

3D Vision



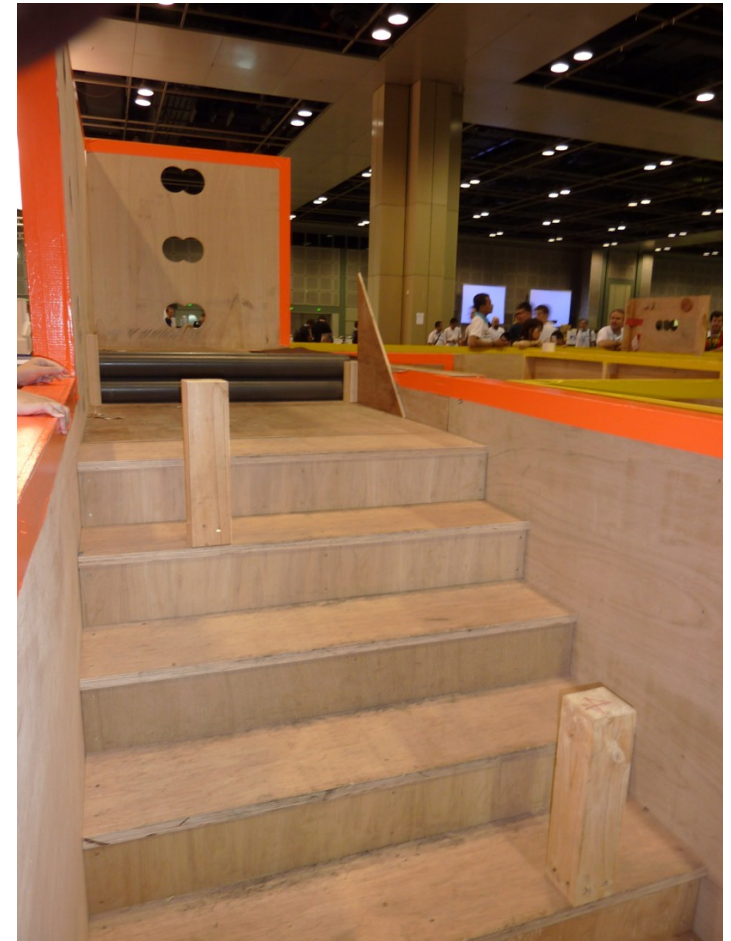
3D Cameras

- Stereo used to be the only option
- Now, RGB-D cameras are easier (at least for indoor)
- Stereo still use lower power

