CMSC426
Computer Vision

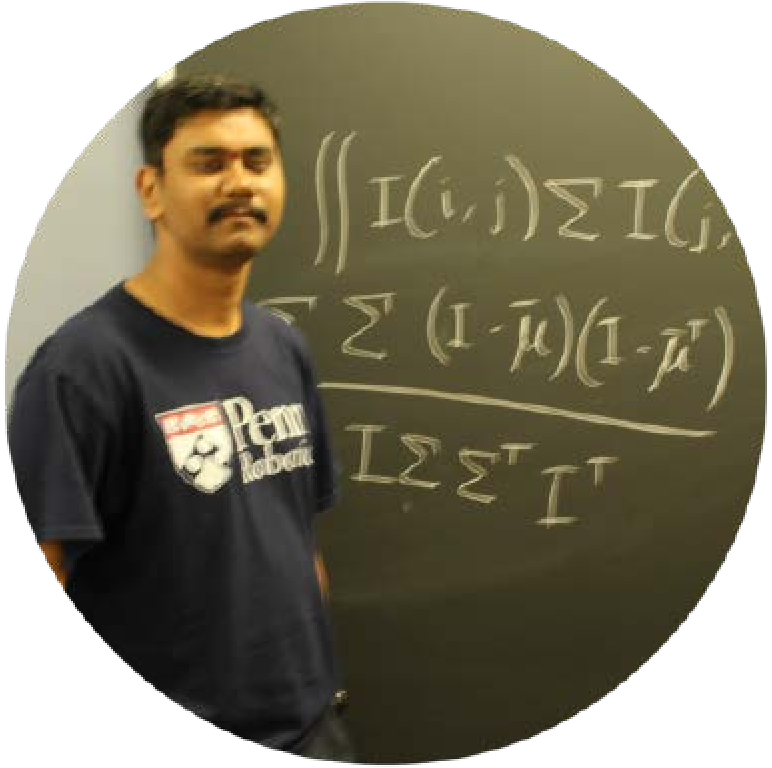
SEXY SEMANTIC MAPPING

Nitin J. Sanket

Who Am I?

Nitin J. Sanket

- BE in Electronics and Communication from India
- MS in Robotics from UPenn
- PhD Candidate in Computer Science at UMD



Perception and Robotics Group

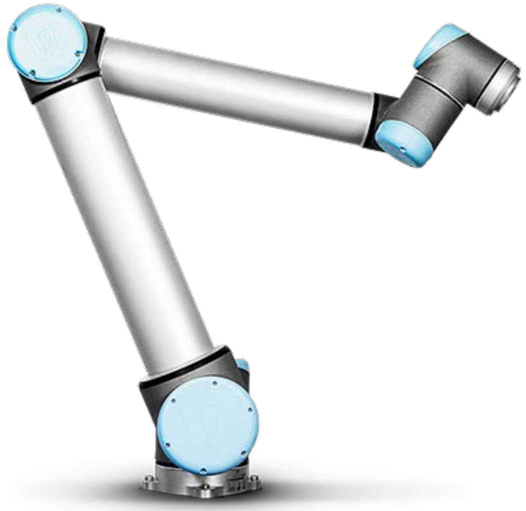
PRG
prg.cs.umd.edu

11 PhDs
5 Masters
3 Undergrads

What do we work on?








Sexy Semantic Mapping

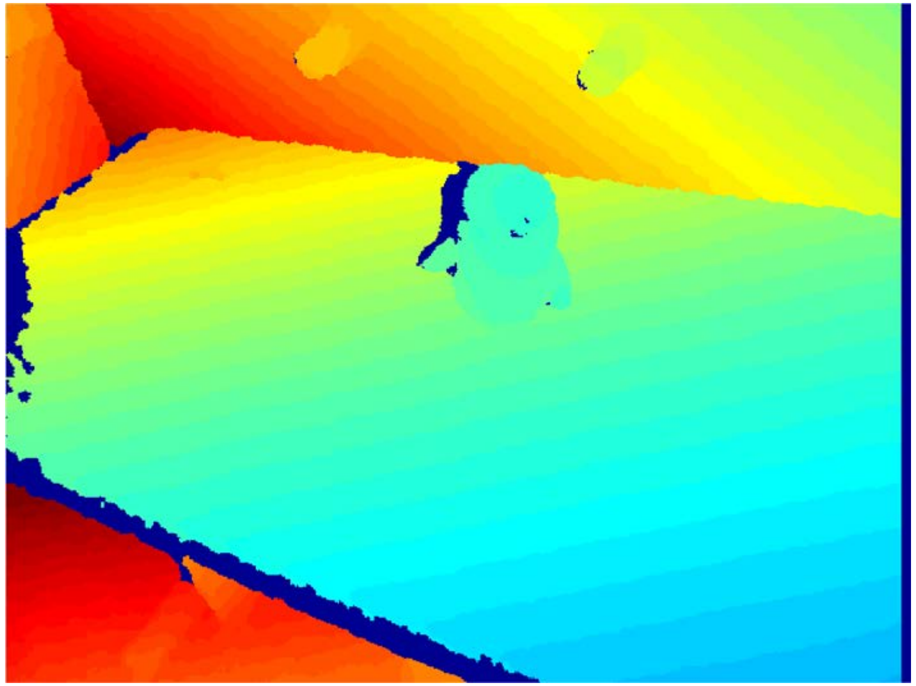
Sexy Semantic Mapping

Ecins, Aleksandrs, Cornelia Fermüller, and Yiannis Aloimonos. "Cluttered scene segmentation using the symmetry constraint." *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016.

