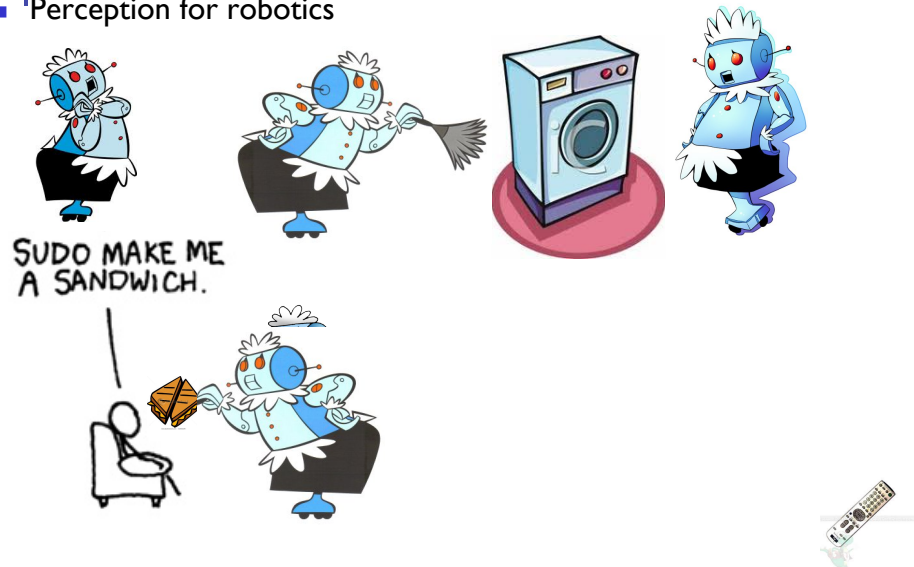


# Perception for Robotics: Instance Detection

Pieter Abbeel  
UC Berkeley EECS

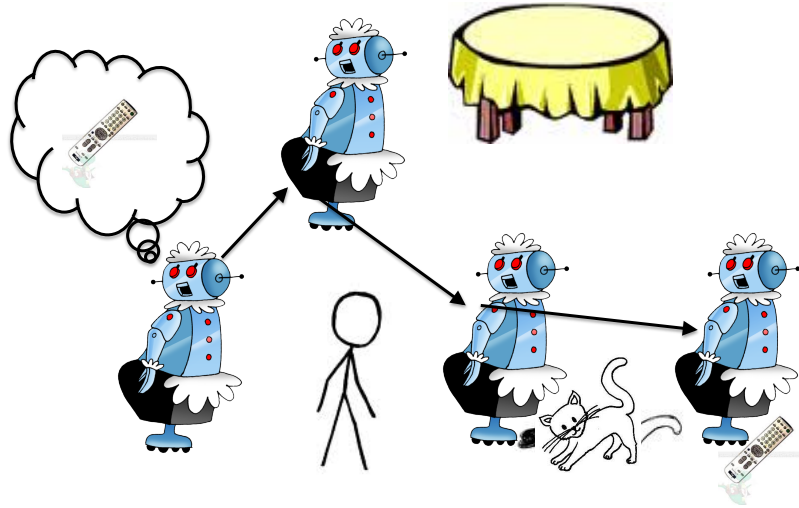
## Overview

- Perception for robotics



## Overview

- Perception for robotics



## Overview

- Perception for robotics
  - Accurately localizing (specific) objects of interest in unstructured environments quickly using multiple sensor modalities.

## Overview

- Perception for robotics
  - Accurately localizing **(specific) objects of interest** in unstructured environments quickly using multiple sensor modalities.

Non-robotic:



Robotic:



## Overview

- Perception for robotics
  - Accurately localizing **(specific) objects of interest** in unstructured environments quickly **using multiple sensor modalities.**

Non-robotic:



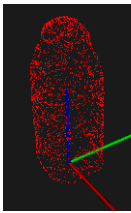
Robotic:



## Outline

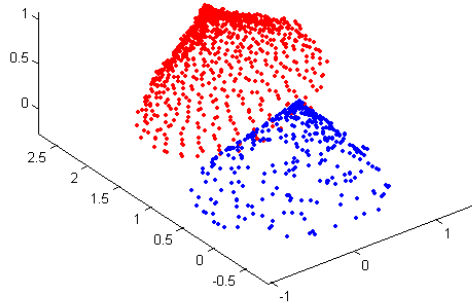
- Point clouds
  - Pose detection for known object
    - Pose scoring function: points and local features
    - Optimization and initialization: ICP and RANSAC
  - Object instance + pose detection
    - Brute force enumeration
    - Faster: Local feature based voting
- Images
  - Local image features: SIFT
  - Global features
- A full instance detection pipeline

## Problem Setting

- Given:
  - 1. From training phase: Point cloud representation of object, with attached coordinate frame
  - 2. At test time: Point cloud of scene containing same object
- Asked for: localize object in the scene (position and orientation)

## Individual Points Based Scoring Function

Different point clouds.



