Robotic grasping

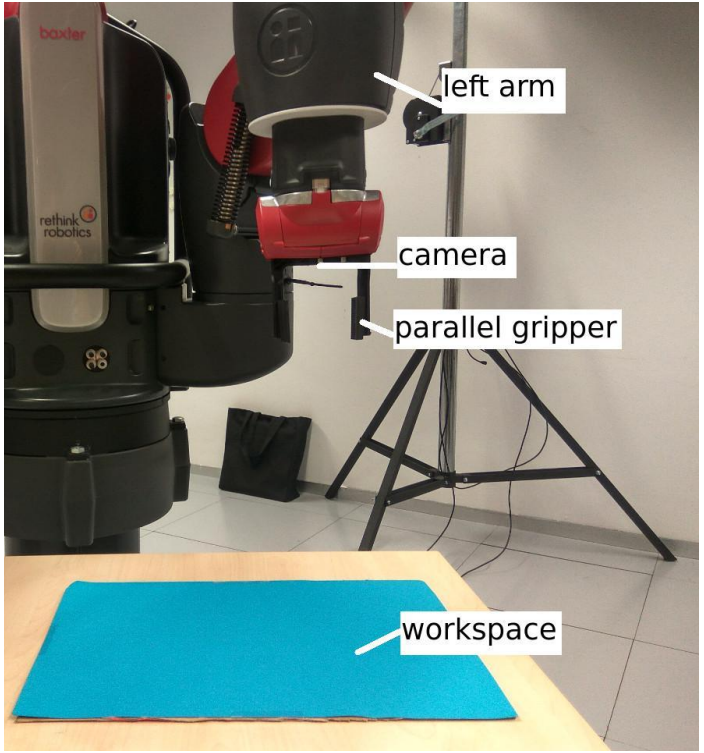
Group WangThinh
Mengnan Wang
Nguyen Truong Thinh

Overview

- Solve robotic grasping of single object in *2D* plane.
- Apply reinforcement learning and supervised learning.

System

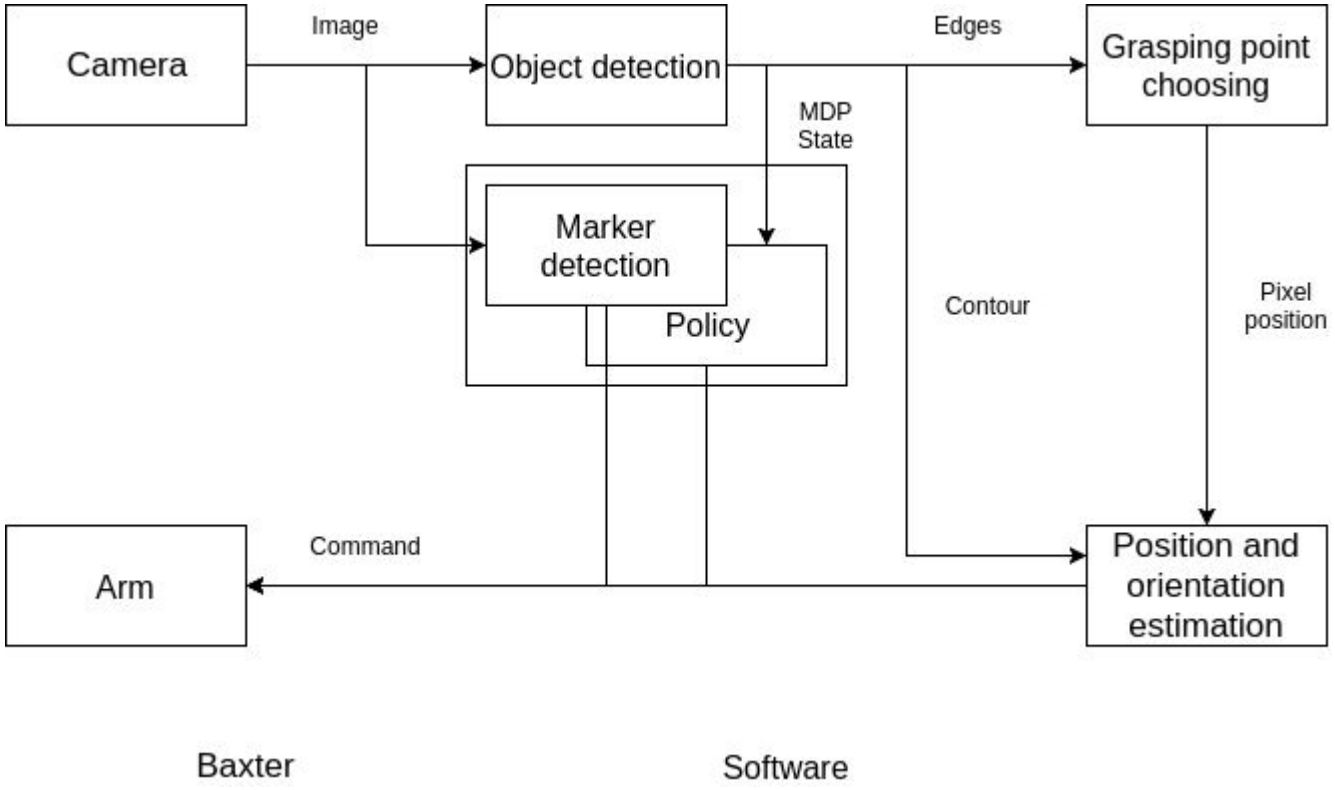
- Baxter
  - Left arm and parallel plate gripper
  - Left hand camera
- Workspace
  - Blue board
  - Light source(Optional, for better object detection)