Sexy Semantic Mapping

Data

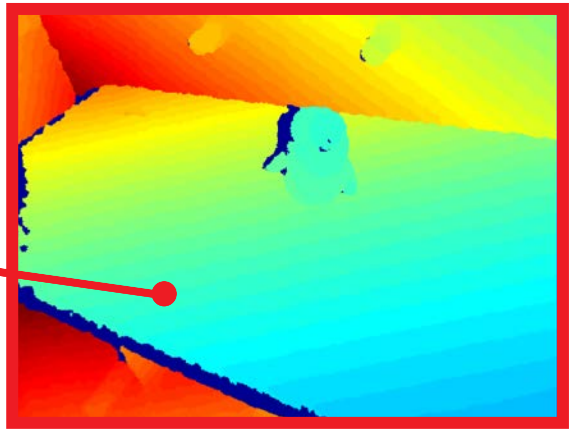
Data



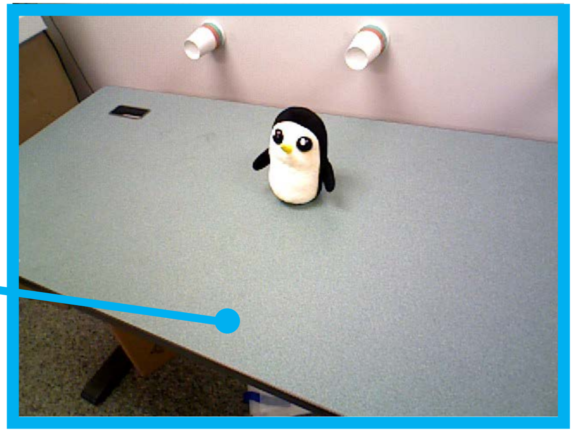
Data

Point Cloud: (x, y, z, r, g, b)

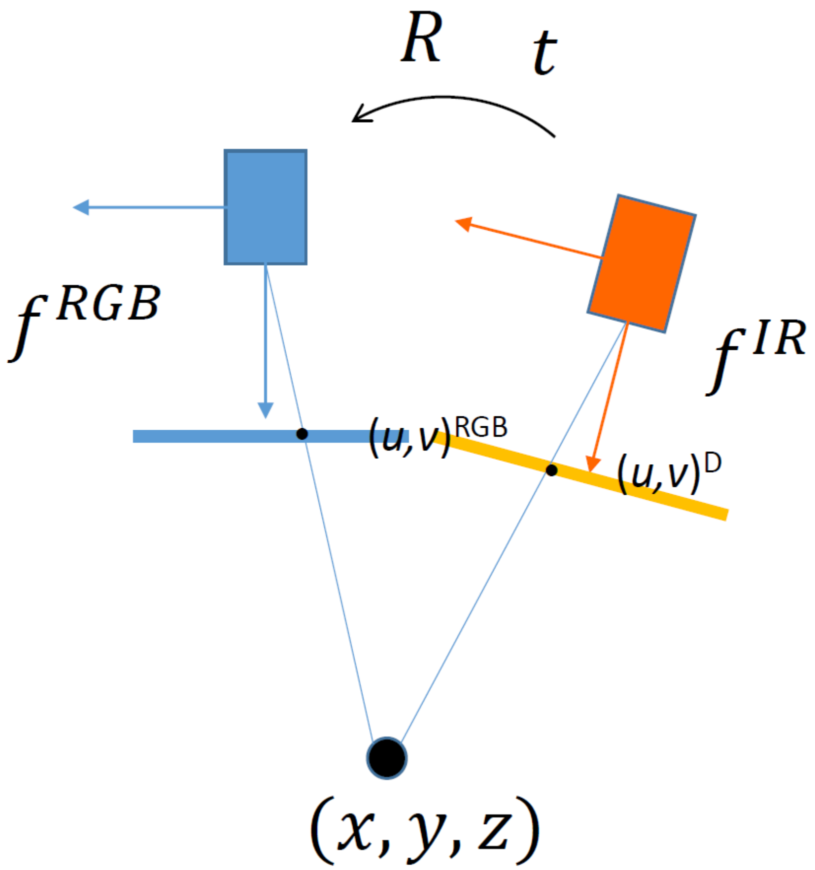
(r, g, b) at (u, v)



(x, y, z) at (u', v')



Point Cloud from RGB-D



Given all camera parameters (R, t, f), find the corresponding points of a RGB and a depth image.

Point Cloud: (x, y, z, r, g, b)

Taken from Bhoram Lee's slides at University of Pennsylvania

Point Cloud from RGB-D

Recall $\frac{u}{f_u} = \frac{x}{z}$

1. Compute 3D co-ordinate X^{IR} in the IR camera frame

$$x^{IR} = uz/f^{IR} \qquad y^{IR} = vz/f^{IR} \qquad X^{IR} = [x^{IR} \quad y^{IR} \quad z^{IR}]$$

2. Transform into the RGB frame

$$X^{RGB} = RX^{IR} + t$$

3. Re-project them into the image plane

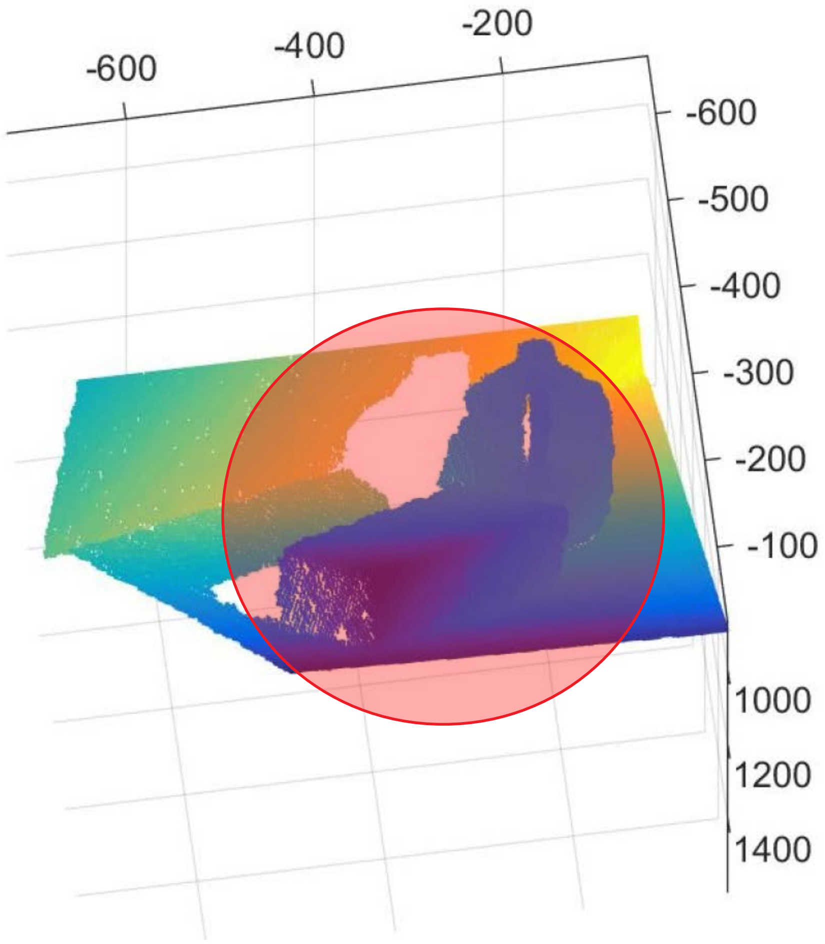
$$u^{RGB} = f^{RGB}\frac{x^{RGB}}{z^{RGB}} \quad v^{RGB} = f^{RGB}\frac{y^{RGB}}{z^{RGB}}$$

