UC San Diego

Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics

Presenter: Cheng Shen

May 7th 2020

Outline

- Introduction
  - Problem Statement
  - Current Challenge
- Related work
  - Dex-Net 1.0
  - Grasp Planning: Analytical and Empirical
- Method
  - Epsilon Quality
  - Antipodal grasps
- Experiments
- Conclusion

Problem Statement

- We want to grasp some objects, but where?
- Need a policy:

$$\pi : s \rightarrow a$$

- Analytic methods:
  - model-based planning (Dex-Net 1.0)
  - Assume known contact model

$$a \leftarrow \underset{a}{\operatorname{argmax}} P_f(s, a)$$

- Empirical methods:
  - Learn the policy function directly
  - Expensive data

Recall Q-Learning

- Learn a Q-function

$$Q_\theta(a, o) = \mathbb{E}[S|a, o]$$

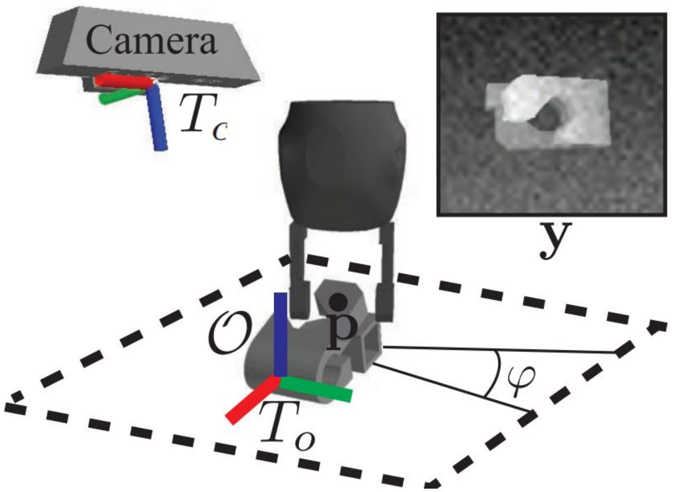
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}[\mathcal{L}(S, Q_\theta(a, o))]$$

- Grasp can be derived from the Q-function

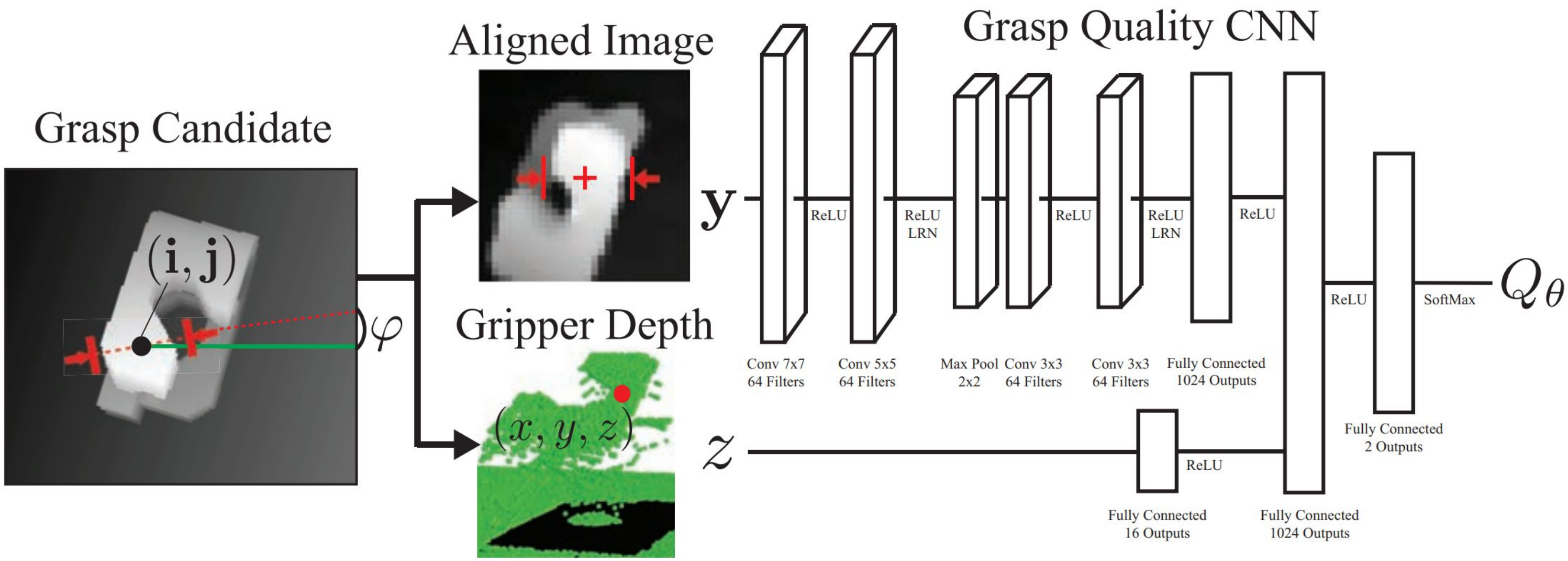
$$\pi_\theta(o) = \underset{a}{\operatorname{argmax}}\, Q_\theta(a, o)$$

