pl4,axisX).*

distributed_along(pl5,axisY).

Feature extractions

- **Convex Hull Ratio**

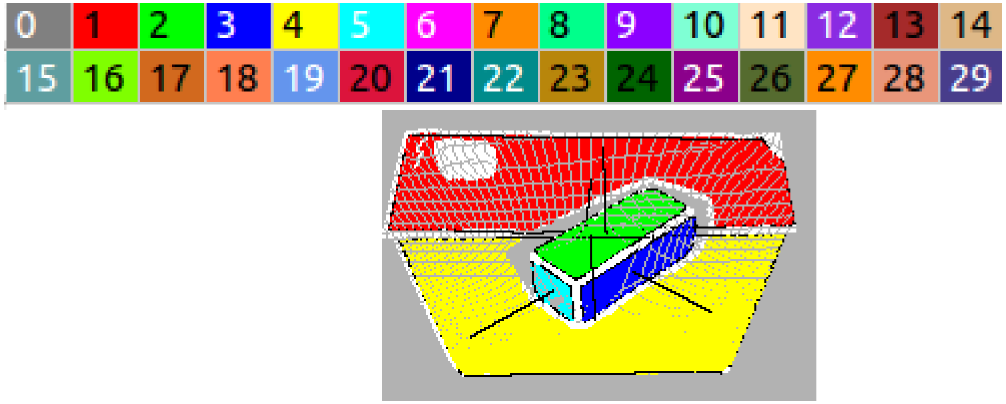
ch_ratio(pl1,'4.0±0.25').

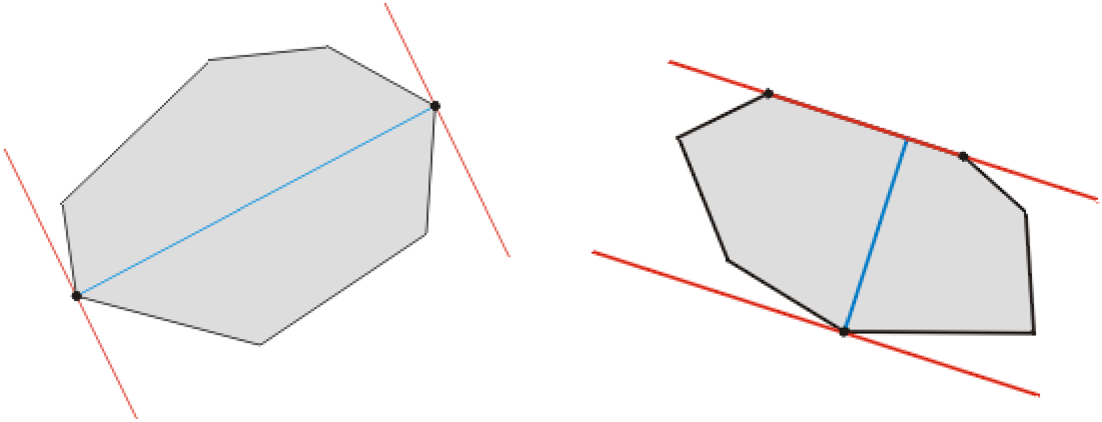
ch_ratio(pl2,'2.5±0.25').

ch_ratio(pl3,'3.5±0.25').

ch_ratio(pl4,'2.0±0.25').