4. Read (r, g, b) at $(u, v)^{RGB}$

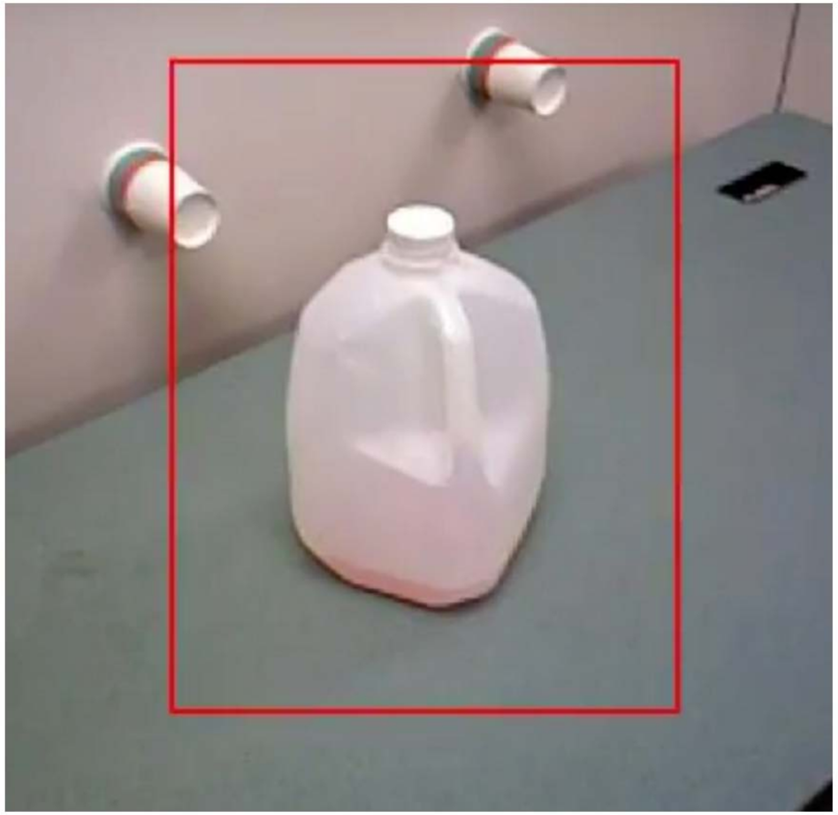
(r, g, b) is the color of X^{IR} point.

This is implemented in **`depthToCloud_full_RGB.p`** given to you.