Subtasks

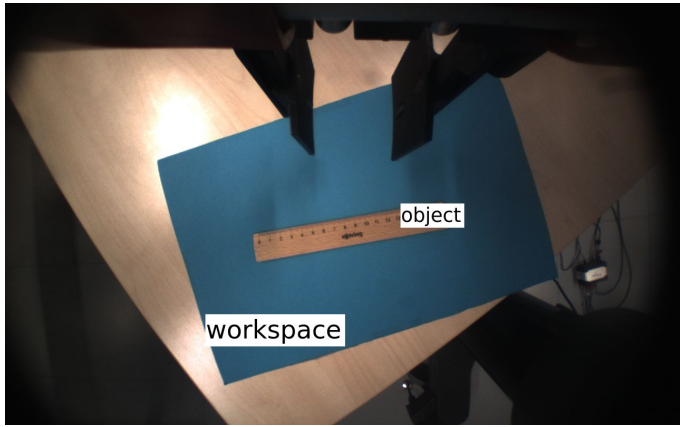
I. Approaching object in *2D* plane
II. Choosing grasping point
III. Estimating grasping position and orientation
IV. Executing grasping

System diagram



Object detection

- Workspace detection
  - Color filtering ⟶ Convex hull
- Background subtraction
  - Contour of object ⟶ Centroid