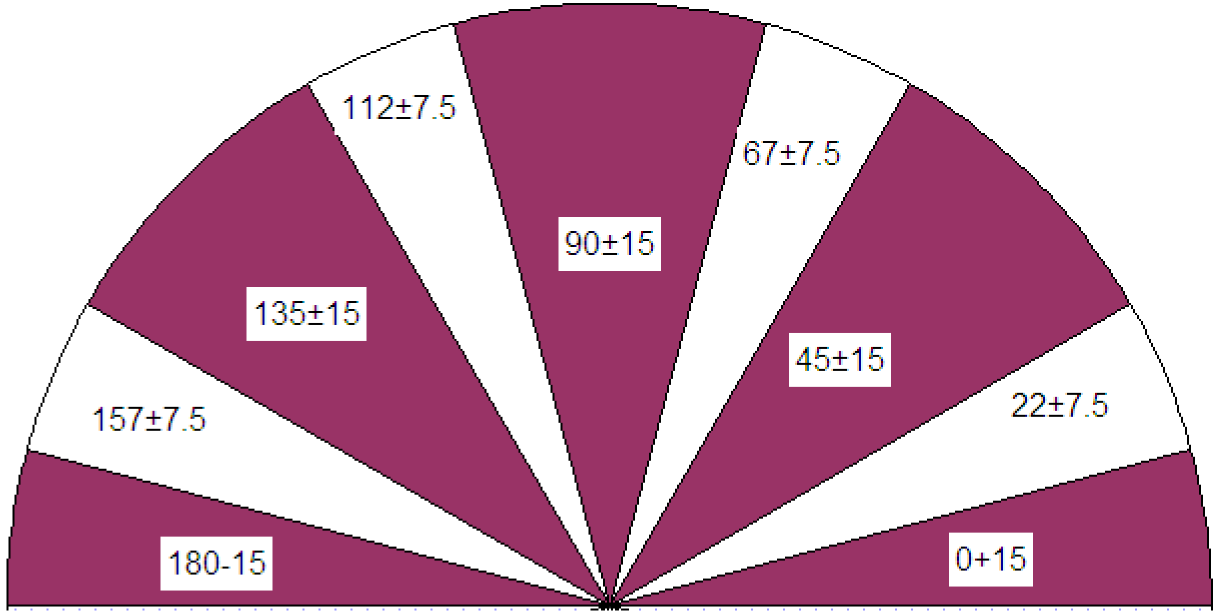
ch_ratio(pl5,'1.5±0.25').





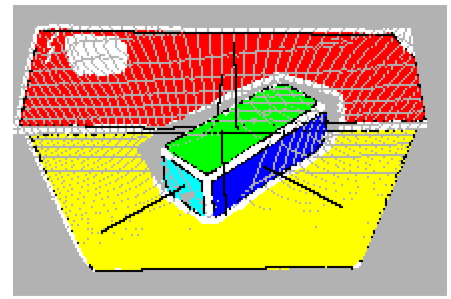
Binning

- Angle bins



Feature extraction

- **Region's normal vector in spherical coordinates**



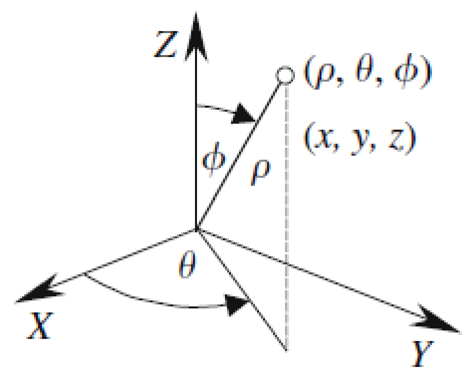
normal_spherical_theta(pl1,'-90±15').

normal_spherical_phi(pl1,'135±15').

…

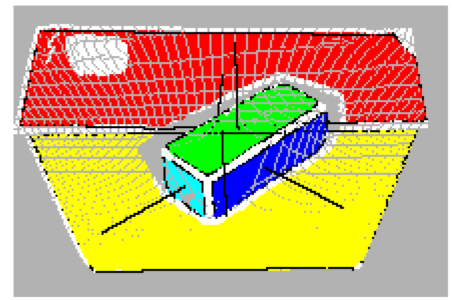
normal_spherical_theta(pl5,'-135±15').

normal_spherical_phi(pl5,'112±15').



Feature extraction

- **Angle between two regions**



angle(pl1,pl2,'90±15').
angle(pl1,pl3,'45±15').
angle(pl1,pl4,'90±15').
angle(pl1,pl5,'45±15').
angle(pl2,pl3,'90±15').
angle(pl2,pl4,'0±15').
angle(pl2,pl5,'90±15').
angle(pl3,pl4,'90±15').
angle(pl3,pl5,'90±15').
angle(pl4,pl5,'90±15').