Setting

- parallel-jaw gripper
- depth camera
- rigid objects on planar work surface
- depth image from camera
- grasp: antipodality constraint
- soft-finger contact model



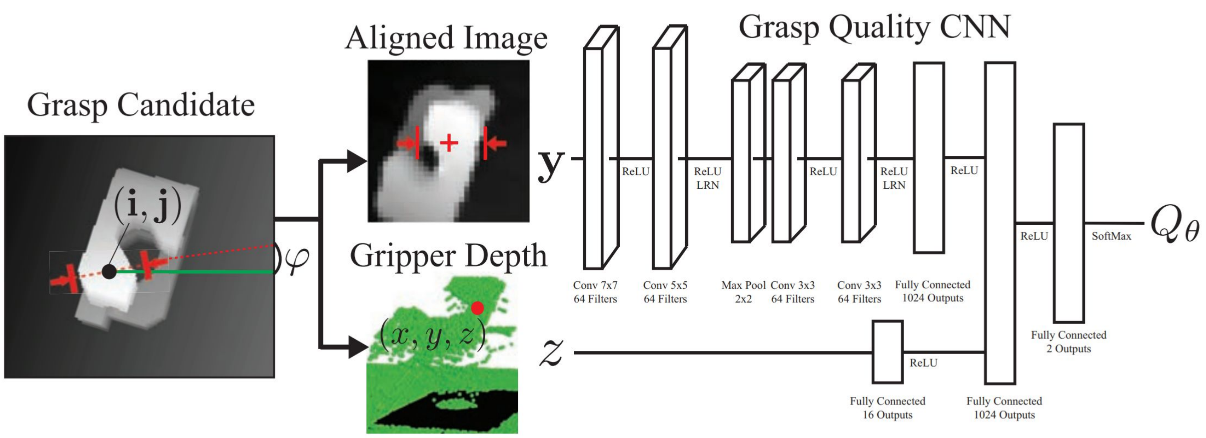
Architecture & Training



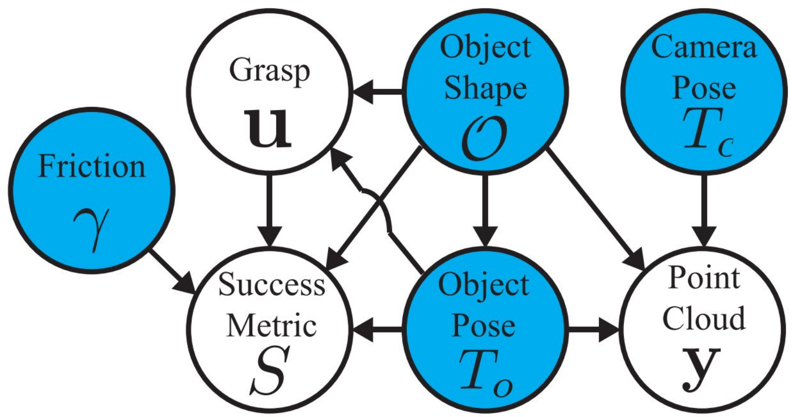
- The aligned image removes the need to learn rotation-invariant features
- Stochastic Gradient Descent

What should be our data?