Urban Search and Rescue



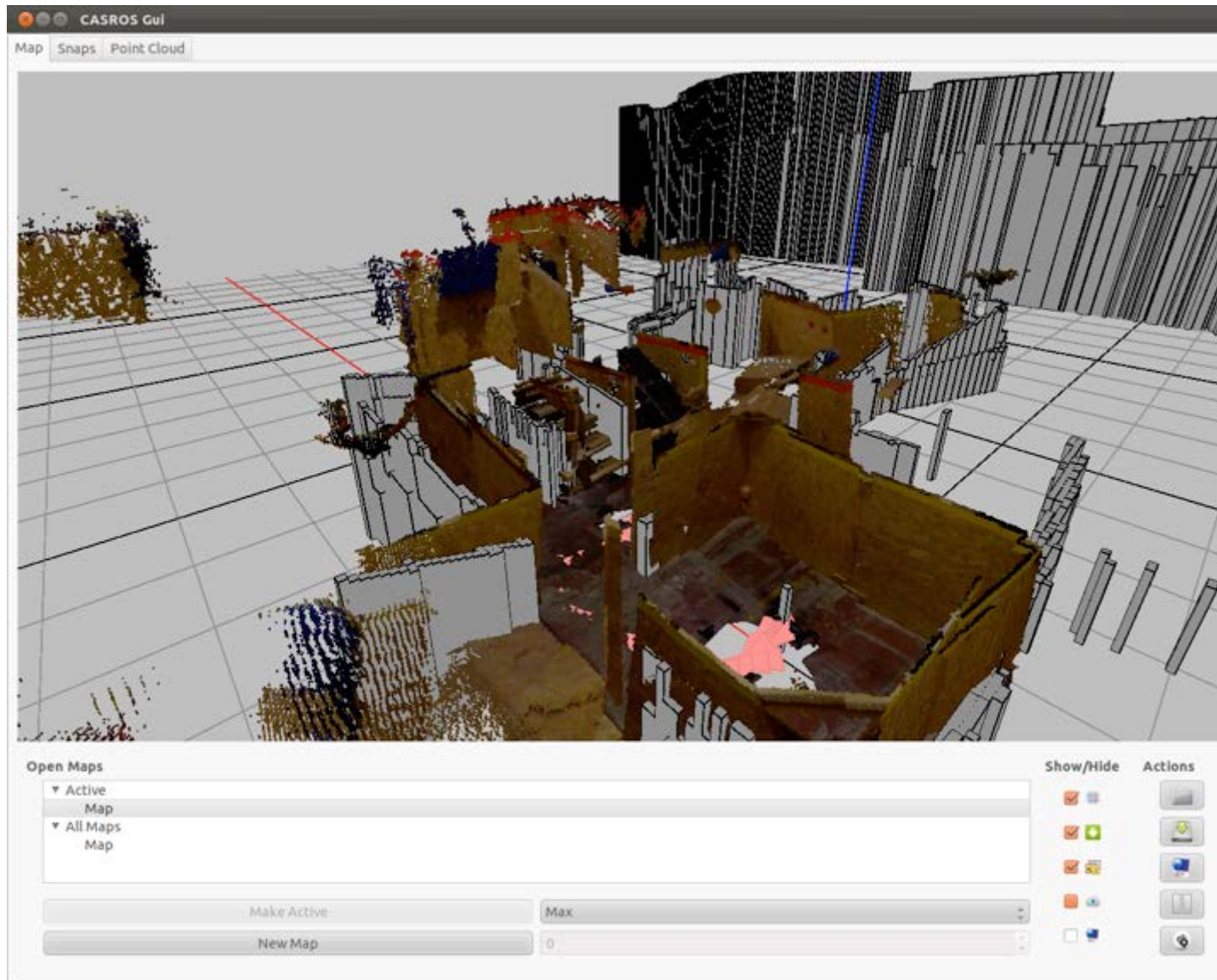
Rescue Arena



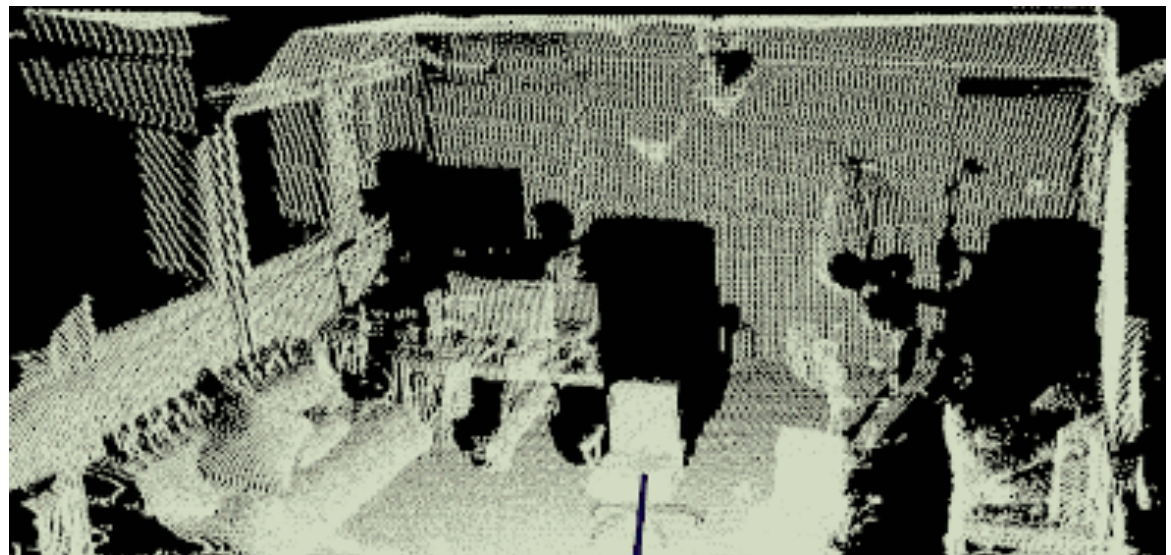
Rescue Arena



Explore, map and find victims

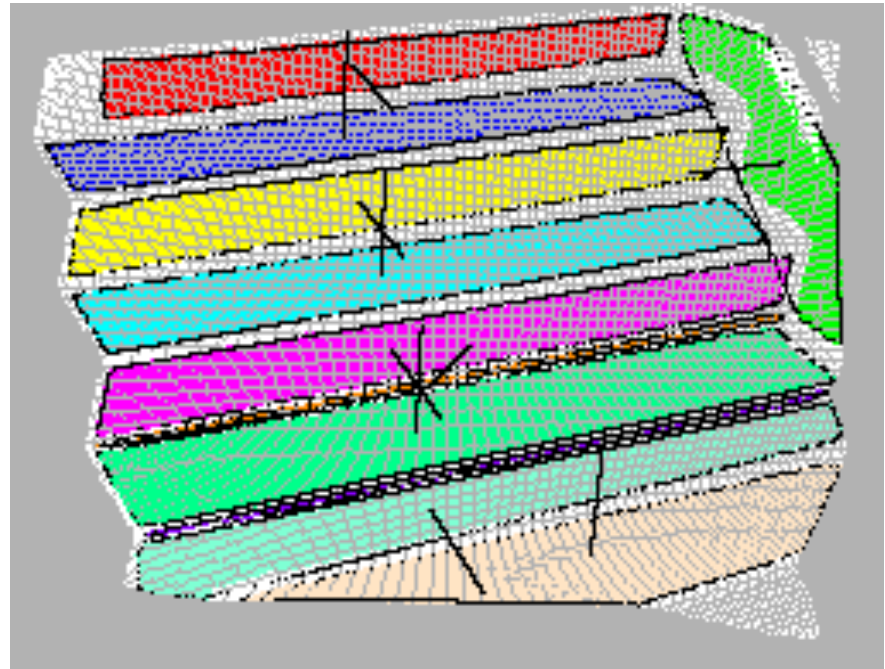


What does the world look like?



