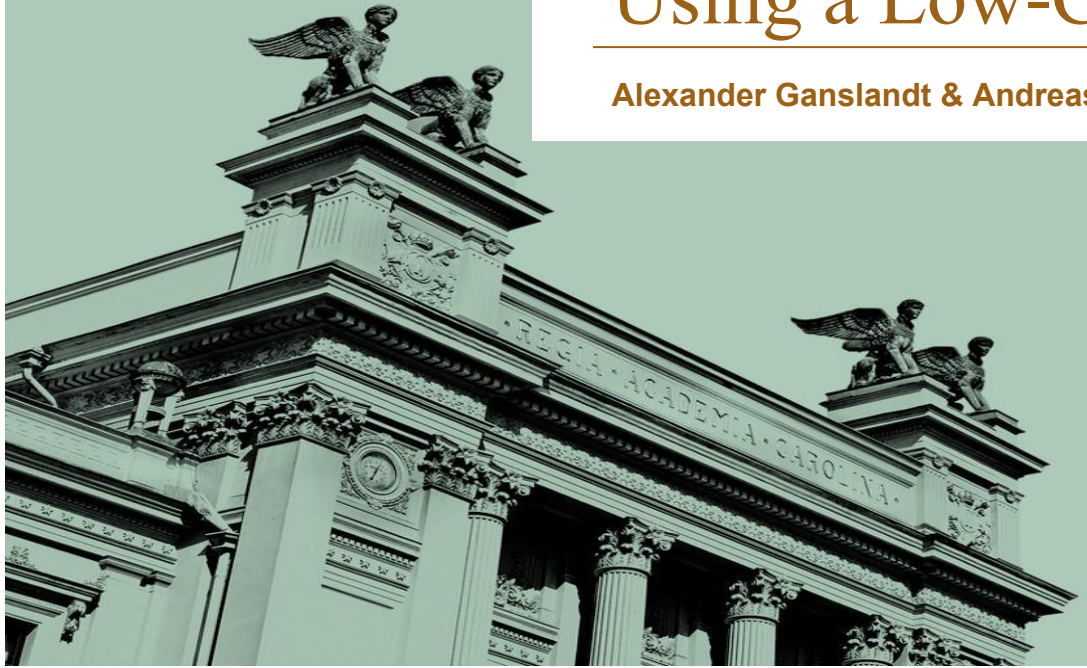
LUND UNIVERSITY

Object Pose Estimation in Robotics Using a Low-Cost RGB-D Camera

Alexander Ganslandt & Andreas Svensson



Background

- Picking up an object in a structured and predefined environment is no match for today’s industrial robots
- The task becomes more cumbersome when the objects are moved by e.g. a human operator
- With today's increasing demand for human-robot interaction and cooperation, the robot needs to be able to adapt to unstructured environments

LUND UNIVERSITY

Problem

- Can we estimate the pose of an object i.e. position and rotation using an RGB-D camera?
- How accurately can we measure the pose of the object?
- Can we transform the pose estimation into the robot's coordinate reference system?

Contents

- Basics
  - RGB-D Camera Introduction
  - Point Clouds and Point Cloud Library
- Project Description and Results
  - Constructing Point Cloud Models
  - Object Pose Estimation
  - Connecting to a Robot
- Future Work and Conclusion

LUND UNIVERSITY

RGB-D Camera Introduction

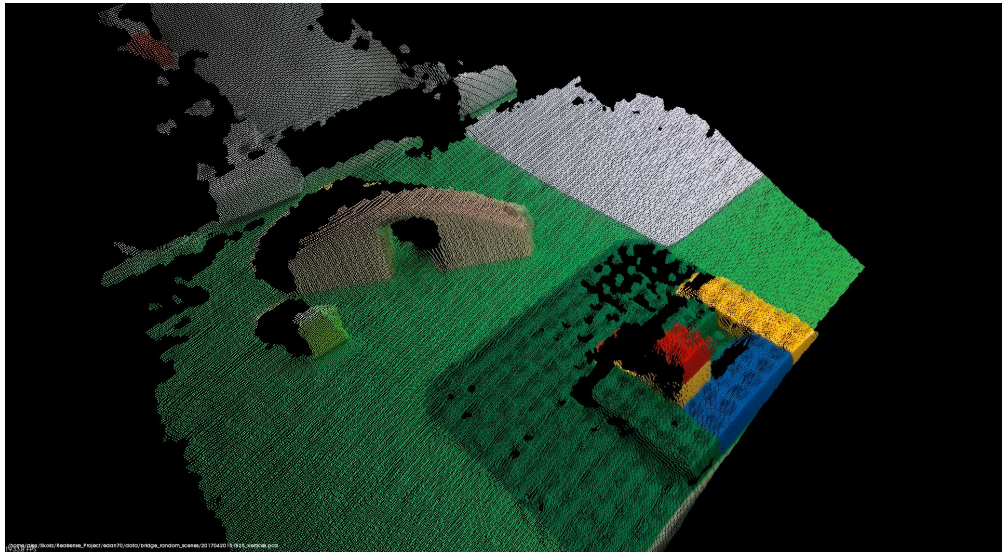
- RGB camera + Depth sensor → 2.5 D
- Time-of-Flight (laser, phase-shift)
- Passive triangulation
- Active triangulation (structured light)

Intel RealSense SR300

- Small RGB-D camera using structured light
- Low-cost (around $149)
- Uses Intel RealSense SDK with lots of examples
- Poor and incomplete documentation