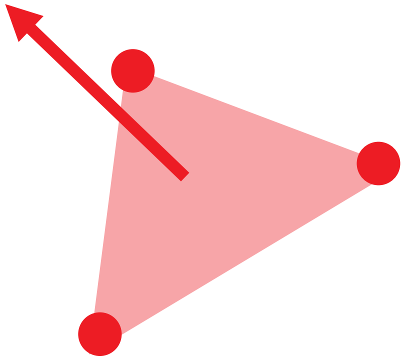
Obtain ROI by Filtering

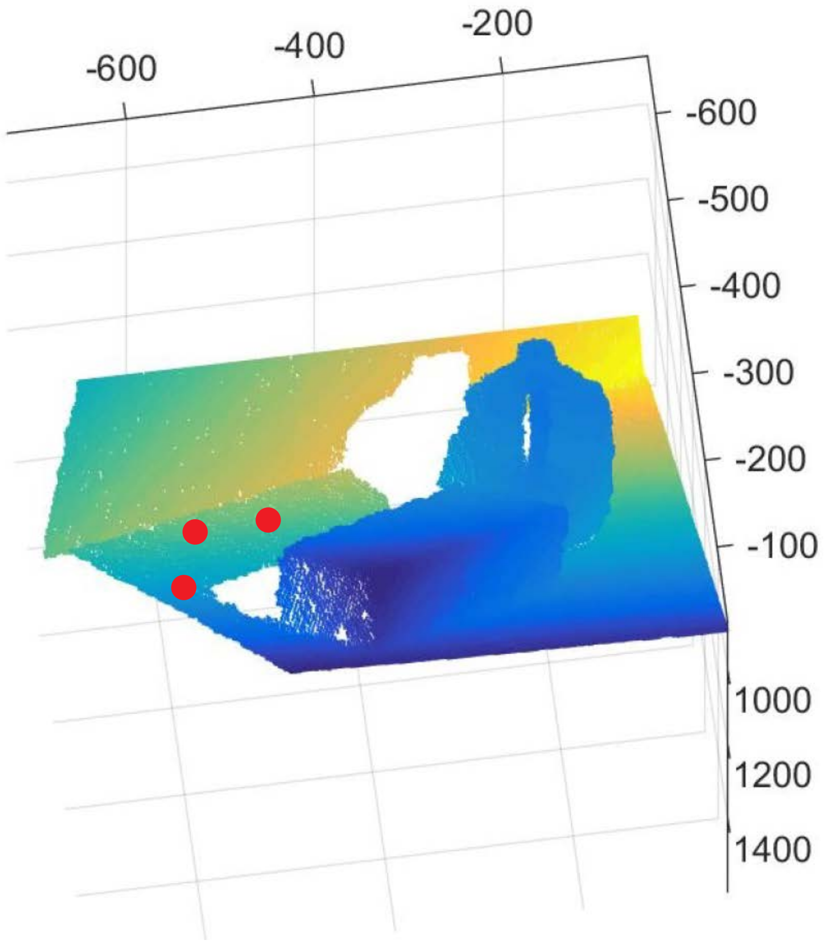


Maintain ROI by Mean-Shift Tracking



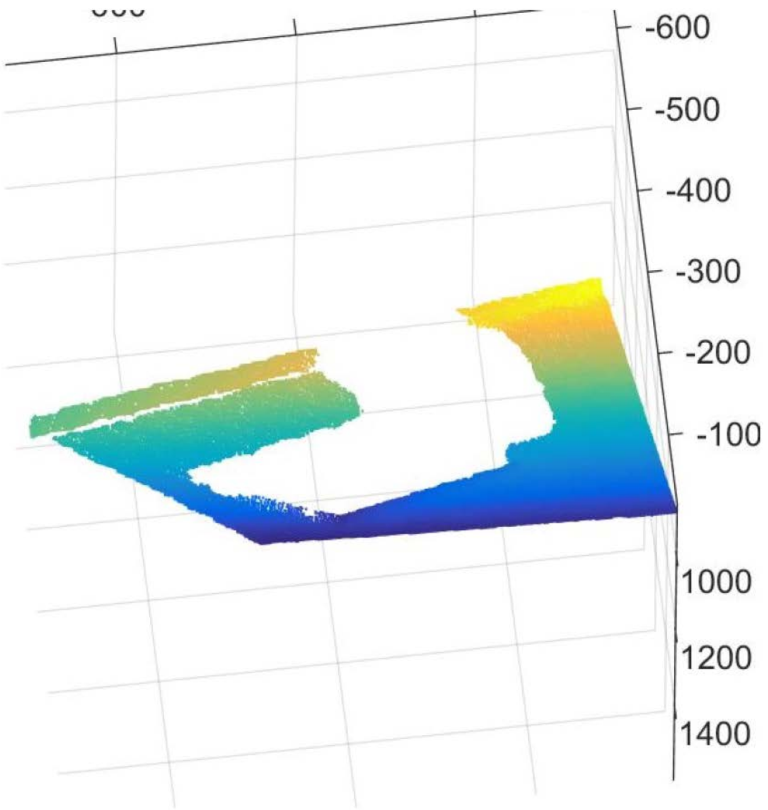
Remove the Table and wall using RANSAC

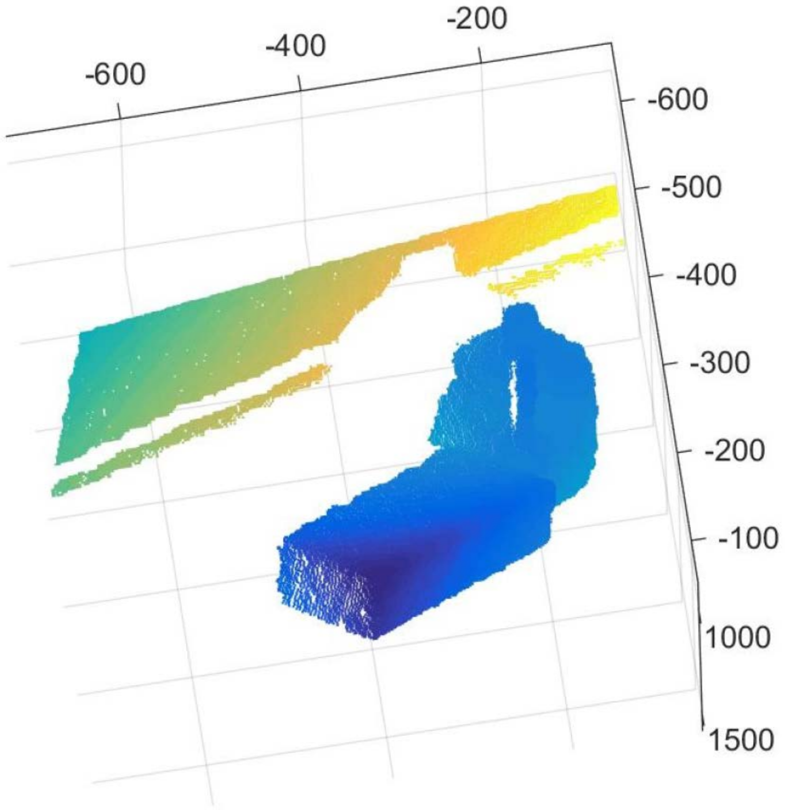




$$ax + by + cz + e = 0$$
$$err = ax + by + cz + e$$
$$err \leq \tau$$

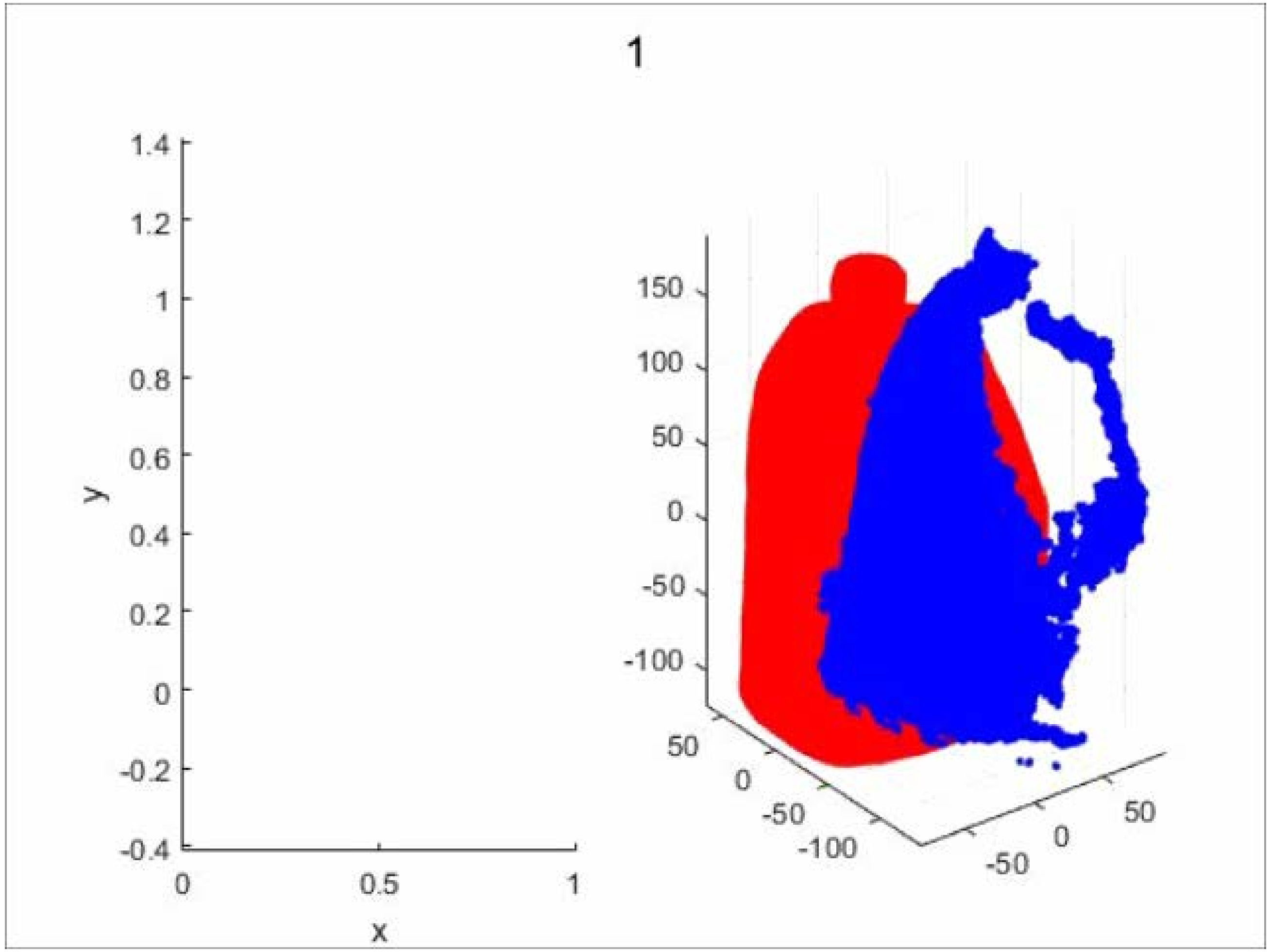
Plane Removal Output





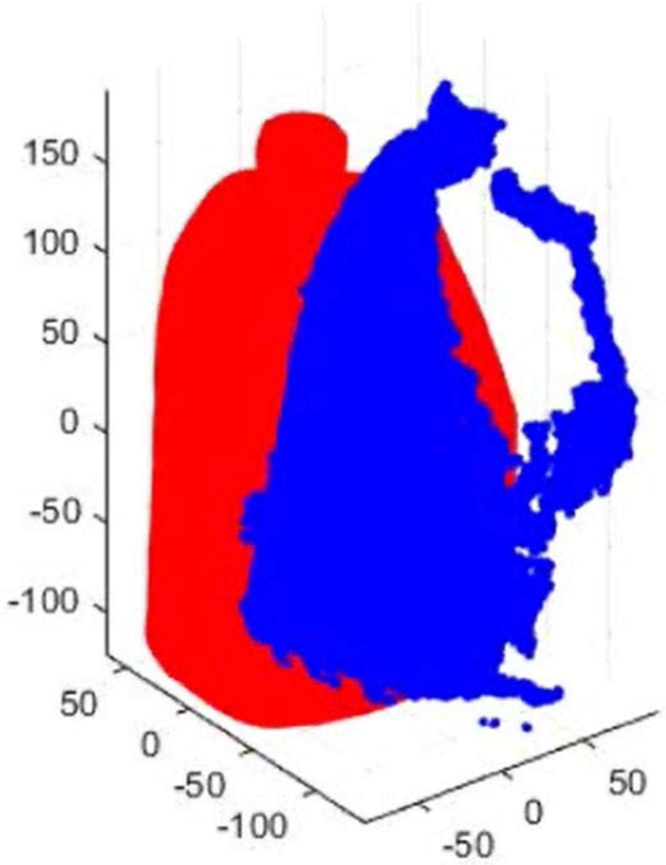
ROI Output

Iterative Closest Point



PRG
COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Point to Point Iterative Closest Point
P2P-ICP



$$\tilde{q}_i = q_i - \frac{1}{N}\sum_{\forall i} q_i$$

$$\tilde{p}_i = p_i - \frac{1}{N}\sum_{\forall i} p_i$$

$$\min_{R,T} \sum_{\forall i} \|\tilde{q}_i - R\tilde{p}_i\|^2$$

$$\hat{T} = q - \hat{R}p$$

Use **`KDTreeSearcher`** or **`knnsearch`** for point to point correspondence search.

Arun, K. Somani, Thomas S. Huang, and Steven D. Blostein. "Least-squares fitting of two 3-D point sets." *IEEE Transactions on pattern analysis and machine intelligence* 5 (1987): 698-700.