Learning Object classes

- Object classes (positive/negative examples)
  - Step 199/731
  - Wall 105/819
  - Box 144/780
  - Pitch/roll ramp 131/205

- Training by labelling
  - **Segmented** point cloud as planes
  - User interface to group extracted planes into objects
  - Label each selected set as positive example of an object class and negative example for other object classes

  Example: Staircase 4 to 7 planes for each set

Learning Object classes

staircase([pl1,pl2,pl4,pl5]).

staircase([pl2,pl4,pl5,pl7]).

staircase([pl4,pl5,pl7,pl8]).

staircase([pl5,pl7,pl8,pl10]).

staircase([pl1,pl2,pl4,pl5,pl7]).

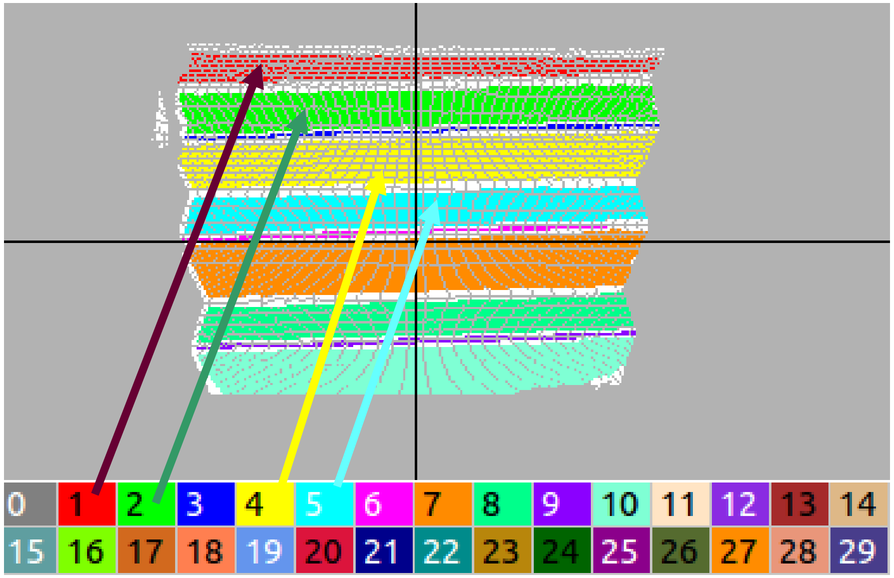
staircase([pl2,pl4,pl5,pl7,pl8]).

staircase([pl4,pl5,pl7,pl8,pl10]).

staircase([pl1,pl2,pl4,pl5,pl7,pl8]).

staircase([pl2,pl4,pl5,pl7,pl8,pl10]).

staircase([pl1,pl2,pl4,pl5,pl7,pl8,pl10]).



Learning Object classes

- Positive and negative example for each object class
- The result of labelling as PROLOG predicates
- Using **ALEPH** to construct one classifier for each type of object

Description of a Staircase

Staircase(PLANESET_B) :-
member(C, PLANESET_B),
member(D, PLANESET_B),
member(E, PLANESET_B),
angle(D, C, '0±15'),
angle(E, D, '90±15'),
angle(E, C, '90±15'),
distributed_along(E, axisX).

Staircase(PLANESET_B) :-
member(C, PLANESET_B),
member(D, PLANESET_B),
angle(D, C, '0±15'),
member(E, PLANESET_B),
member(F, PLANESET_B),
angle(F, D, '0±15'),
angle(F, C, '0±15'),
dr_xy(E, F, south).

Staircase(PLANESET_B) :-
n_of_parts(PLANESET_B, 4),
member(C, PLANESET_B),
distributed_along(C, axisX).