- Supervised learning requires inputs and labels
- Inputs:
  - Observation of object state
  - Execution of grasp
- Labels:
  - Success or not



Variables in the Dataset



$$p(S, \mathbf{u}, \mathbf{x}, \mathbf{y}) = p(\mathbf{x}) \cdot p(\mathbf{y}|\mathbf{x}) \cdot \underline{p(\mathbf{u}|\mathbf{x}) \cdot p(S|\mathbf{u}, \mathbf{x})}$$

DexNet 1.0

- grasp sampling
- Multi-Armed Bandit Search
- Force closure measuring

Input: Object $\mathcal{O}$, Number of Candidate Grasps N_g, Number of Nearest Neighbors N_n, Dex-Net 1.0 Database $\mathcal{D}$, Features maps ψ and η, Maximum Iterations T, Prior beta shape α_0, β_0, Lower Bound Confidence p, Random Variables ν, ξ, and γ

Result: Estimate of the grasp with highest P_F, $\hat{\mathbf{g}}^*$

```
// Generate candidate grasps and priors
```

Γ = AntipodalGraspSample($\mathcal{O}, N_g$) ;

$\mathcal{A}_0 = \varnothing, \mathcal{B}_0 = \varnothing$;

for $\mathbf{g}_k \in \Gamma$ **do**

```
    // Equations VI.1 and  VI.2
```

  $\alpha_{k,0}, \beta_{k,0} =$ ComputePriors($\mathcal{O}, \mathbf{g}_k, \mathcal{D}, N_n, \psi$);

  $\mathcal{A}_0 = \mathcal{A}_0 \cup \{\alpha_{k,0}\}, \mathcal{B}_0 = \mathcal{B}_0 \cup \{\beta_{k,0}\}$;

end

```
// Run MAB to Evaluate Grasps
```

for $t = 1, .., T$ **do**

  $j =$ ThompsonSample($\mathcal{A}_{t-1}, \mathcal{B}_{t-1}$);

  $\hat{\nu}, \hat{\xi}, \hat{\gamma} =$ SampleRandomVariables(ν, ξ, γ);

  $F_j =$ EvaluateForceClosure($\mathbf{g}_j, \mathcal{O}, \hat{\nu}, \hat{\xi}, \hat{\gamma}$);

```
    // Equations VI.3 and  VI.4
```

  $\mathcal{A}_t, \mathcal{B}_t =$ UpdateBeta(j, F_j, Γ);

  $\mathbf{g}_t^* =$MaxLowerConfidence($\mathcal{A}_t, \mathcal{B}_t, p$);

end

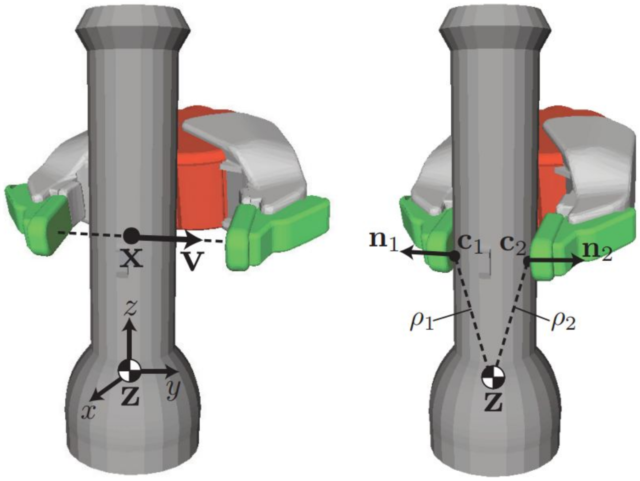
return $\mathbf{g}_T^*$;

