An Object-based World Model and its Uses

Mac Mason

Bhaskara Marthi
“I just want a robot that keeps track of my stuff”
- Aditi Nabar (spouse)
Pipeline

Kinect → Point Clouds → Plane Finder → Planes in frame → Plane Data Association → Object Data Association

Clustering → Points above planes → Object clusters → Object Data Association

DB

Set of planes
Set of objects + metadata
Implementation

- PR2 + Kinect
- ROS, PCL, Nav stack
- MongoDB for storing info
  - Binary blobs + metadata (color, shape, ...)
  - Partition by run
  - Efficient queries over location, attributes
Data Association

- Match within runs based on spatial overlap
- Don’t commit to matches across runs
- Discover likely matches at query time
Attribute perception

• No object categories or instances

• Instead, extract descriptive features

Color: Red
Shape: Cylinder
Diameter: 0.1m
Sensors to symbols

- Sensor level
- Point clouds, images,
- Normals, SIFT features, ...
- Symbolic level: objects, categories, relations
- How to connect?
Bridging the gap

[Galindo et al, 2008]
Reify Geometry

- Surfaces, clusters, cluttered regions, ...
- Color, shape, topology, ...
- A lot can be done at this level!
Conclusions

• Object-based world model
• Allows
  • Change detection
  • Semantic querying
• Robust to noisy perception
Conclusions

- Object-based world model
- Allows
  - Change detection
  - Semantic querying
- Robust to noisy perception

Thank you!