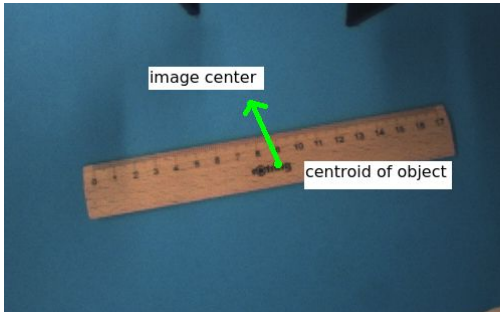


I. Approach object

- **State space**
  Vector points from centroid of object to image center

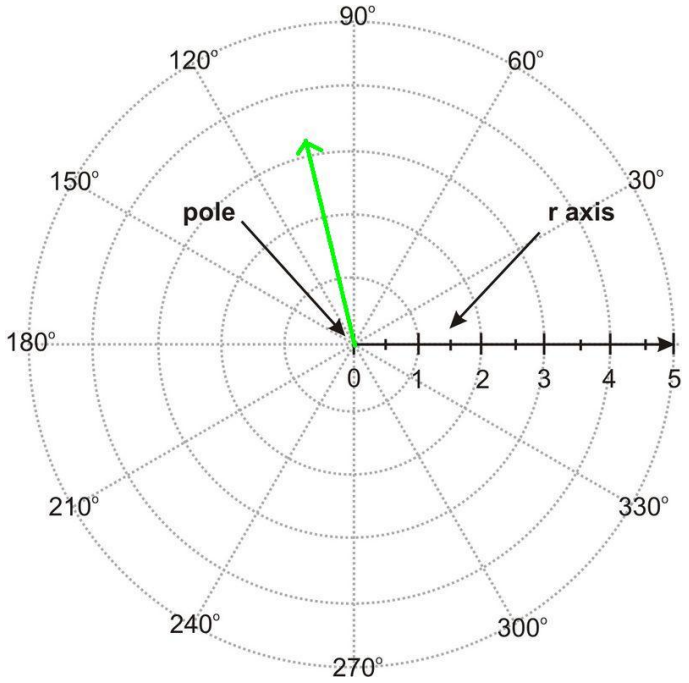
- **Action set**
  Forward/Backward/Left/Right

- **Goal state**
  Vector become zero ⟶ centroid align with image center



Reinforcement learning

- Discrete length and angle of vector in pole coordinates
  - 100 length bin
  - 36 angle bin
- Size of state space: 3600



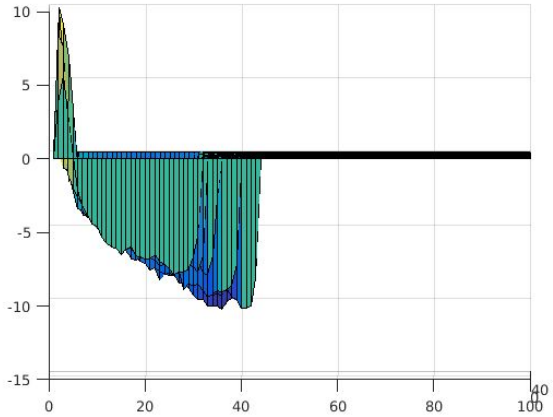
Reinforcement learning

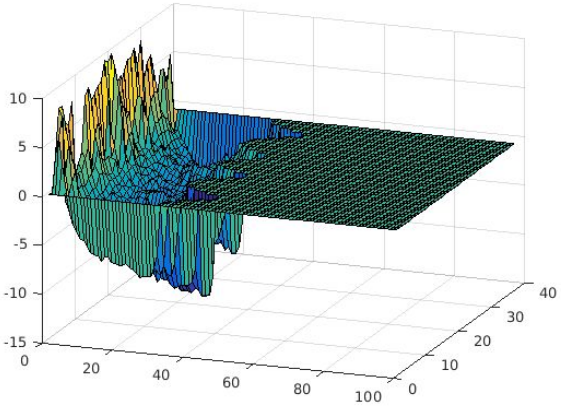
Tabular Q-Learning

Model-free approximation of State-action value Q(s,a).

Training: 20000 episodes in MLR code base

(One episode terminates when goal is reached)



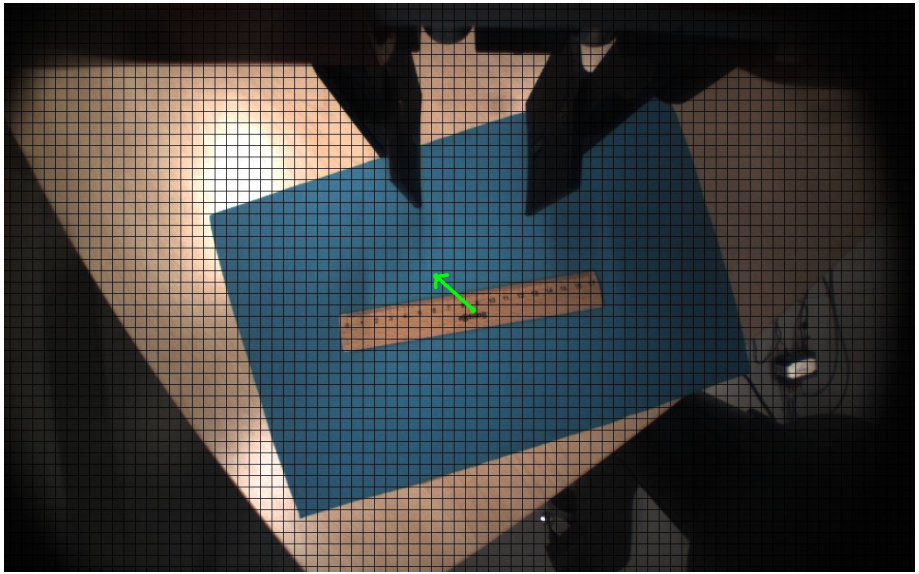


Problem: Transition dynamic on real robot is different with simulation

Reinforcement learning

- ➢ Discrete image into 201x201 grids and vector is mapped to a grid index
- ➢ Larger state space: 40401