Point to Point Iterative Closest Point P2P-ICP

$$H = \sum_{\forall i} \tilde{p}_i \tilde{q}_i^T$$

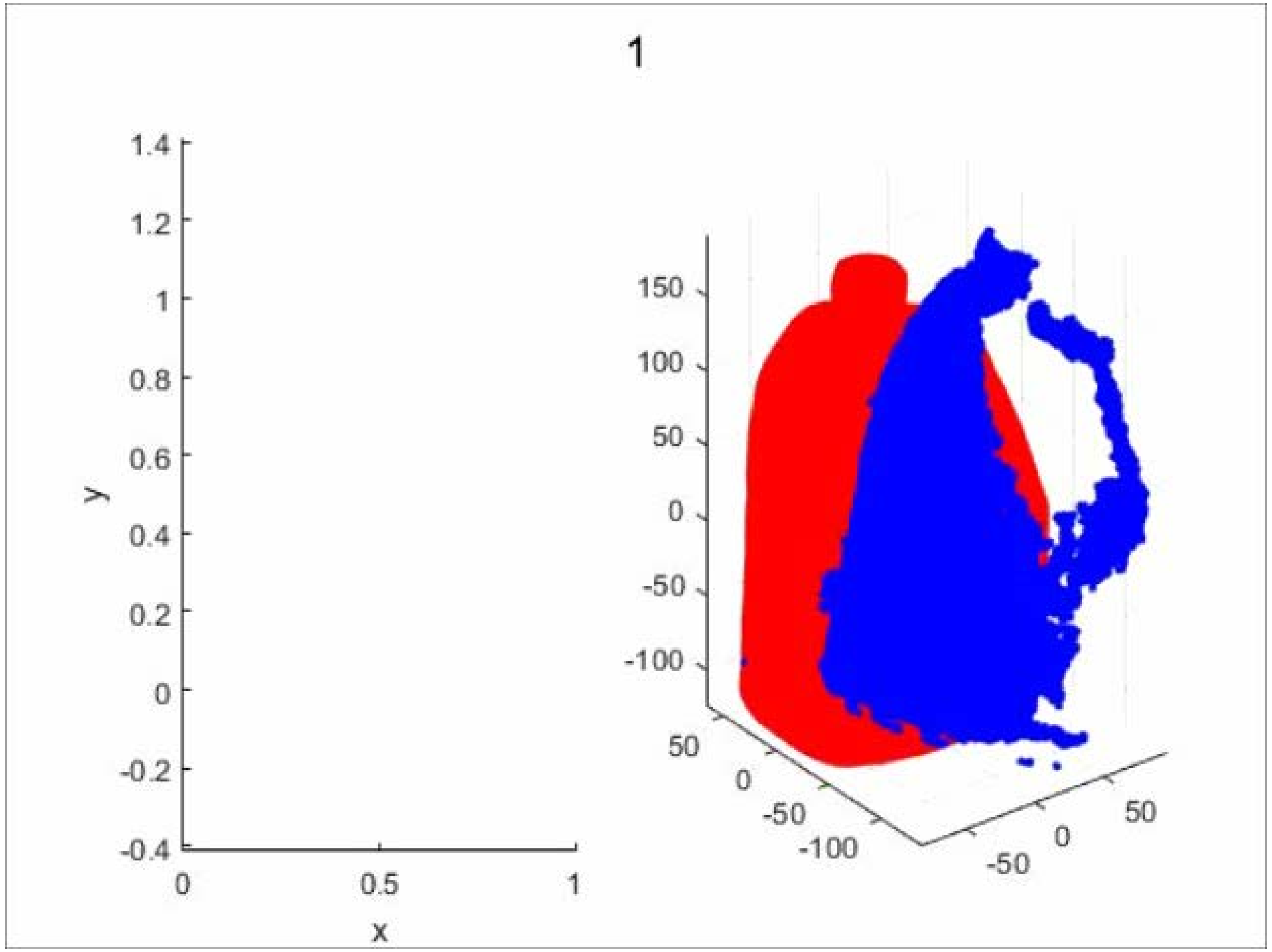
$$H = U \Lambda V^T$$

$$X = V U^T$$

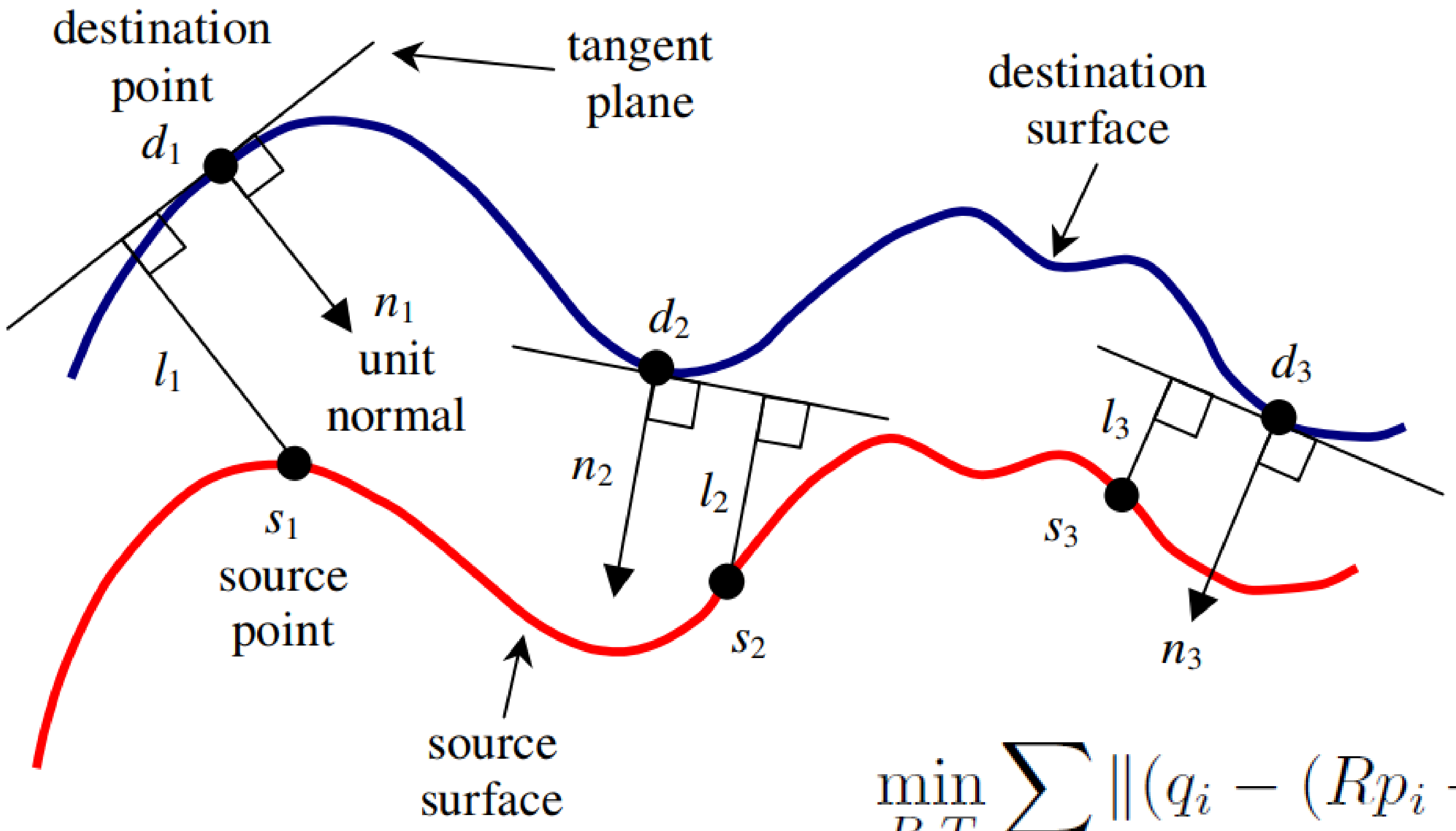
$$\text{If } \det X = 1 \Rightarrow \hat{R} = X.$$

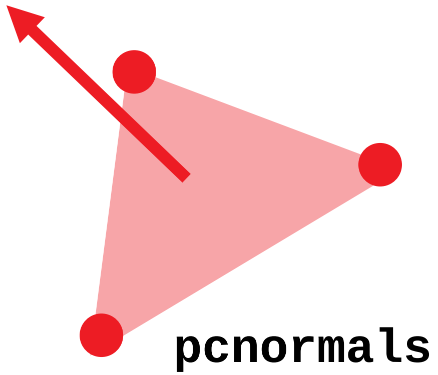
$$\text{If } \det X = -1 \Rightarrow, \text{ the algorithm failed.}$$