Segmentation

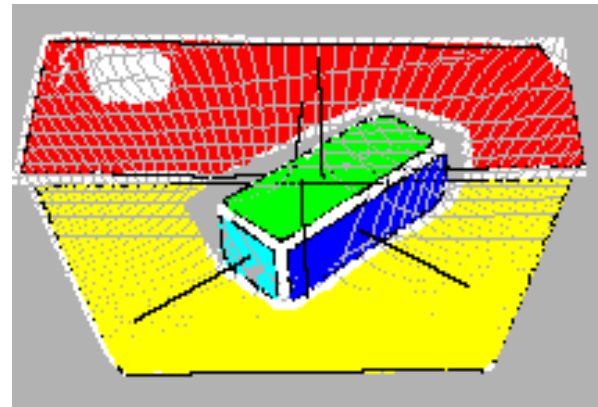
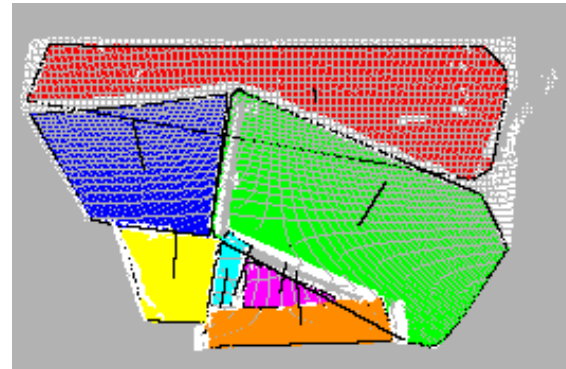
- Point cloud segmentation
- Using planes as primitives
- Represent each region's boundary by a convex hull
- Using plane's normal vector for orientation