→ Using LSPI



Reinforcement learning

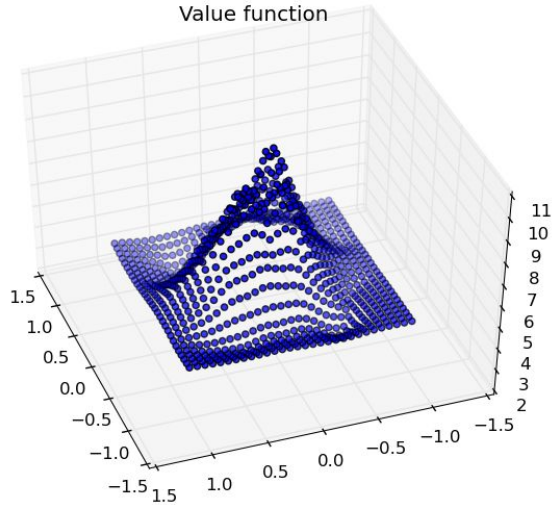
Least square policy iteration(LSPI)

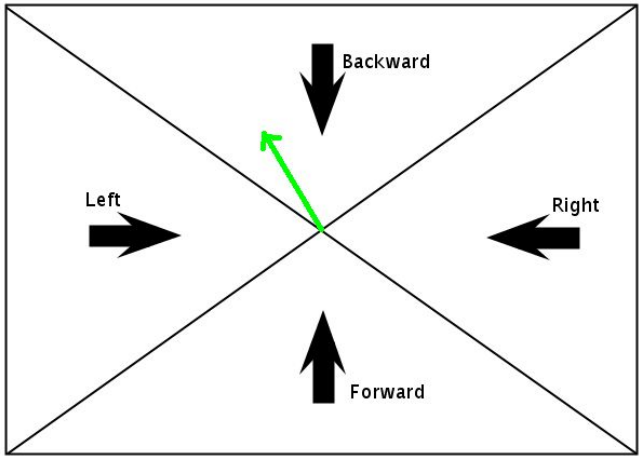
Linear approximation of Q(s,a)

$$Q(s,a) = \phi(s,a)^T \beta$$

Generalize to unvisit state

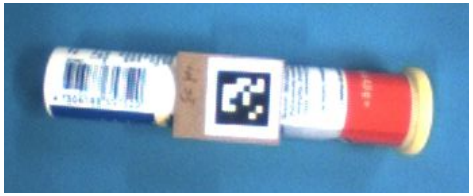
Training: 1000 episodes in Python





Aruco marker

Problem: Moving by policy is slow due to step size of action
⟶ Use Aruco marker



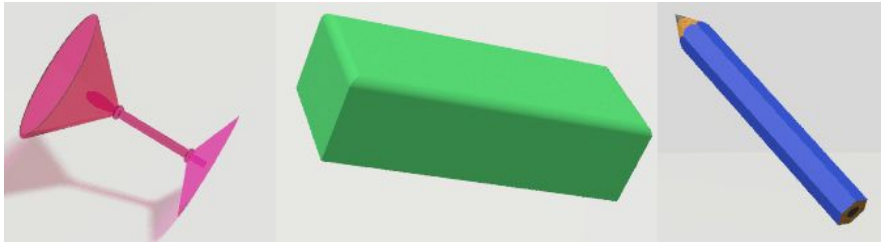
1. Estimate position of the marker

2. Policy is used to track object's centre when marker detection fail

3. Move to 20cm above marker

II. Choose grasping point

Supervised learning

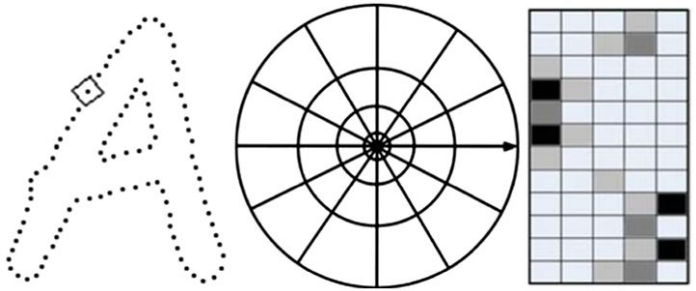


- Training data

  Stanford grasping data (Thickpencil, Martini glass, Eraser)