LUND UNIVERSITY

Point Cloud



- Set of data points in 3D
- Can contain color information

LUND UNIVERSITY

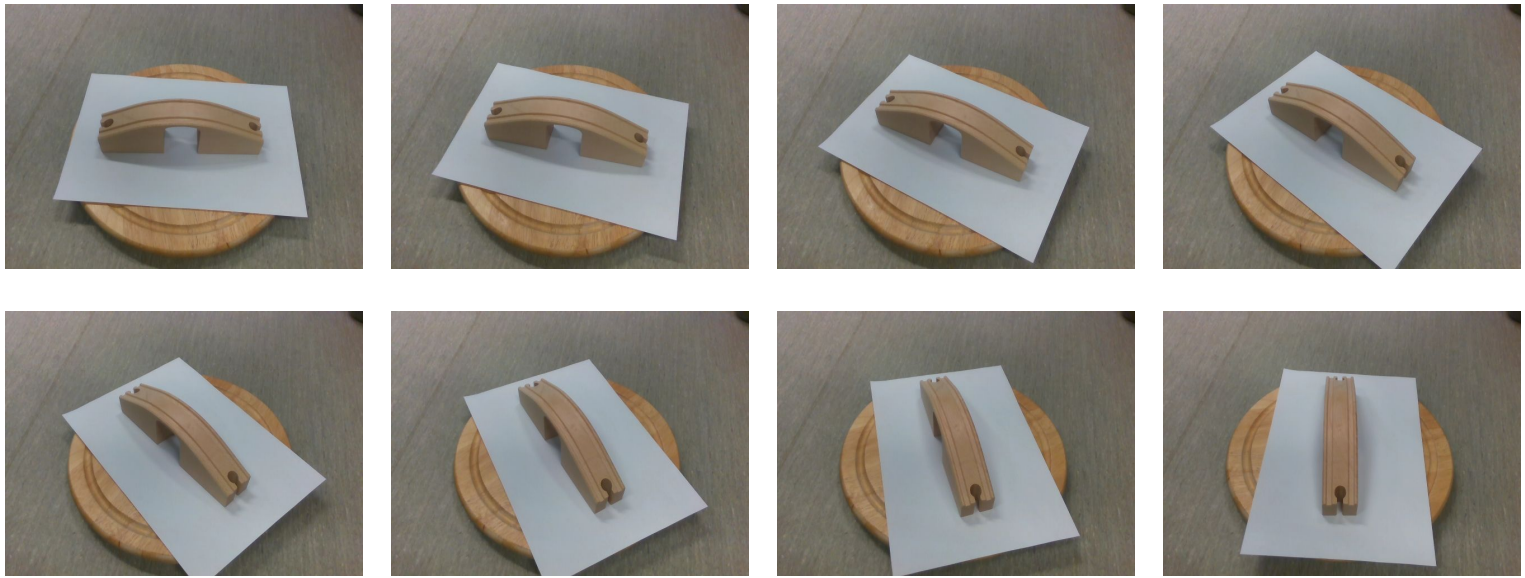
Point Cloud Library (PCL)

- Library for 2D/3D image and point cloud processing
- Large scale, open source and cross-platform
  - Well written documentation
  - Great tutorials
  - Supports many point cloud formats (PCD OBJ PLY)
- Written in C++
  - Some support for Visual Studio (2008 and 2010)

Constructing Point Cloud Models

- We need a 3D model of the object for the pose estimation
- Achieved by segmentation and merging point clouds from different views

Constructing Point Cloud Models

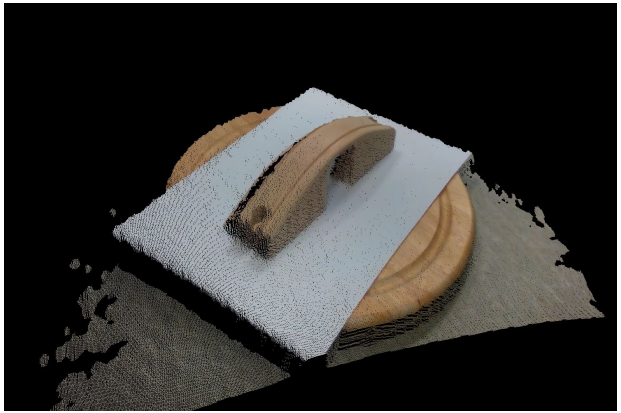


LUND UNIVERSITY

Constructing Point Cloud Models

Segmentation

- RANSAC to find and remove plane
- Euclidean Cluster Extraction to extract point cloud of object



LUND UNIVERSITY

Constructing Point Cloud Models

Random sample consensus (RANSAC)

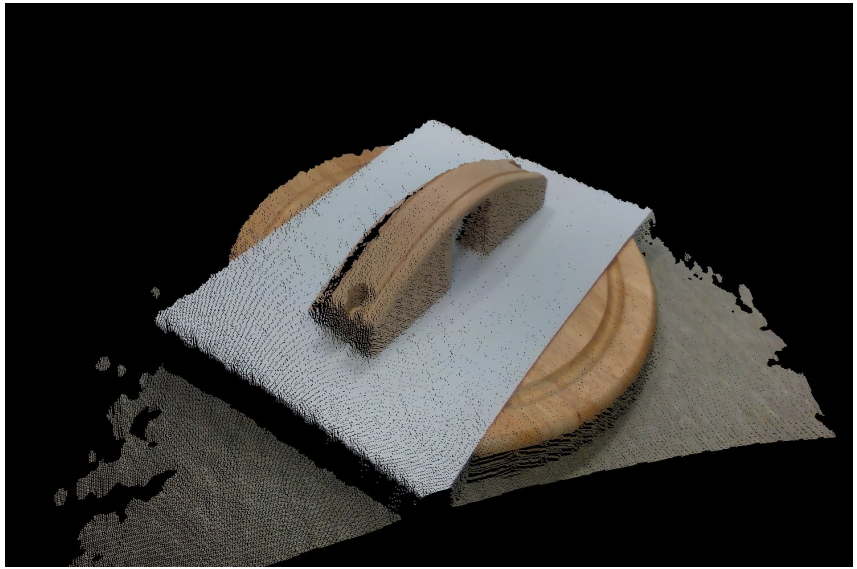
- Mainly used as an outlier detector
- RANSAC to remove a plane:
  - Equation of a plane: $ax+by+cz+d = 0$
  - Inliers are points in a close proximity of the plane
  - Find the set of inliers to the plane and remove them