Point to Point Iterative Closest Point P2P-ICP



Point to Plane Iterative Closest Point
P2Pl-ICP

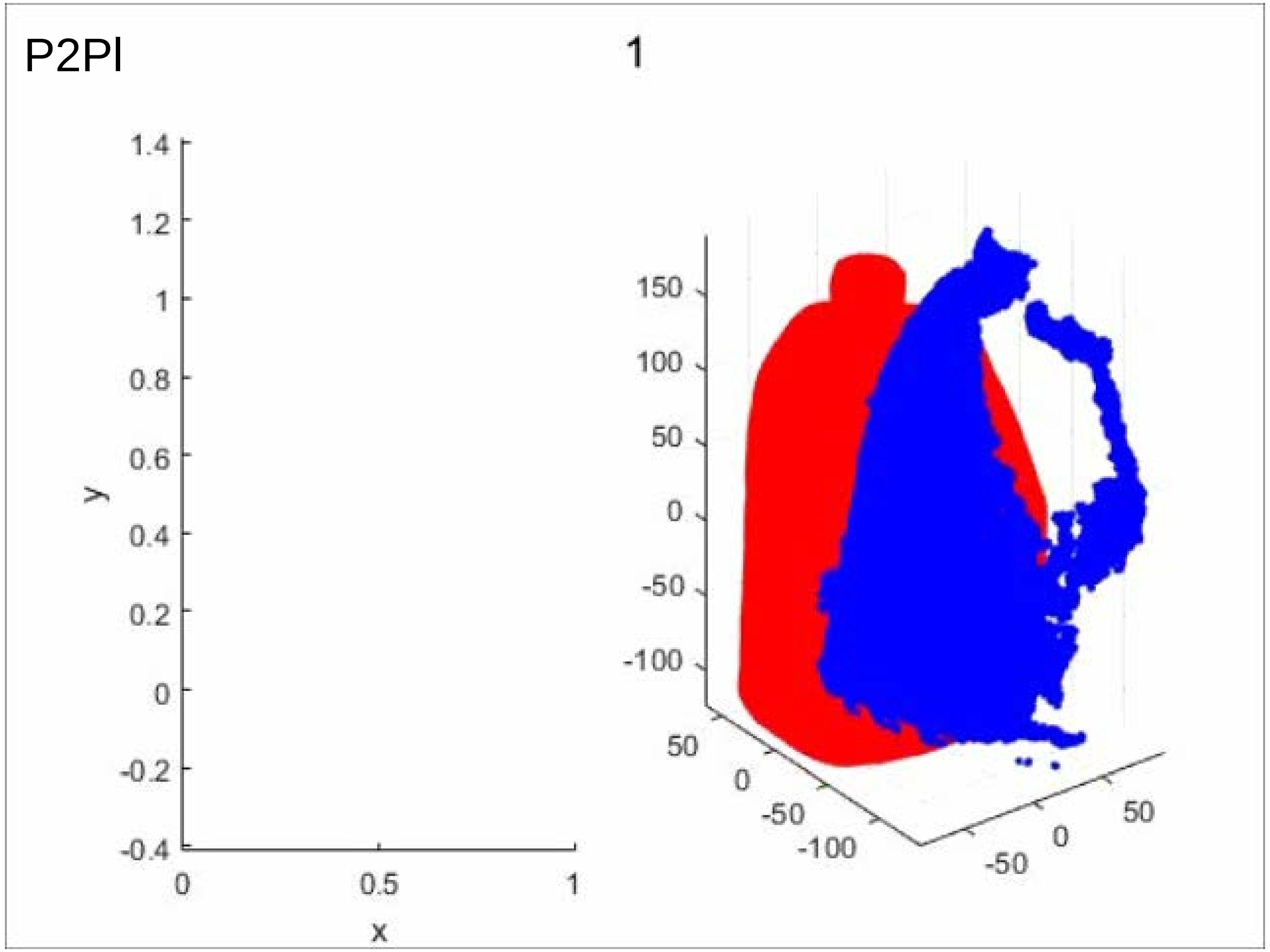


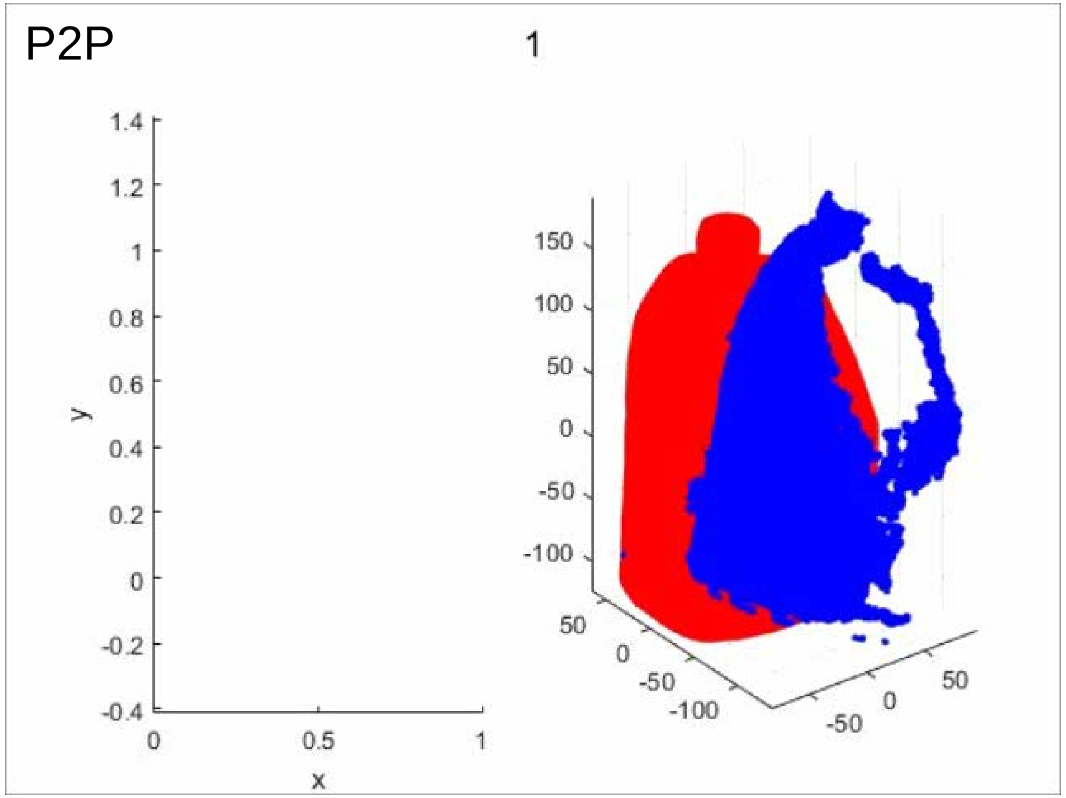


$$\min_{R,T} \sum_{\forall i} \| (q_i - (Rp_i + T)) \bullet n_i \|^2$$

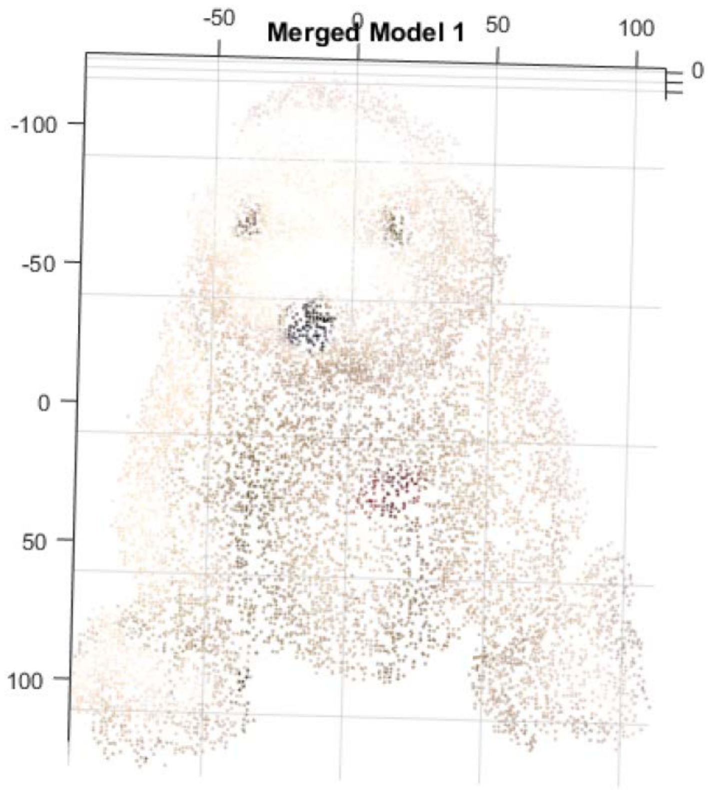
Low, Kok-Lim. "Linear least-squares optimization for point-to-plane icp surface registration." *Chapel Hill, University of North Carolina* 4 (2004).

Point to Plane Iterative Closest Point
P2Pl-ICP