- Feature extraction

  Shape context + 3 spatial scales

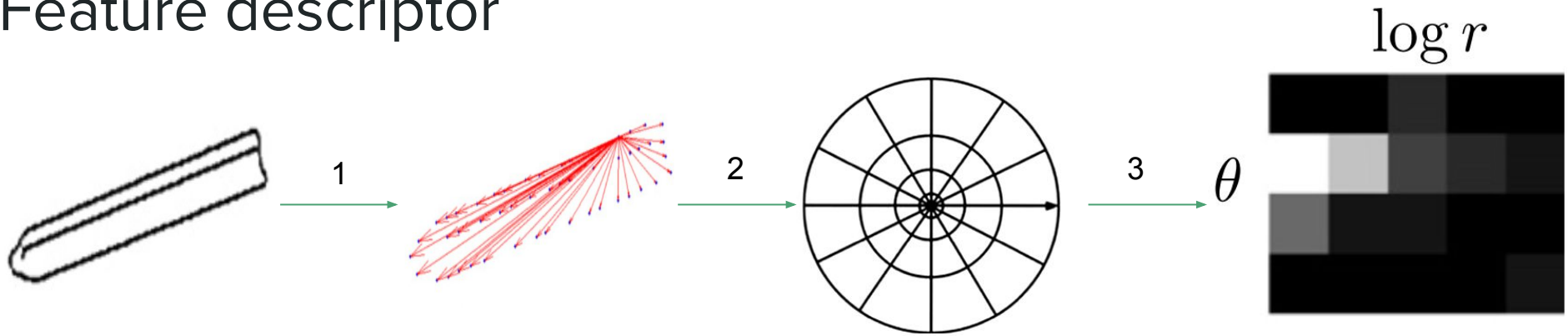


SHAPE CONTEXT

- Training model:

  Logistic regression/Non-linear SVM

Feature descriptor

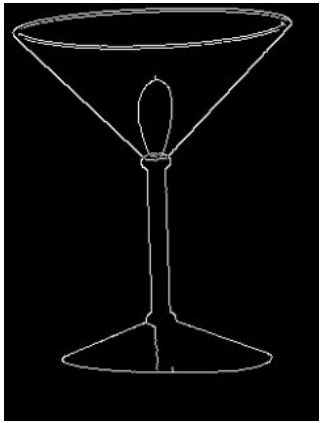


➢ **Shape context**

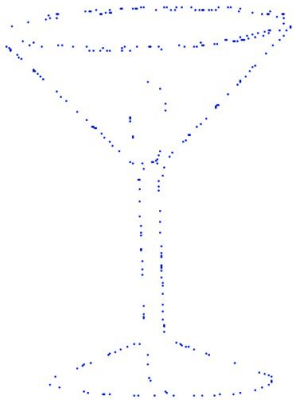
1. Sample edge into points. Calculate vector starts from each point to all others
2. Separate each vector into 5 log-radius bins and 4 angle bins
3. Store the histogram of each bin into vector of 5x4 = 20 Dimensions