Red: test point cloud

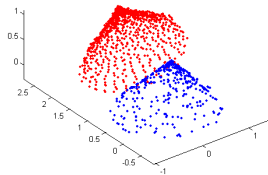
Blue: attempted match of model

Score of this match: compute distance from each blue point to closest Red point and sum the squared distances

## Optimizing the Pose with Iterated Closest Points (ICP)

- Idea: to find the optimal pose iterate over:
  - Keep pose fixed, for each (blue) point find closest match amongst (red) points
  - Keep matches fixed (aka “known correspondences”), find the rigid transformation (translation + rotation) that minimizes the sum of the squared distances between each (blue) point and its matched (red) point

Different point clouds.



## Known Correspondences

- Given: two corresponding point sets:

$$X = \{x_1, \dots, x_n\}$$

$$P = \{p_1, \dots, p_n\}$$

- Wanted: translation  $t$  and rotation  $R$  that minimizes the sum of the squared error:

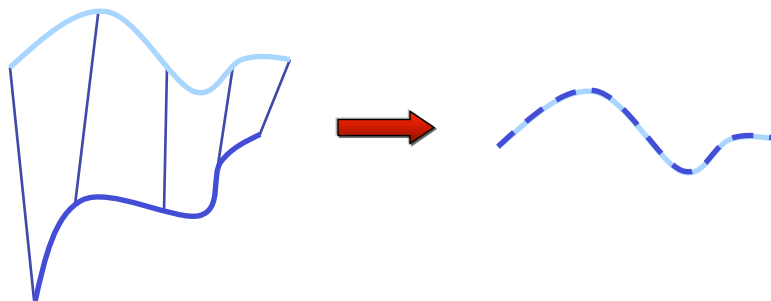
$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - Rp_i - t\|^2$$

Where  $x_i$  and  $p_i$  are corresponding points.

11

## Key Idea

- If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form.



12

## Center of Mass

$$\mu_x = \frac{1}{N_x} \sum_{i=1}^{N_x} x_i \quad \text{and} \quad \mu_p = \frac{1}{N_p} \sum_{i=1}^{N_p} p_i$$

are the centers of mass of the two point sets.

**Idea:**

- Subtract the corresponding center of mass from every point in the two point sets before calculating the transformation.
- The resulting point sets are:

$$X' = \{x_i - \mu_x\} = \{x'_i\} \quad \text{and} \\ P' = \{p_i - \mu_p\} = \{p'_i\}$$

13

## SVD

Let  $W = \sum_{i=1}^{N_p} x'_i p'^T_i$

denote the singular value decomposition (SVD) of  $W$  by:

$$W = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

where  $U, V \in \mathbb{R}^{3 \times 3}$  are unitary, and  $\sigma_1 \geq \sigma_2 \geq \sigma_3$  are the singular values of  $W$ .

14

## SVD

**Theorem** (without proof):

If  $\text{rank}(W) = 3$ , the optimal solution of  $E(R,t)$  is unique and is given by:

$$R = UV^T$$
$$t = \mu_x - R\mu_p$$

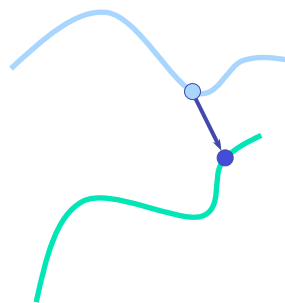
The minimal value of error function at  $(R,t)$  is:

$$E(R,t) = \sum_{i=1}^{N_p} (\|x'_i\|^2 + \|y'_i\|^2) - 2(\sigma_1 + \sigma_2 + \sigma_3)$$

15

## Closest-Point Matching

- Find closest point in other point set



The matching point is not a great match even though distance-wise close.