Normals Vectors

- Point P_1 has 8 neighbours in depth plane
- Form eight triangles:
 $P_1P_2P_3, P_1P_3P_4, \dots, P_1P_8P_9, P_1P_9P_2$
- Calculate normal vector from the cross product of P_1P_i and P_1P_j
- The normal vector for P_1 is average of eight normal vectors, and normalising the result

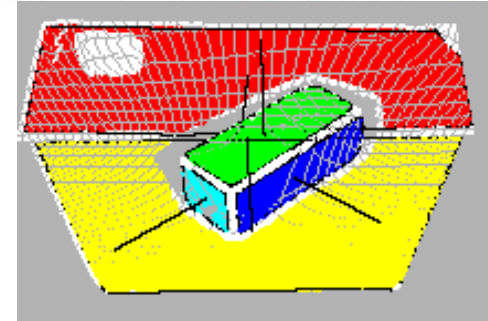
Segmentation



Segmentation

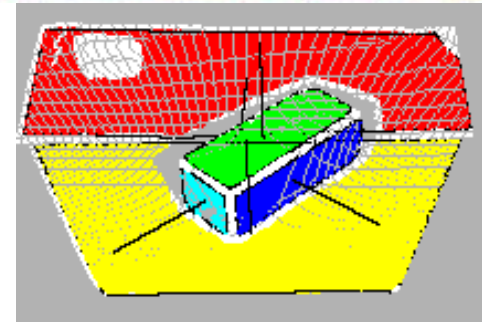
- Form regions by clustered neighbouring points that have (almost) the same normal vectors
- Find convex hull around points in the same region

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).

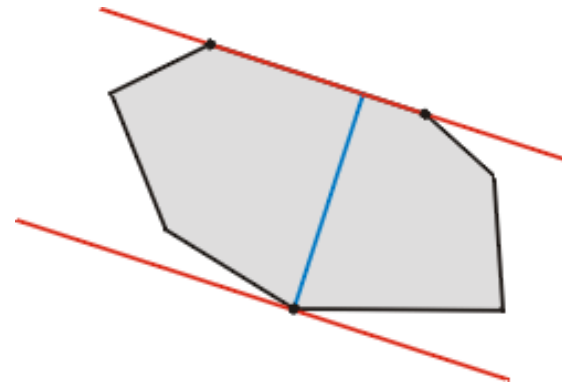
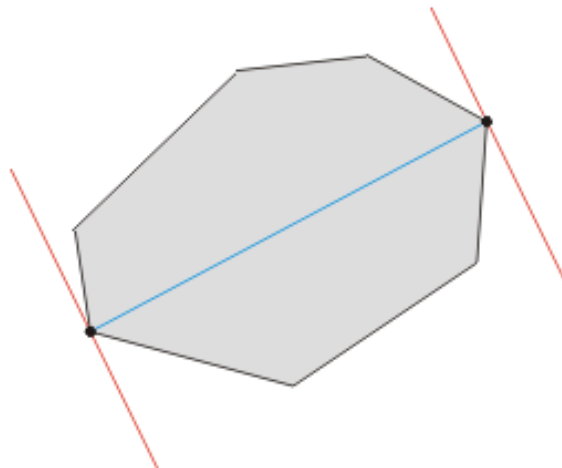
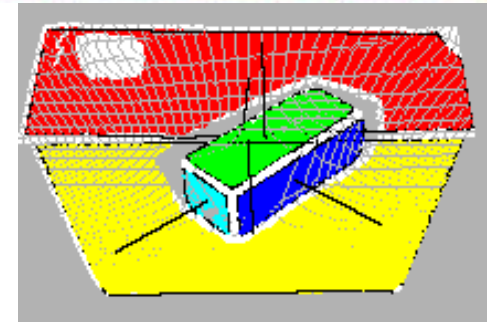
distributed_along(pl5,axisY).