Feature descriptor

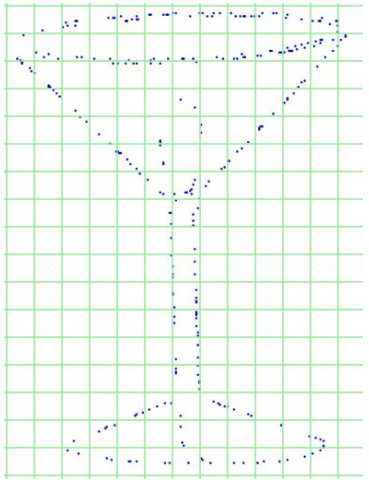
1



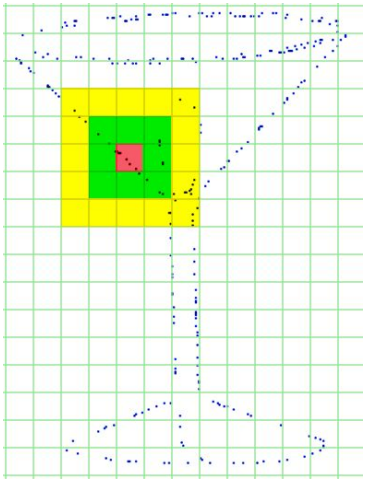
2



3



4



1. Extract object edge (Canny)
2. Sample 300 points from the edges
3. Divide image into 10x10 pixel patches
4. Concatenate shape context histogram of 3 scales into 1 vector (Our feature)

Non-linear SVM

- Kernel: Radial Basis Function
- Require suitable parameters for training model

→ Use Grid search

- Grid search time: 8 days for 5x5 parameters table
- Training time: about 2 hours for 121000 data points
- Accuracy: 57%

Logistic regression

Given data $D = \{(x_i, y_i)\}_{i=1}^{n}$, we minimize

$$L^{\text{logistic}}(\beta) = -\sum_{i=1}^{n} \log p(y_i \mid x_i) + \lambda \|\beta\|^2$$

- Training time: 10 minutes
- Accuracy: greater than 70%

→ Fast and good enough for predicting grasping point

→ Choose Logistic Regression for our system

II. Choose grasping point

- Test result
  - Synthetic objects

Martini Glass	Eraser	Thick Pencil
77.34%	58.08%	79.58%

  - Similar objects

Screwdriver	Cylinder-shape Objects
76.84%	77.42%

II. Choose grasping point

Problem

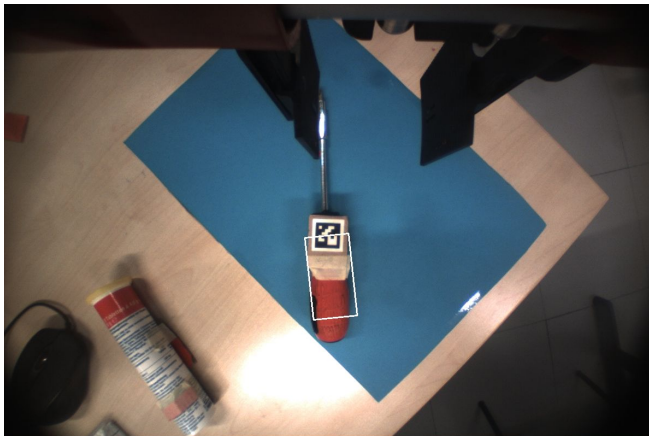
- Use *3D* grasping data to train *2D* grasping
- Unbalanced data set for classification:
  - 0.05% data are grasping point
  - 95.5% data are non-grasping point