Antipodal Grasp

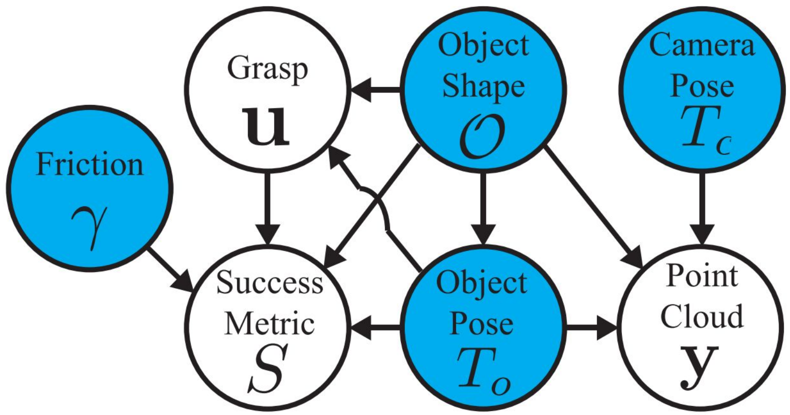
$$p(S, \mathbf{u}, \mathbf{x}, \mathbf{y}) = p(\mathbf{x}) \cdot p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{u}|\mathbf{x}) \cdot p(S|\mathbf{u}, \mathbf{x})$$

- For parallel-jaw gripper, the contact points are "antipodal"
- First sample a contact point c1 and grasp direction v, then calculate the antipodal grasp

$$\mathbf{v}^T\mathbf{n} \leq \cos(\arctan(\hat{\gamma}))$$



Variables in the Dataset



$$p(S, \mathbf{u}, \mathbf{x}, \mathbf{y}) = p(\mathbf{x}) \cdot p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{u}|\mathbf{x}) \cdot \underline{p(S|\mathbf{u}, \mathbf{x})}$$

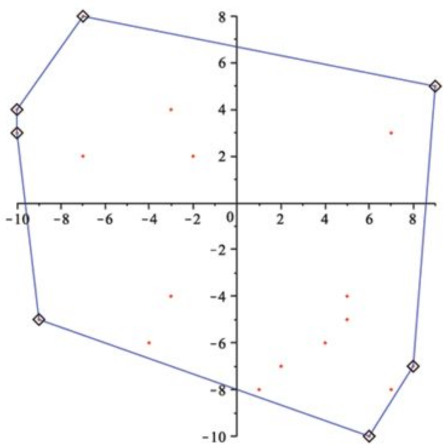
Force Closure Evaluation

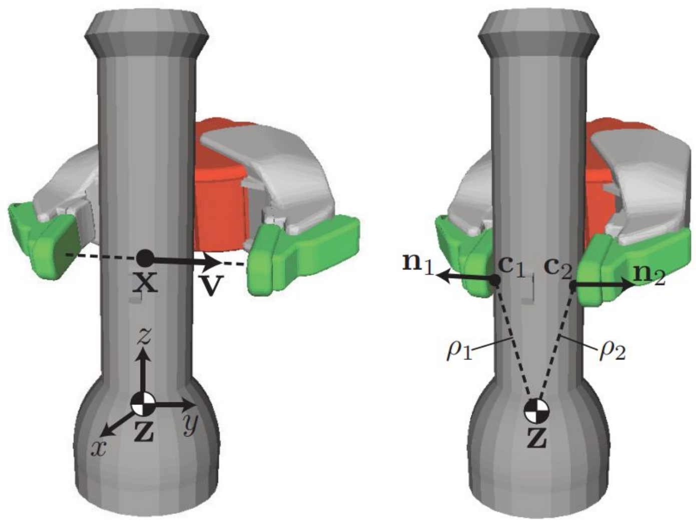
- Recall soft-finger contact model

$$\mathcal{F}_i = \{\mathbf{n}_i + \hat{\gamma}\cos(\frac{2\pi j}{l})\mathbf{t}_i) + \hat{\gamma}\sin(\frac{2\pi j}{l})\mathbf{t}_i\}$$

$$\mathcal{W} = \{\mathbf{w} = (\mathbf{f}, \tau) | \mathbf{f} \in \mathcal{F}\}$$

- 0 in/not in convex hull





DexNet 1.0

- Force closure measuring -- need some noise
- Take a closer look at DexNet 1.0

Input: Object $\mathcal{O}$, Number of Candidate Grasps N_g, Number of Nearest Neighbors N_n, Dex-Net 1.0 Database $\mathcal{D}$, Features maps ψ and η, Maximum Iterations T, Prior beta shape α_0, β_0, Lower Bound Confidence p, Random Variables ν, ξ, and γ