Feature Extraction

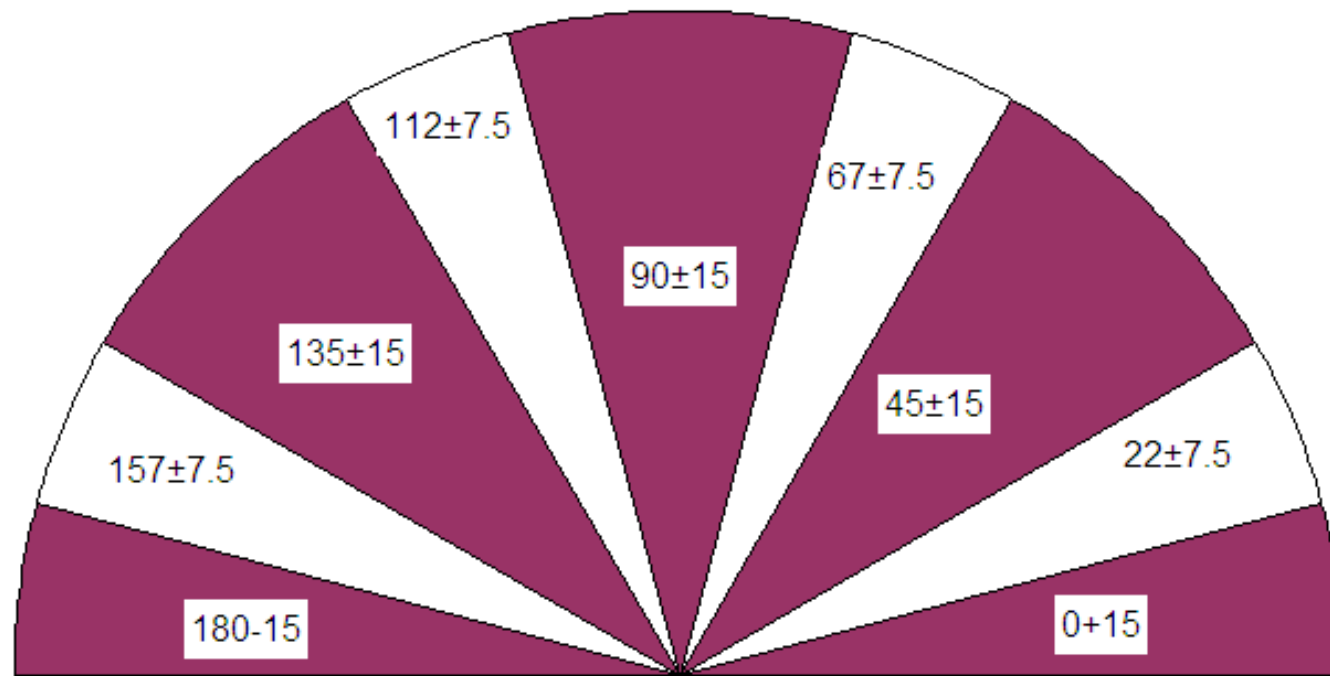
Convex Hull Ratio

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').