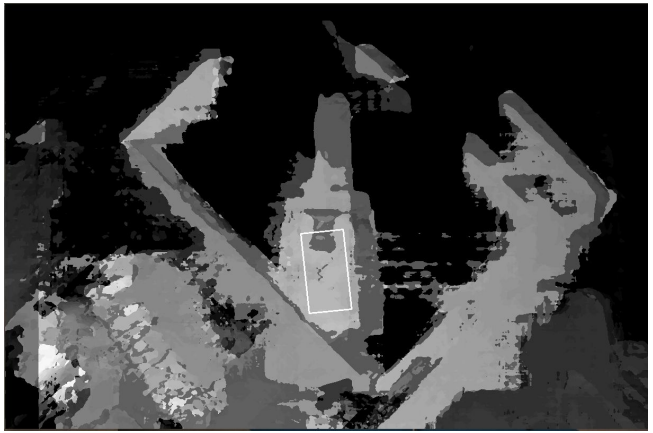
→ Balancing data set

III. Grasping point position estimation

- Stereo matching:
  - Three images, two stereo pairs

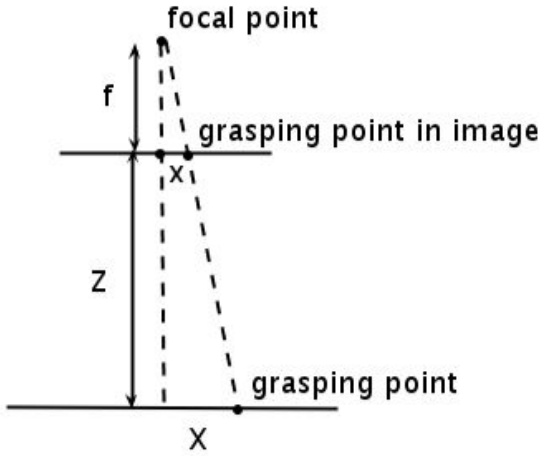
Problem: Unstable due to performance of baxter camera and loose joint

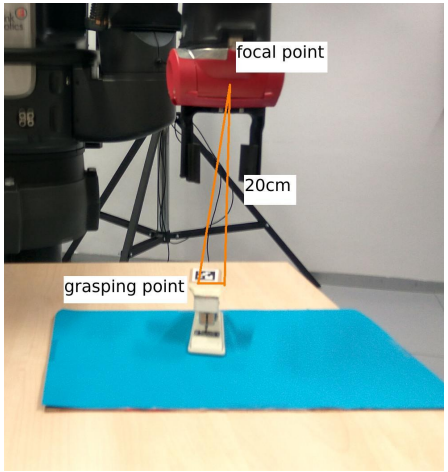




Projection relation

$$\frac{x}{X} = \frac{f}{Z}$$





Drawback: Must give depth Z first

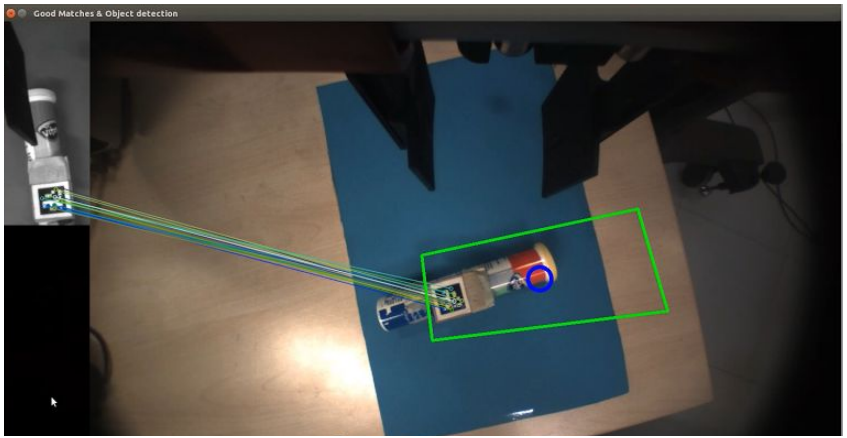
Feature tracking (SIFT)