Result: Estimate of the grasp with highest P_F, $\hat{\mathbf{g}}^*$

`// Generate candidate grasps and priors`

Γ = AntipodalGraspSample($\mathcal{O}, N_g$) ;

$\mathcal{A}_0 = \varnothing, \mathcal{B}_0 = \varnothing$;

for $\mathbf{g}_k \in \Gamma$ **do**

    `// Equations VI.1 and  VI.2`

    $\alpha_{k,0}, \beta_{k,0} =$ ComputePriors($\mathcal{O}, \mathbf{g}_k, \mathcal{D}, N_n, \psi$);

    $\mathcal{A}_0 = \mathcal{A}_0 \cup \{\alpha_{k,0}\}, \mathcal{B}_0 = \mathcal{B}_0 \cup \{\beta_{k,0}\}$;

end

`// Run MAB to Evaluate Grasps`

for $t = 1, .., T$ **do**

    $j =$ ThompsonSample($\mathcal{A}_{t-1}, \mathcal{B}_{t-1}$);

    $\hat{\nu}, \hat{\xi}, \hat{\gamma} =$ SampleRandomVariables(ν, ξ, γ);

    $F_j =$ EvaluateForceClosure($\mathbf{g}_j, \mathcal{O}, \hat{\nu}, \hat{\xi}, \hat{\gamma}$);

    `// Equations VI.3 and  VI.4`

    $\mathcal{A}_t, \mathcal{B}_t =$ UpdateBeta(j, F_j, Γ);

    $\mathbf{g}_t^*$ =MaxLowerConfidence($\mathcal{A}_t, \mathcal{B}_t, p$);

end

return $\mathbf{g}_T^*$;

CCBP(Correlated Continuous Beta Process)

- Probability of force closure as beta distribution

$$Beta(\alpha, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

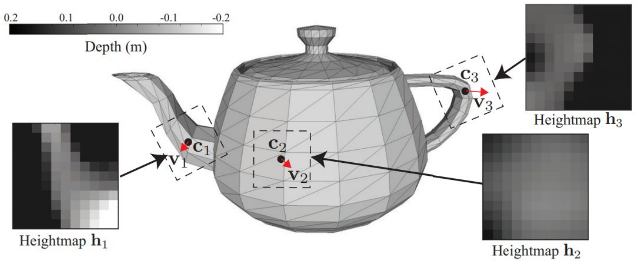
- Distribution as a function of probability theta

- Update rule:

$$\alpha_t = \alpha_{t-1} + k(\mathcal{Y}, \mathcal{Y}_{i,l})F_l$$
$$\beta_t = \beta_{t-1} + k(\mathcal{Y}, \mathcal{Y}_{i,l})(1 - F_l)$$

How to capture similarity?

- grasp parameters similarity
- local surface geometry
  - heightmaps
- object similarity
  - learned by CNN with object photo



$$k(\mathcal{Y}_p, \mathcal{Y}_q) = \exp(-\frac{1}{2}\sum_{m=1}^{3} ||\varphi_m(\mathcal{Y}_p) - \varphi_m(\mathcal{Y}_q)||_{C_m}^2)$$

$$\varphi_1(\mathcal{Y}) = (\mathbf{x}, \mathbf{v}, ||\rho_1||_2, ||\rho_2||_2)$$

$$\varphi_2(\mathcal{Y}) = \eta(\mathbf{g}, \mathcal{O})$$

$$\varphi_3(\mathcal{Y}) = \phi(\mathcal{O})$$

Robust Epsilon Quality

- Evaluate how likely a grasp can be successful
- Add uncertainty to gripper pose, object pose and friction
- Key application of DexNet-1.0 in DexNet-2.0