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |

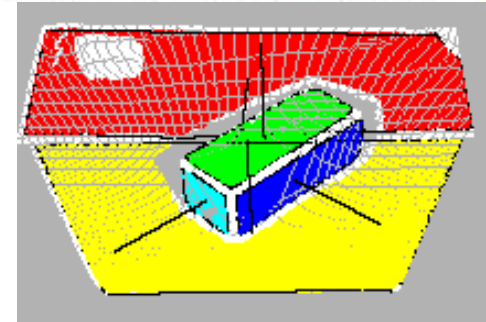


Angle Bins



Feature Extraction

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |



- Region's normal vector in spherical coordinates

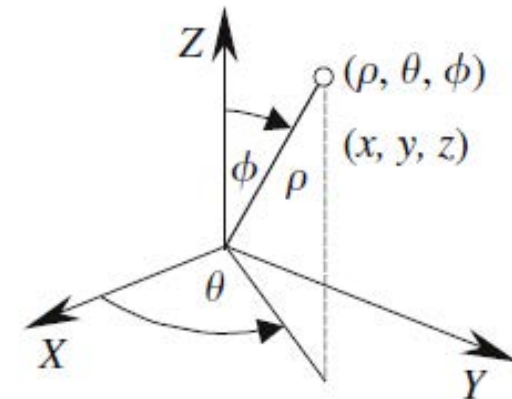
$\text{normal_spherical_theta}(pl1, '-90 \pm 15')$.

$\text{normal_spherical_phi}(pl1, '135 \pm 15')$.

...

$\text{normal_spherical_theta}(pl5, '-135 \pm 15')$.

$\text{normal_spherical_phi}(pl5, '112 \pm 15')$.



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')`.

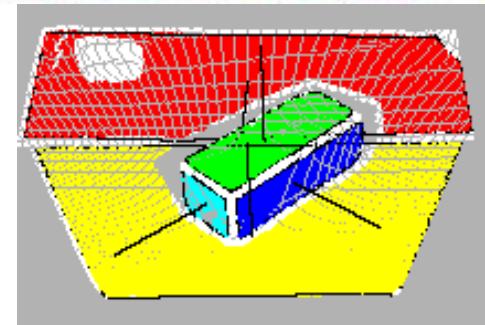
`angle(pl2,pl4,'0±15')`.

`angle(pl2,pl5,'90±15')`.

`angle(pl3,pl4,'90±15')`.

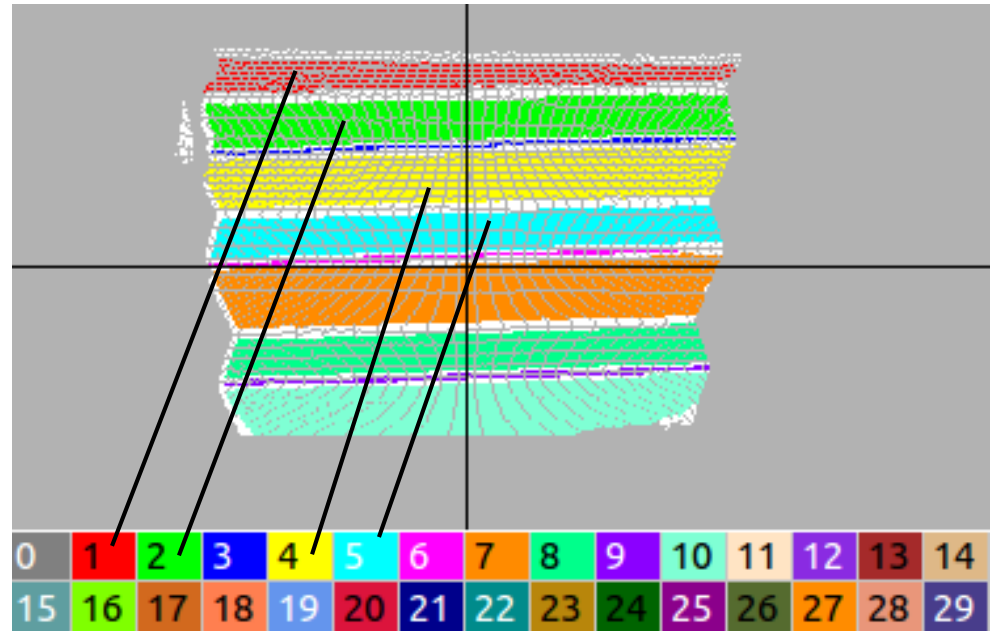
`angle(pl3,pl5,'90±15')`.