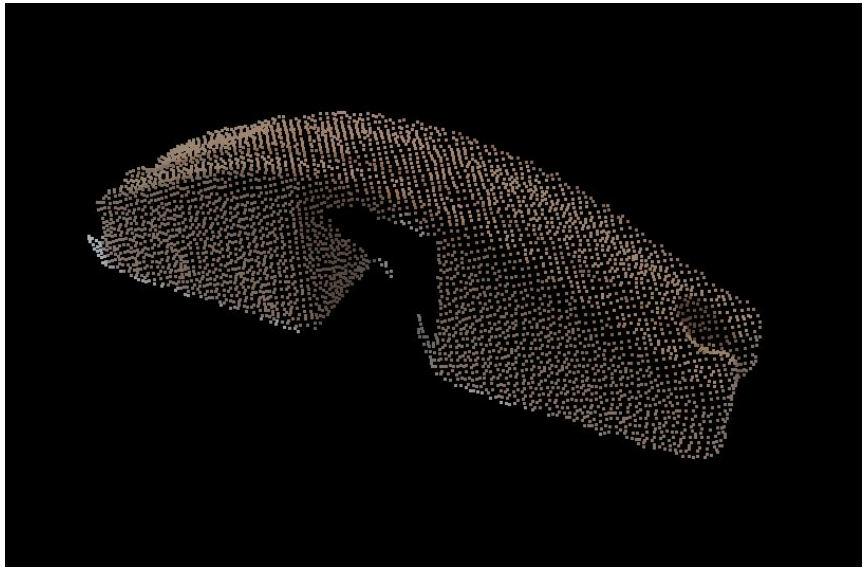
Constructing Point Cloud Models

Euclidean Cluster Extraction

- Searches for the set of neighbors of a point that are within a sphere
- Uses a Kd-tree structure for finding the nearest neighbors

Constructing Point Cloud Models





LUND UNIVERSITY

Constructing Point Cloud Models

Pairwise registration

- Two consecutive segments have different rotation and translation.
- Pairwise registration tries to find the transformation between the segments
- Once the transformation has been found the point clouds are merged and smoothed.

Constructing Point Cloud Models

Pairwise registration - Finding the transformation

- First a pose estimation algorithm is used to find a rough transformation using a heavily downsampled version of the segment (more on this later)
- Once a rough estimate of the transformation has been found an Iterative Closest Point (ICP) algorithm is used to get a better estimate using the full size of the sample set
- The point cloud in the second image is then transformed to match the point cloud in the first image

Constructing Point Cloud Models

Iterative Closest Point (ICP)

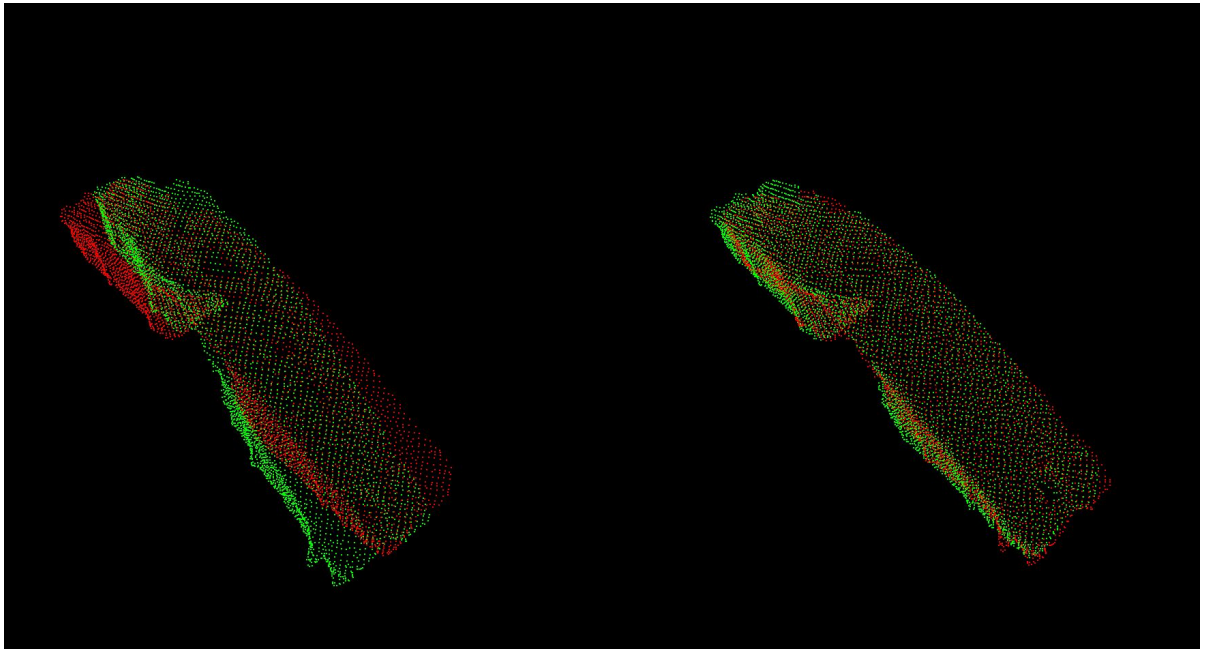
- Used to minimize the distance between two point clouds.
- Achieved by estimating a rotation ***R*** and translation ***t***
- This is done by minimizing a cost function:

$$V(R,t) = \sum_i ||x_i^{(1)} - Rx_i^{(2)} - t||^2$$

- Convergence is improved if the point clouds are initially close

LUND UNIVERSITY

Constructing Point Cloud Models



LUND UNIVERSITY

Final Point Cloud Model

LUND UNIVERSITY

Final Point Cloud Model

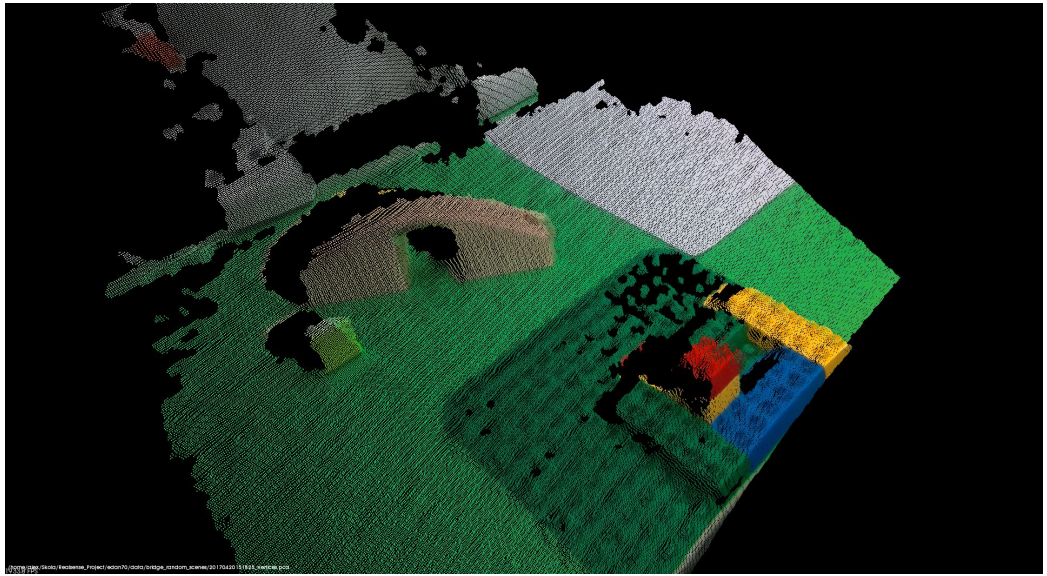
- Extracted model is far from perfect
  - Hard to achieve a true 3D model of symmetric objects
- Alternatively:
  - Create and use a CAD model of the object
  - Use 3D-scanning software (price)
  - Reconstruct the 3D image by using the RGB images
    - Extract matching features and use triangulation

LUND UNIVERSITY

Object Pose Estimation

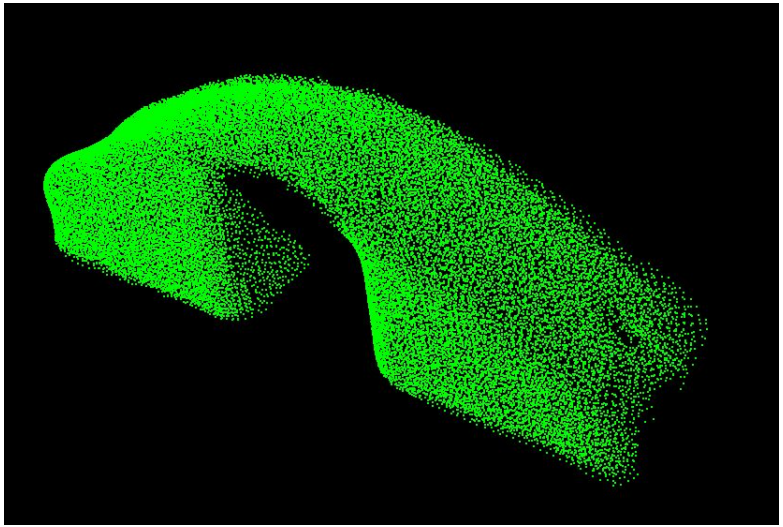
- We want to estimate 6-DoF
  - Position of object (3-DoF)
  - Rotation of object (3-DoF)