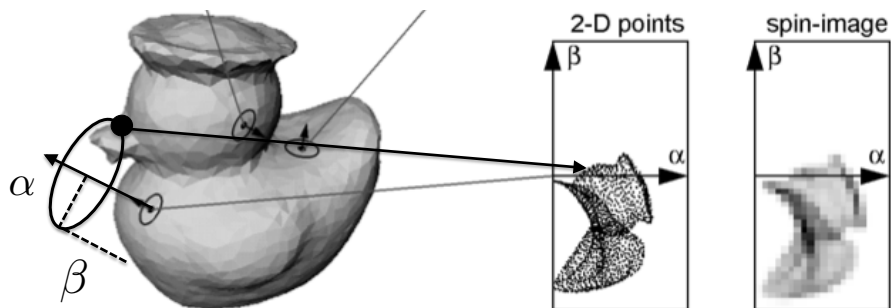
16



## Local Features

- Local features characterize geometry around a point
- Examples:
  - All pairwise distances between points within certain radius of current point
  - Spin Image
  - 3D Shape Context
  - Heat Kernel Signature
  - Point Feature Histogram (PFH), Fast PFH (FPFH)

## Example: Spin Images



## Feature Based Closest Point Matching

- Now distance between two points
- = Euclidean distance (as before)
- + distance in feature space

## Remaining Issue: ICP only finds local optimum → initialization?

- RANSAC:
  - Amongst points on the test model that have distinguished local features (i.e., very few reasonable matches on the training model)
    - Pick a few points at random, as well as randomly pick amongst their reasonable feature matches on the training model
    - Initialize pose estimate by lining up these few points as well as possible
    - Then start ICP
  - Also allows to handle outliers, see next slides

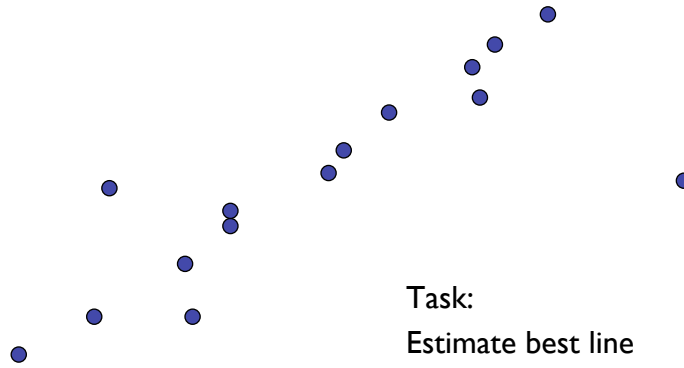
## RANSAC

- RANdom Sample Consensus
- Approach: we want to avoid the impact of outliers, so let's look for "inliers", and use those only.
- Intuition: if an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points.

## RANSAC

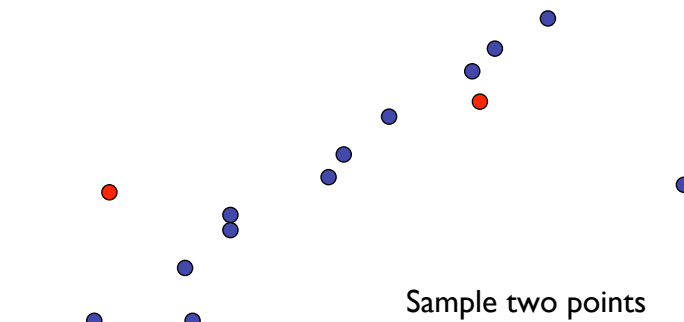
- RANSAC loop:
  1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
  2. Compute transformation from seed group
  3. Find *inliers* to this transformation
  4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers
- Keep the transformation with the largest number of inliers