Cumulative Learning

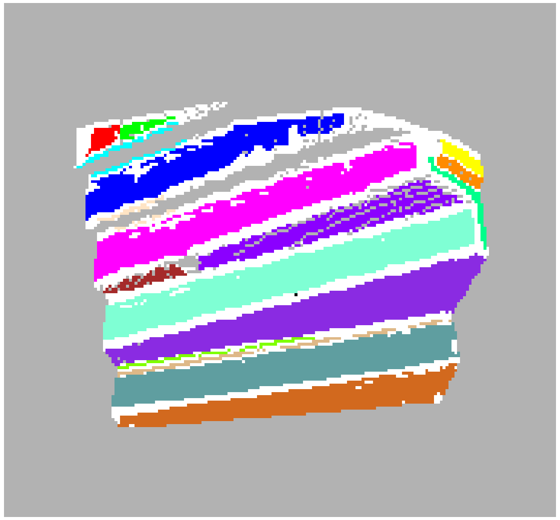
```
staircase(PlaneSet) :-
    member(C, PlaneSet),
    ratio_xz(C, '5.0±0.25'),
    subset(Set1, PlaneSet),
    subset(Set2, PlaneSet),
    step(Set1),
    step(Set2),
    intersect(Set1, Set2).
```

Evaluation

Table 1. Results for 10-fold cross validation

Object	No. positive	No. negative	Accuracy =(TP+TN)/N	Error rate =(FP+FN)/N	Precision =TP/(TP+FP)	Recall =TP/(TP+FN)
Step	199	731	0.9645	0.0355	0.9368	0.8945
Staircase	241	665	0.9956	0.0044	0.9917	0.9917
Wall	105	819	0.9881	0.0119	0.9608	0.9333
Box	144	780	0.9741	0.0259	0.9618	0.8690
Pitch/roll ramp	131	205	0.9464	0.0536	0.9520	0.9084

Evaluation on New data, New camera



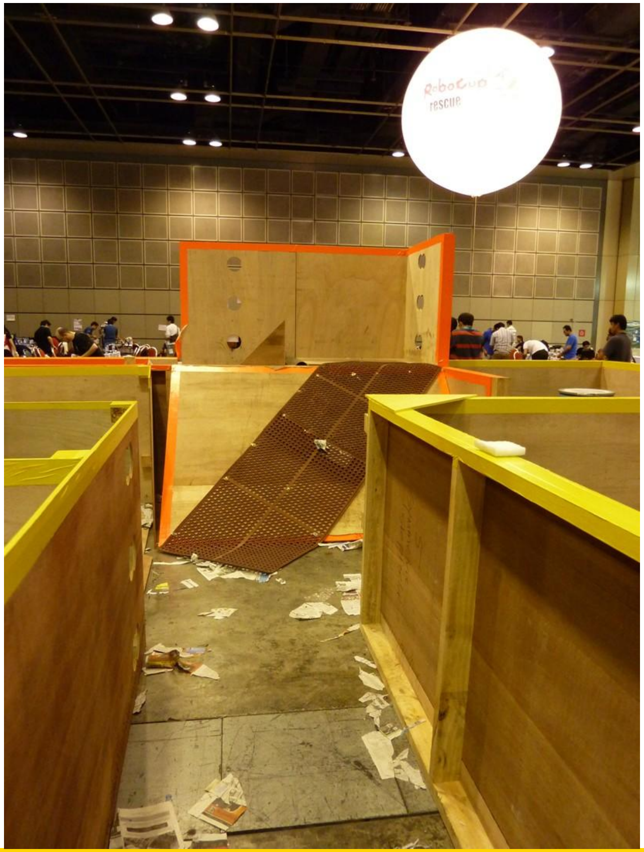
- Spiral stairs

  950 more positive example

  accuracy: 99% (sampled from one staircase over several floors)

Future Work

- Noise reduction for better segmentation
- Greater variety of objects like barrels and ramps
- Learning bins (angles bin and ratio bins)



Future Work

- Noise reduction for better segmentation
- Greater variety of objects like barrels and ramps
- Learning bins (angles bin and ratio bins)
- more primitive especially for non-flat surfaces
- Unsupervised learning

Questions

?