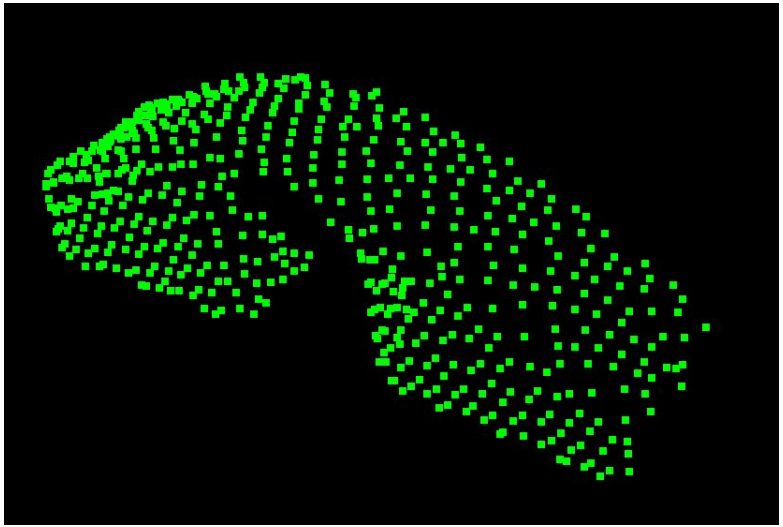
Pipeline - Input



LUND UNIVERSITY

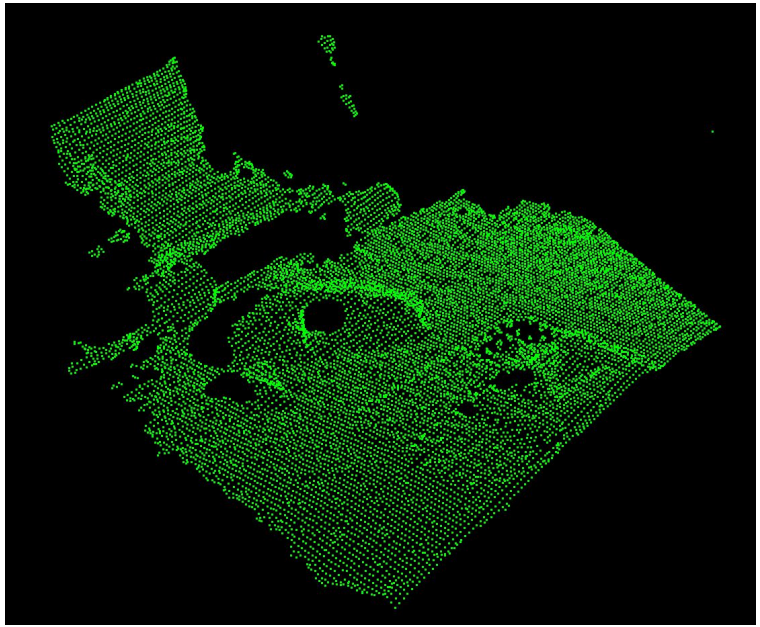
Pipeline - Downsample

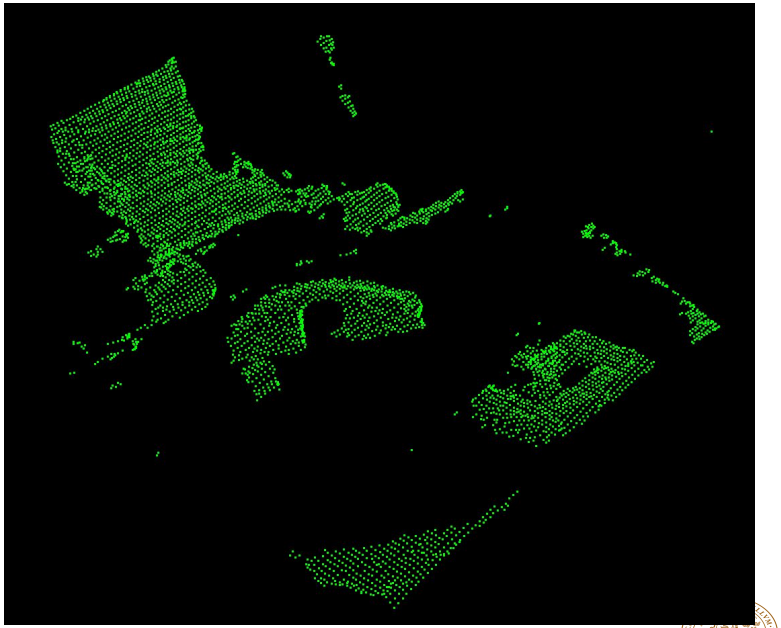




LUND UNIVERSITY

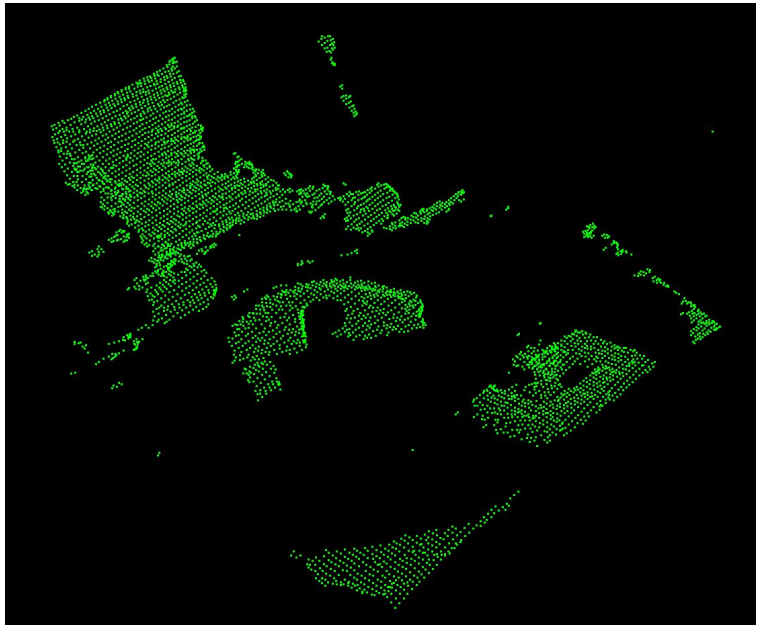
Pipeline - Remove largest plane

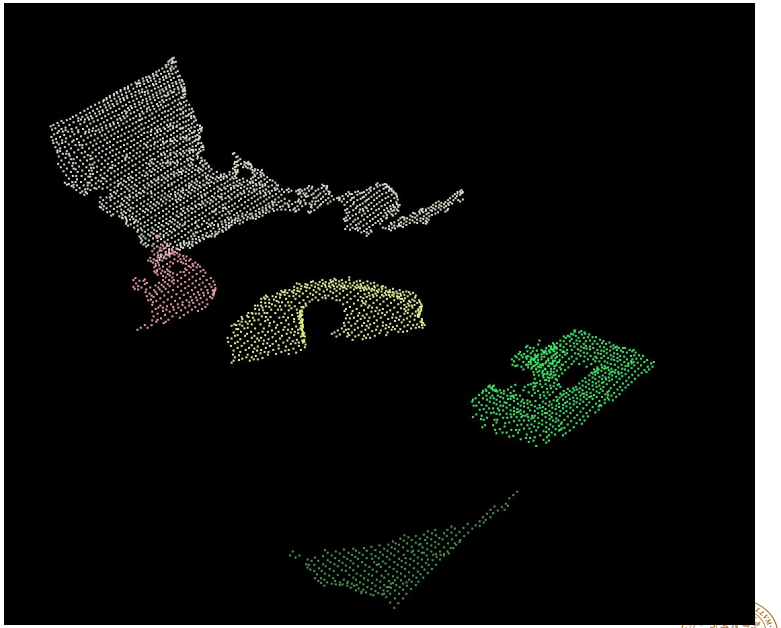




LUND UNIVERSITY

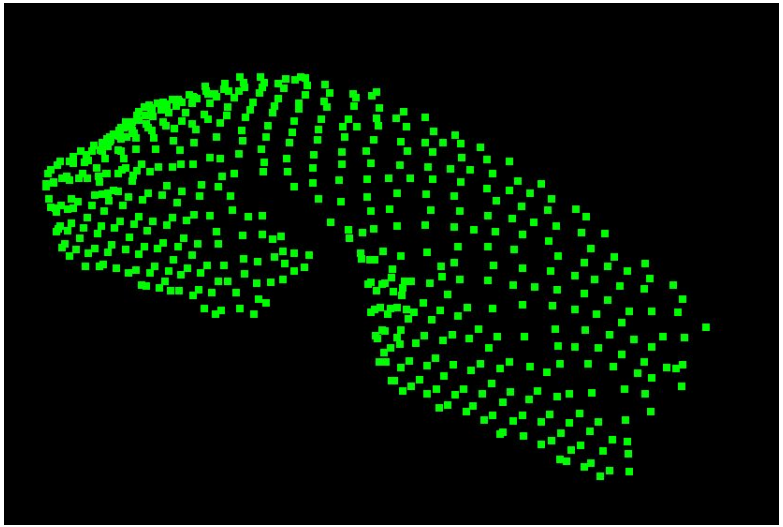
Pipeline - Extract clusters

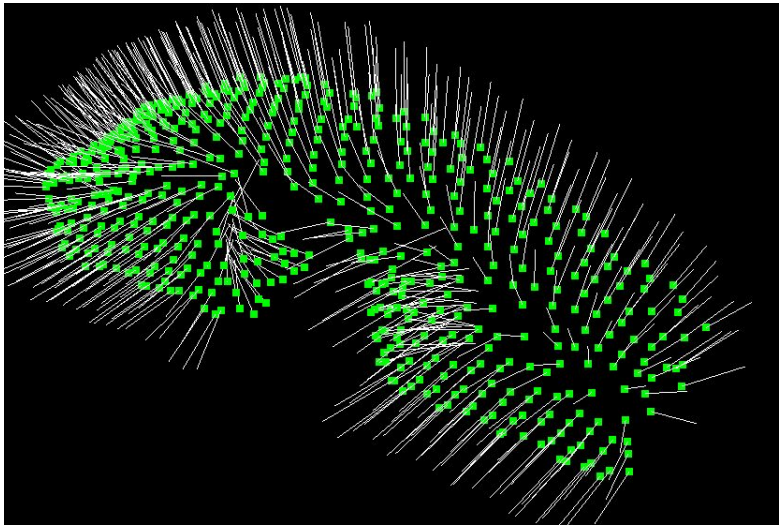




LUND UNIVERSITY

Pipeline - Estimate normals





LUND UNIVERSITY

Pipeline - Estimate features

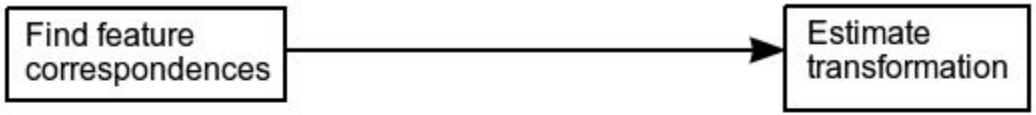
- Encode a points geometrical properties
- We use Fast Point Feature Histograms (FPFH)
  - Looks at the k-neighborhood of each point
  - Computes features based on direction of normals
- Many different point feature representations available
  - Viewpoint Feature Histogram (2010)
  - Color Point Pair Feature (2015)