## RANSAC Line Fitting Example

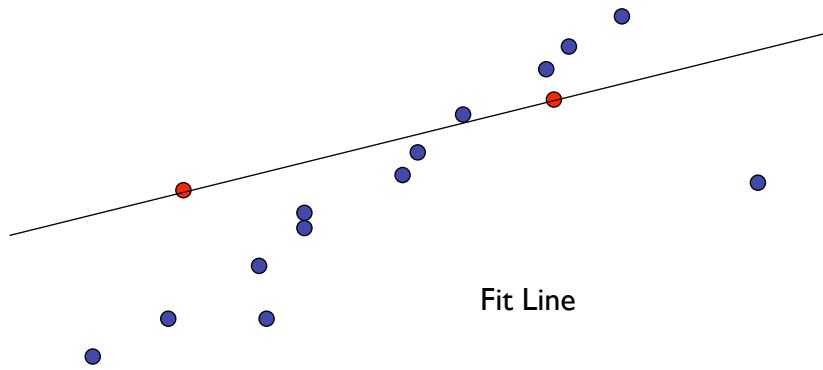


Slide credit: Jinxiang Chai, CMU

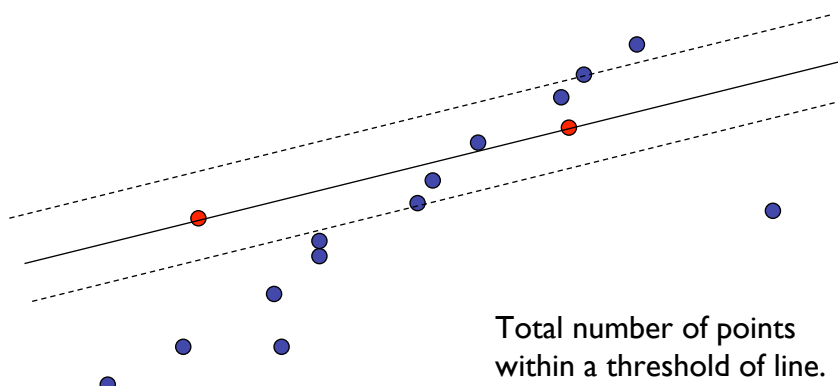
## RANSAC Line Fitting Example



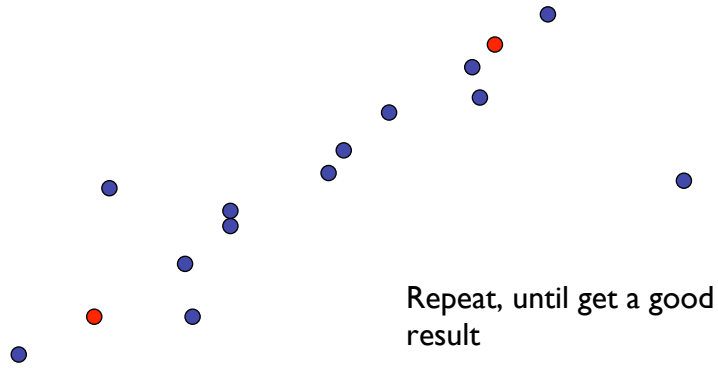
## RANSAC Line Fitting Example



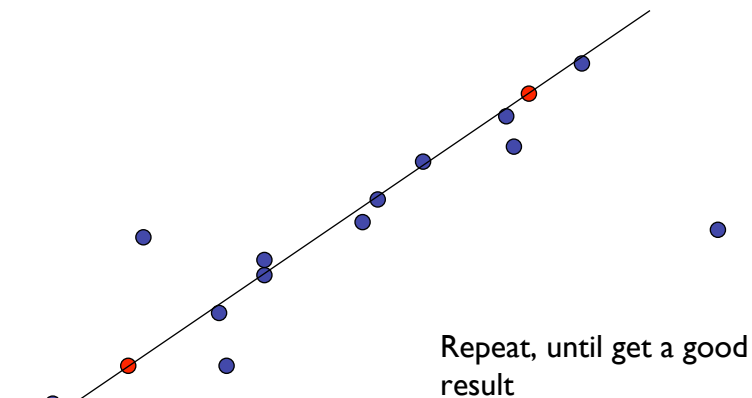
## RANSAC Line Fitting Example



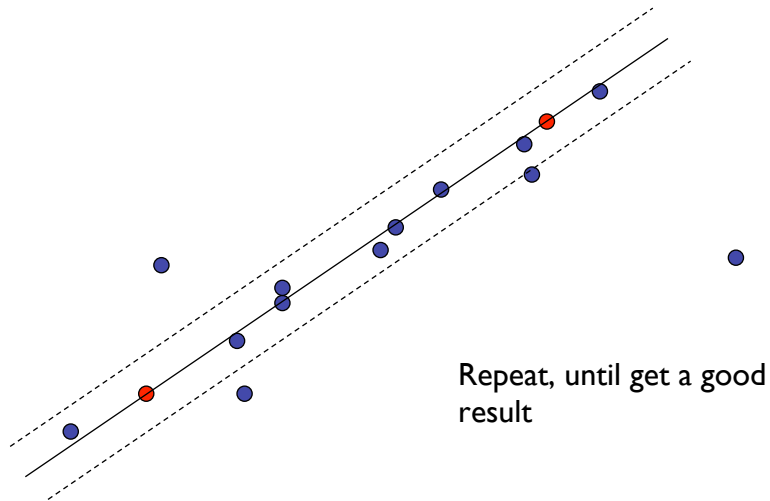
## RANSAC Line Fitting Example



## RANSAC Line Fitting Example



## RANSAC Line Fitting Example



## Outline

- Point clouds
  - Pose detection for known object
    - Pose scoring function: points and local features
    - Optimization and initialization: ICP and RANSAC
  - **Object instance + pose detection**
    - **Brute force enumeration**
    - **Faster: Local feature based voting**
- Images
  - Local image features: SIFT
  - Global features
- A full instance detection pipeline

## Object Instance + Pose Detection

- Setting: many training examples



- Naïve approach:
  - Collect point cloud representation for all
  - At test time, test for all in parallel, return instance with lowest error score

## Voting

- At training time:
  - Build nearest-neighbor data structure that stores all local features for all objects
- At test time:
  - For each point in test cloud:
    - compute local feature
    - look it up in nearest-neighbor data structure
    - Vote for instance the nearest neighbor came from
  - For instances receiving most votes, run RANSAC+ICP and return winner (= now called “geometric verification”)