Problem: Failure grasping may displace object

Track a small patch around grasping point

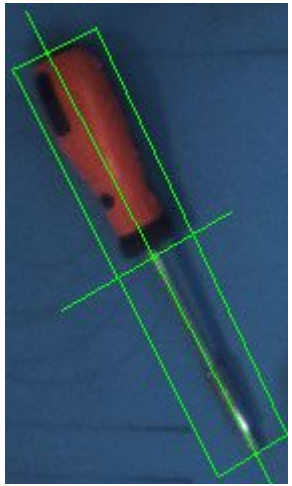
Keep updating grasping point

Re-estimate *3D* position



Orientation estimation

1. Fitting a rectangle to object
2. Choose axis which is peropendicular to the narrow side



IV. Execute grasping

1.Move to grasping point and align with grasping orientation

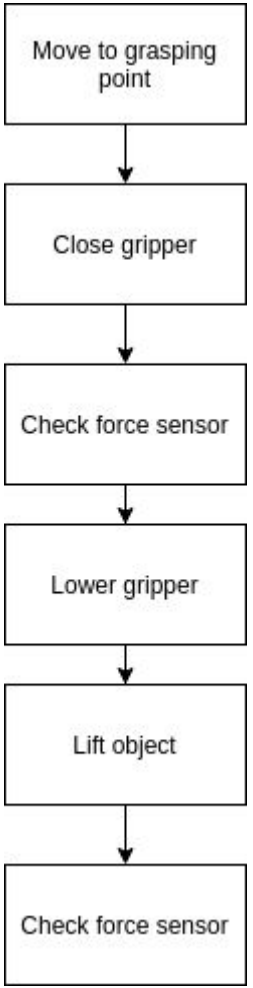
2.Close gripper

3.Check force sensor

4.Open gripper and lower gripper

5.Close gripper and lift object

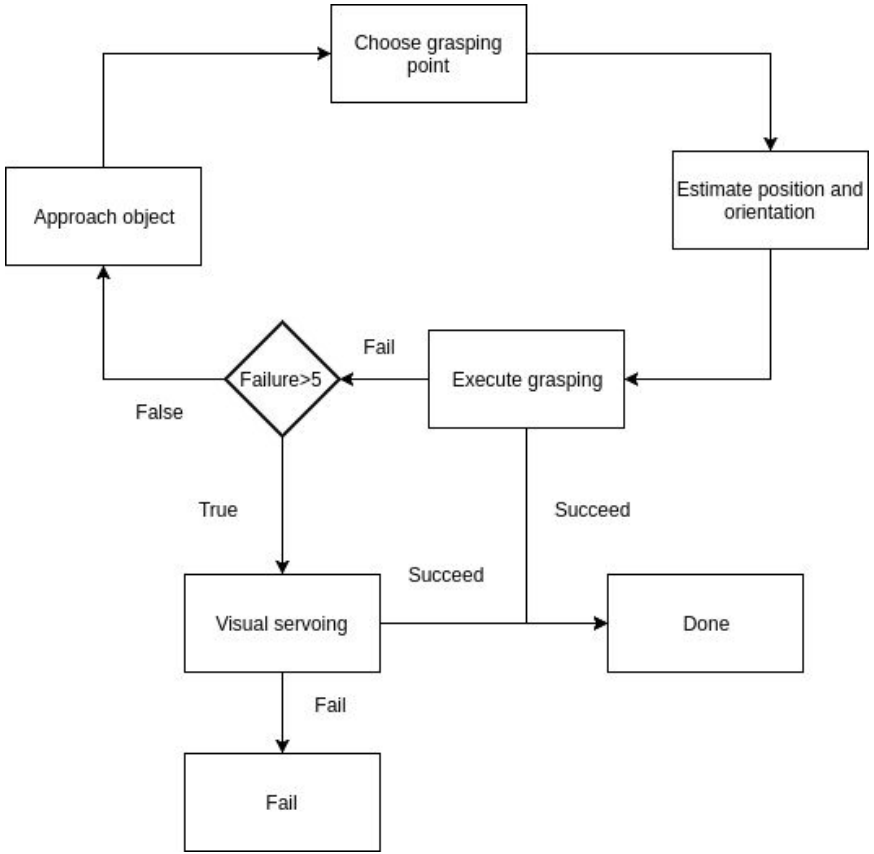
6.Check force sensor again



Grasping logic

1. Try grasping 5 times
2. Visual servoing

a) Approach in *2D* plane
b) Lower the Robot's arm
c) Close the Robot's gripper



Demo

Test result

Test No.	Screwdriver	Cylinder-shape Object	Stamper
1)	1	1	1
2)	1	1	3
3)	1	1	1
4)	1	1	1
5)	1	1	2
6)	1	1	1
7)	1	1	1
8)	2	3	1
9)	1	1	1
10)	2	Failed	2

Number of times that robot has tried to grasp successfully different objects

Multiple objects

We try to extend our projects to multiple objects

Problem:

Illumination change ⟶ Affect object detection

Conclusion

Minimum goal	Expectation	Done
Single object	Multiple object	Single and multiple (but unstale) object
Fixed shape	Unseen shape	Similar shape with training data
Fixed position and orientation	Arbitrary position and orientation	Arbitrary position and orientation

Improvement

- Hierarchical learning/SMDP
- Stereo camera for object detection/position estimation ⟶ get rid of marker
- Deep neural networks for learning grasping point

Thanks for your attention

:)