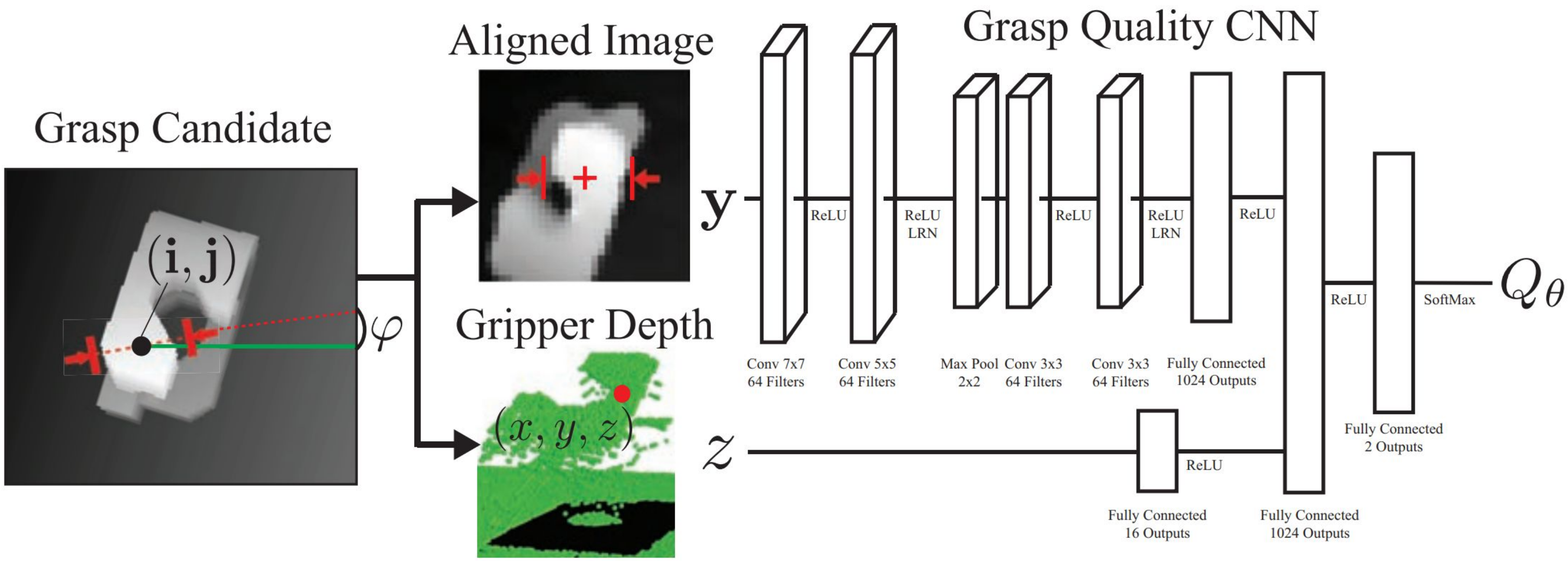
$\hat{\nu}, \hat{\xi}, \hat{\gamma} = \text{SampleRandomVariables}(\nu, \xi, \gamma);$

$F_j = \text{EvaluateForceClosure}(\mathbf{g}_j, \mathcal{O}, \hat{\nu}, \hat{\xi}, \hat{\gamma});$

`// Equations VI.3 and VI.4`

$\mathcal{A}_t, \mathcal{B}_t = \text{UpdateBeta}(j, F_j, \Gamma);$

All data prepared



$$p(S, \mathbf{u}, \mathbf{x}, \mathbf{y}) = p(\mathbf{x}) \cdot p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{u}|\mathbf{x}) \cdot p(S|\mathbf{u}, \mathbf{x})$$

Experiments

	Comparions of Methods					
	Random	**IGQ**	**ML-RF**	**ML-SVM**	**REG**	**GQ-L-Adv**
Success Rate (%)	58±11	70±10	75±9	80±9	95±5	**93±6**
Precision (%)	N/A	N/A	100	100	N/A	**94**
Robust Grasp Rate (%)	N/A	N/A	5	0	N/A	43
Planning Time (sec)	N/A	1.9	0.8	0.9	2.6	**0.8**

Conclusion

- Analogy to AlphaGo and AlphaGo Zero:
  - search V.S. learned function
  - harder sampling; simpler mapping
- GQ-CNN is faster than other methods
- The efficiency of learned function supported by large dataset and backed-up by search-based methods

UC San Diego

S4G: Amodal Single-view Single-Shot SE(3) Grasp Detection in Cluttered Scenes

Presenter: Yiran Xu

May 7th 2020