- Voting variants:
  - Every object gets a vote between 0 and 1 according to nearest-feature distance
  - Vote for object and pose of the object (Hough voting)

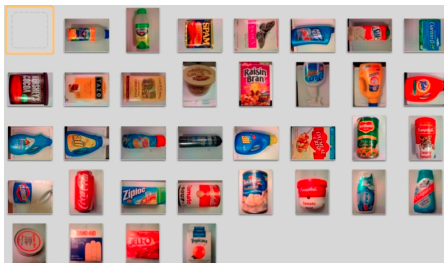


## Outline

- Point clouds
  - Pose detection for known object
    - Pose scoring function: points and local features
    - Optimization and initialization: ICP and RANSAC
  - Object instance + pose detection
    - Brute force enumeration
    - Faster: Local feature based voting
- Images
  - **Local image features: SIFT**
  - **Global features**
- A full instance detection pipeline

## Image / RGB Features

- Point cloud features only exploit shape
- Image features can exploit color, texture on object surfaces

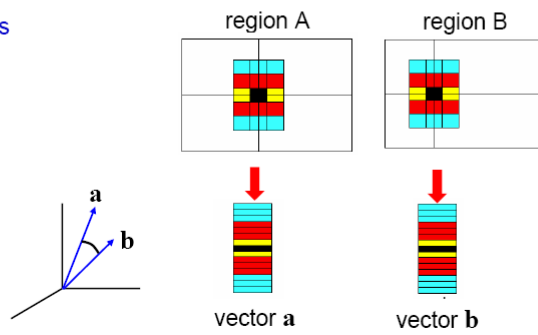


# Local descriptors

- Simplest descriptor: list of intensities within a patch.
- What is this going to be invariant to?

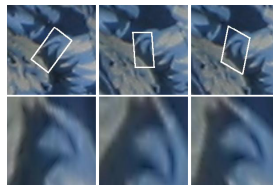
Write regions as vectors

$A \rightarrow \mathbf{a}, B \rightarrow \mathbf{b}$

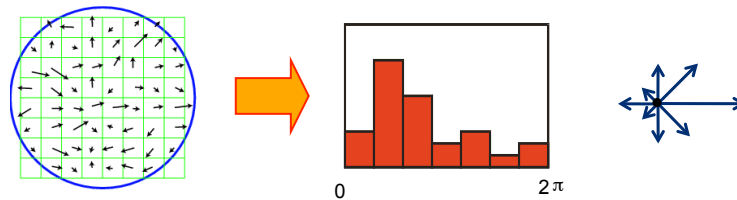


# Feature descriptors

- Disadvantage of patches as descriptors:
  - Small shifts can affect matching score a lot



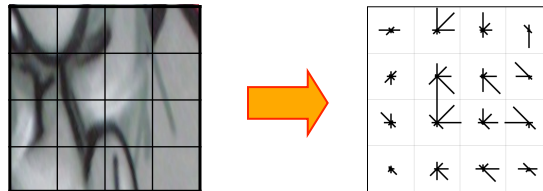
- Solution: histograms



Source: Lana Lazebnik

## Feature descriptors: SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor:  $4 \times 4 \times 8 = 128$  dimensions



David G. Lowe.