`angle(pl4,pl5,'90±15')`.



Learning Object Classes

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 relational learning system to construct a classifier for each type of object

Description of a Staircase

staircase(B) : –

 p_a(B).

staircase([X, Y, Z|B]) : –

 p_a([X, Y, Z]),

 staircase([Z|B]).

p_a(B) : –

 member(X, B),

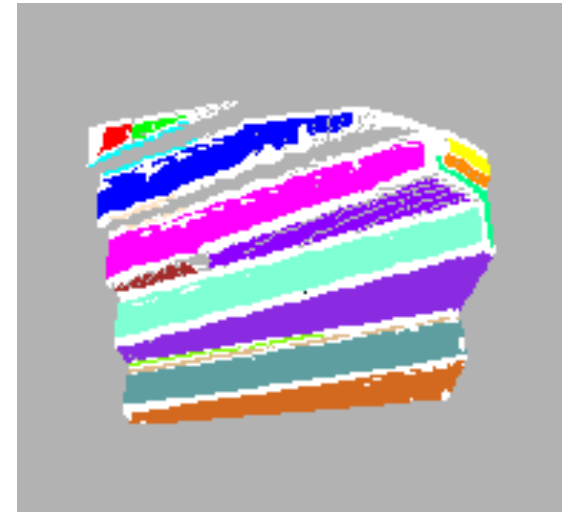
 member(Y, B),

 angle(X, Y, '90±15'),

 member(Z, B),

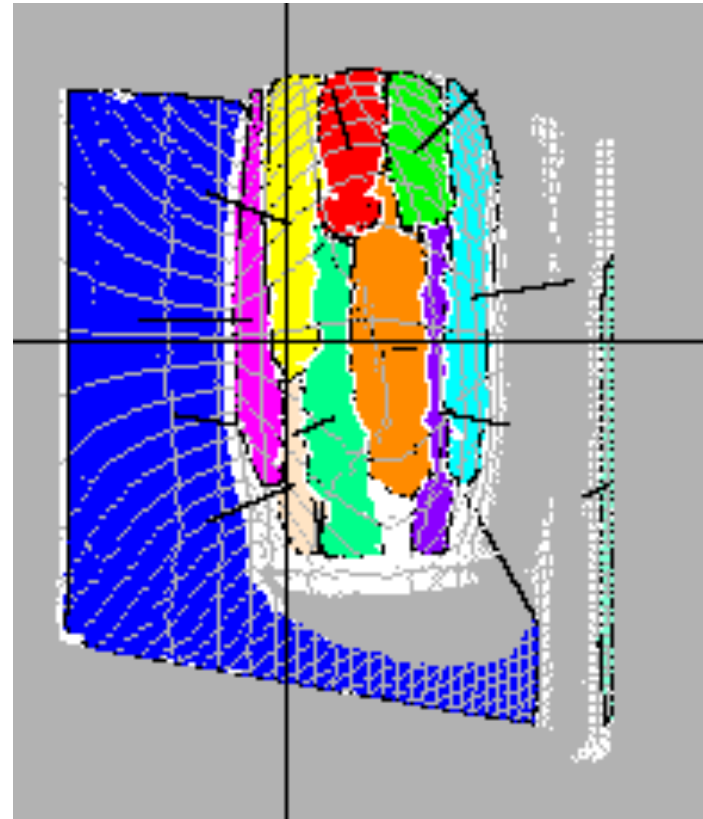
 angle(X, Z, '0±15')

New data, New camera



- Spiral stairs
- 950 more positive examples
- accuracy: 99% (sampled from one staircase over several floors)

Non-planar surfaces



Structural relationships