Pipeline - Estimate pose

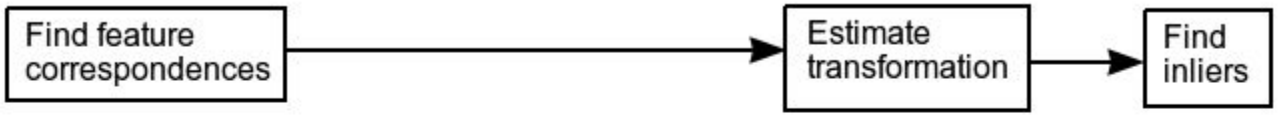
Find feature correspondences

LUND UNIVERSITY

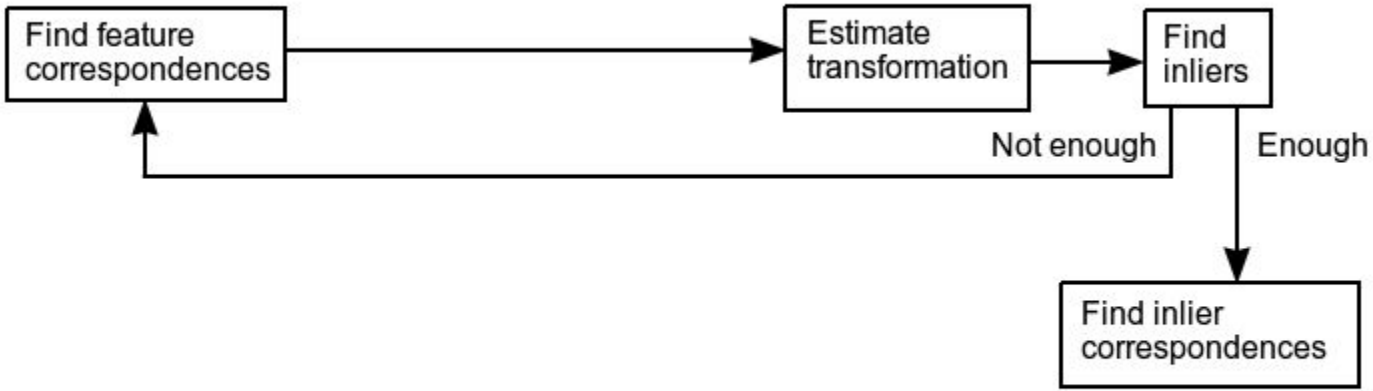
Pipeline - Estimate pose



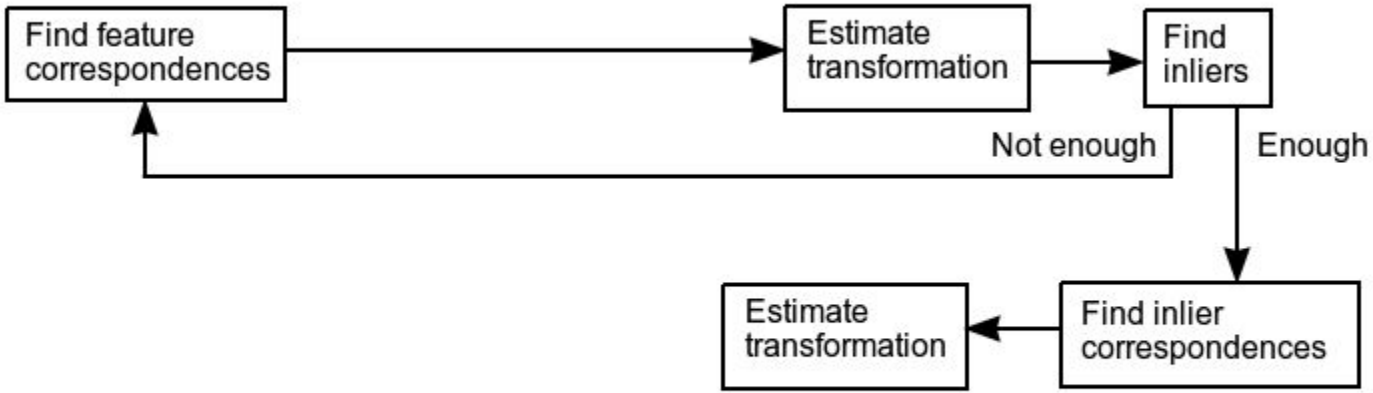
Pipeline - Estimate pose



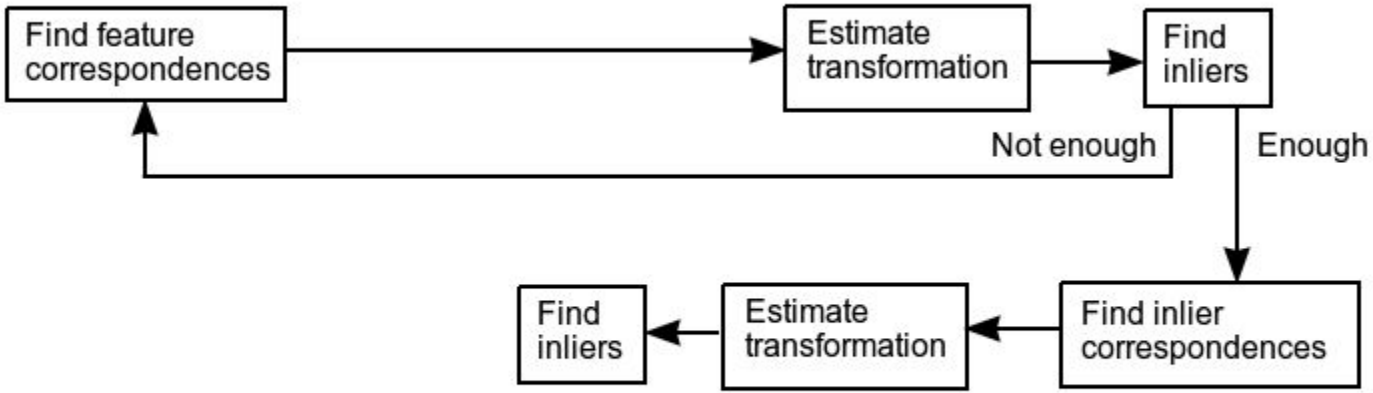
Pipeline - Estimate pose



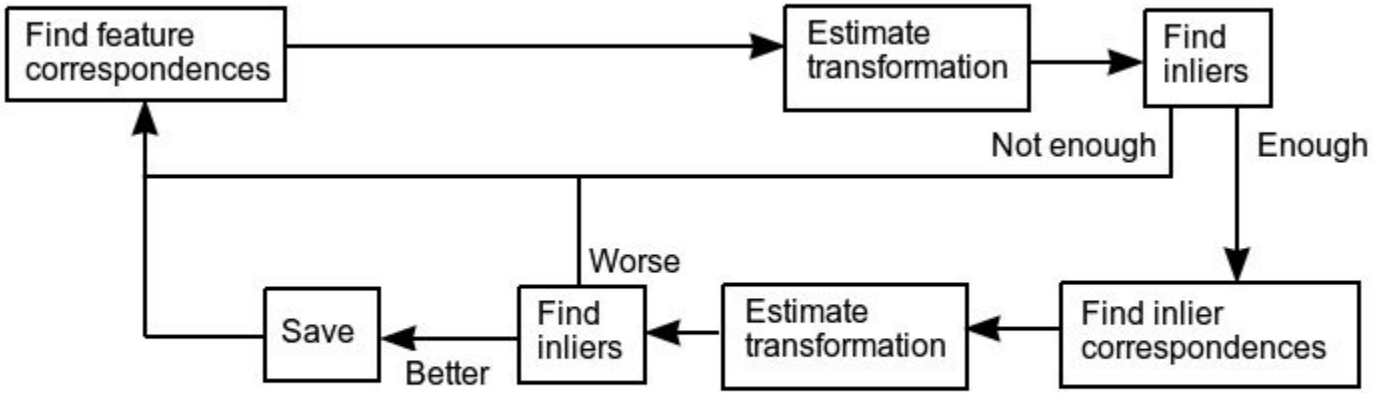
Pipeline - Estimate pose



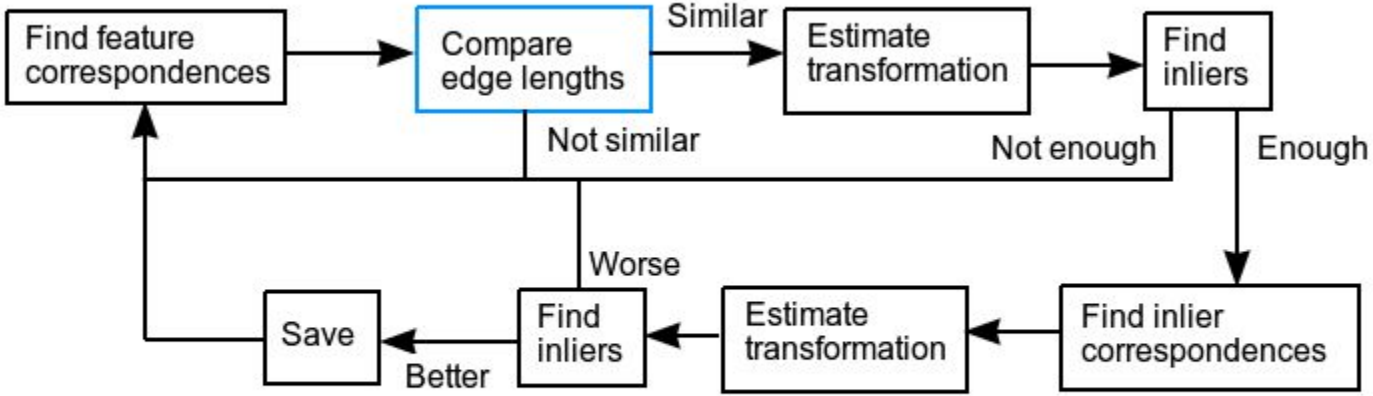
Pipeline - Estimate pose



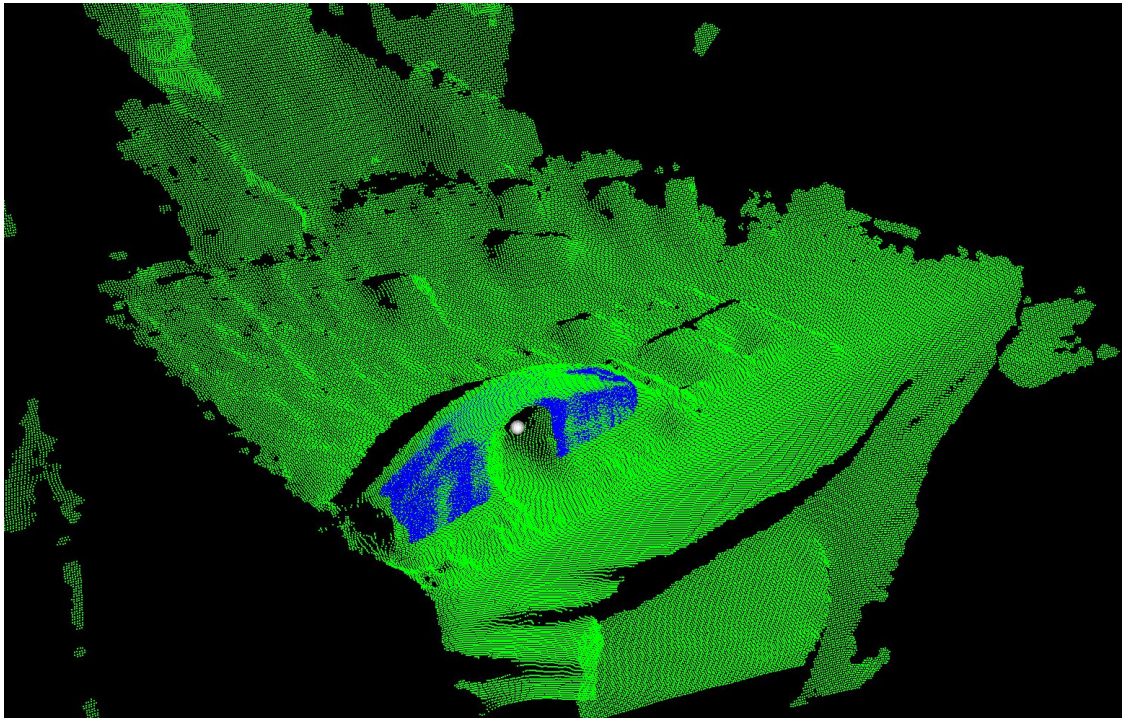
Pipeline - Estimate pose



Pipeline - Estimate pose

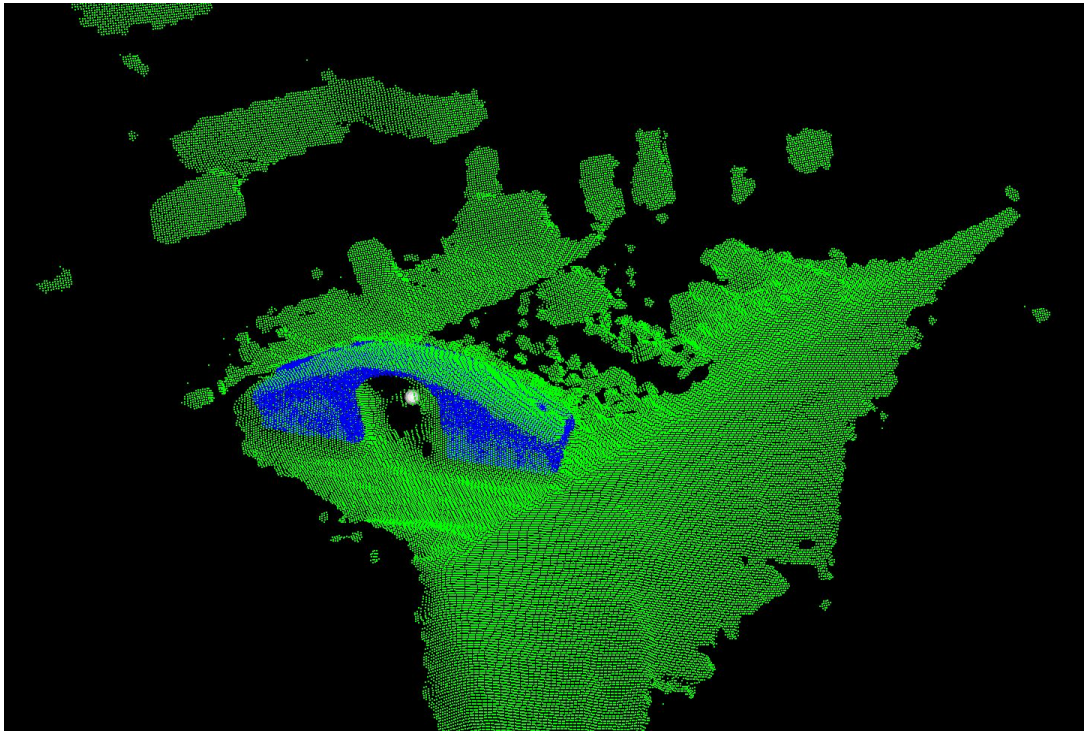


Results



LUND UNIVERSITY

Results



LUND UNIVERSITY

Connecting to a Robot

- Find transformation from camera to robot coordinate system
- Euclidean transformation
  - Need at least 3 points expressed in both coordinate systems
- Could be done automatically using pose estimation
- Could also directly estimate pose of robot if it is visible

Problem

- Can we estimate the pose of an object i.e. position and rotation using an RGB-D camera?
- How accurately can we measure the pose of the object?
- Can we translate the pose estimation into the robot's coordinate reference system?

Future Work

- Autonomous robot-camera calibration
- Using RGB data in features
- Estimating pose for multiple identical objects
- Using multiple RGB-D cameras to avoid occlusions

Thank you for listening!

Questions?

LUND UNIVERSITY