"Distinctive image features from scale-invariant keypoints."

*IJCV* 60 (2), pp. 91-110, 2004.

Source: Lana Lazebnik

## Global Features

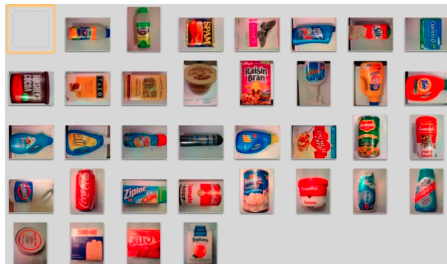
- Global feature we have used:
  - Color histogram
- Added this to the voting scheme

## Outline

- Point clouds
  - Pose detection for known object
    - Pose scoring function: points and local features
    - Optimization and initialization: ICP and RANSAC
  - Object instance + pose detection
    - Brute force enumeration
    - Faster: Local feature based voting
- Images
  - Local image features: SIFT
  - Global features
- **A full instance detection pipeline**

## Solutions in Perception Challenge (ICRA 2011)

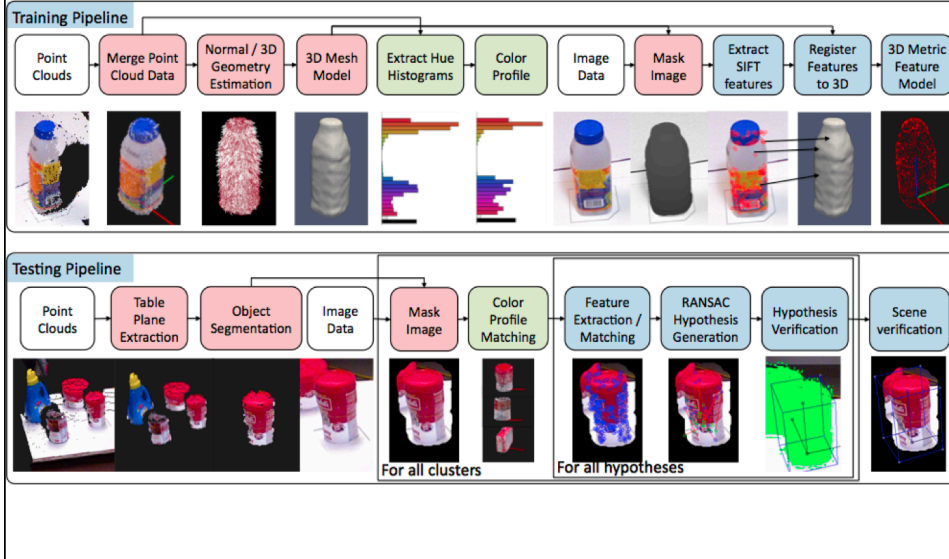
- 35 objects:



- Test examples:



# Our Pipeline



## Rank of True Matches before Geometric Verification

- This tells us how much (luckily, how little) we are losing by speeding things up through the voting scheme:

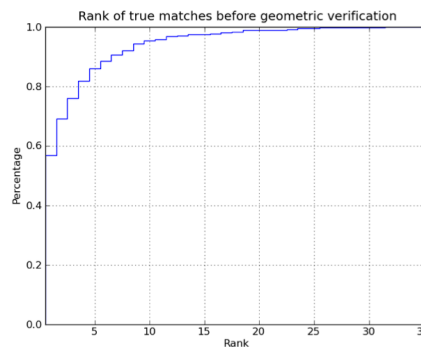


Fig. 5. Cumulative histogram of the rank of the true object after global and local feature matching but before geometric pose verification on the Willow challenge dataset. The true object lies in the top 15 over 95% of the time.

## Performance: Instance

PRECISION AND RECALL RESULTS FOR THE CURRENT PIPELINE AND  
THE ICRA 2011 CONTEST ENTRY.

	Precision	Recall
Willow (Current System)	88.75%	64.79%
Challenge (Current System)	98.73%	90.23%
NIST (Current System)	97.24%	97.70%
Challenge (ICRA 2011 Contest)	95.30%	84.10%

## Performance: Pose Accuracy (if detected correct instance)