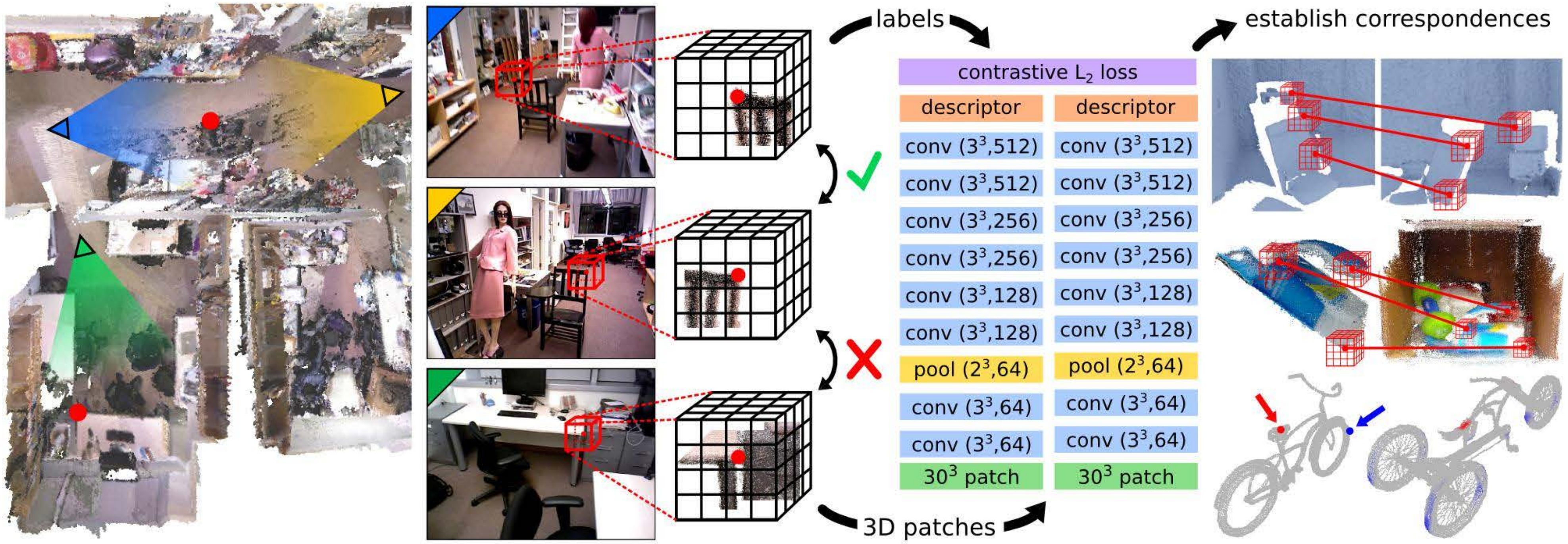
Reconstructed Model



Option 1: Segmenting Scene

Option 1: Segmenting Scene

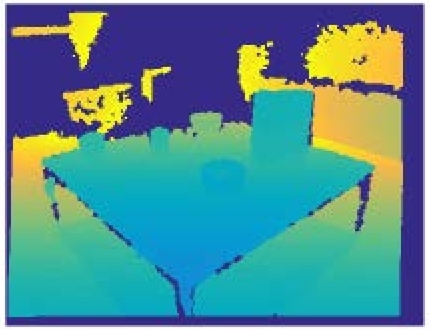
Use ICP to match single object point cloud to 3D scene.
Use 3D Match to match in 3D.

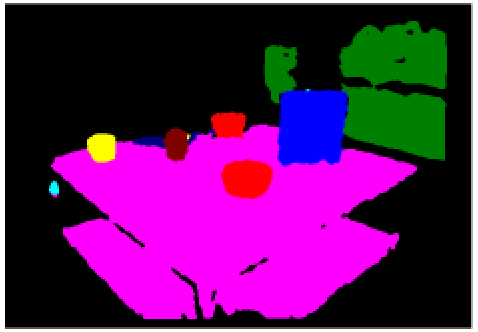


(a) RGB-D reconstructions (b) frames (c) local patches (d) siamese network (e) applications

http://3dmatch.cs.princeton.edu

Option 2: Semantic Map





Teddy is bigger than penguin by 18% in volume.
Teddy is to the right of penguin.

Xiang, Yu, and Dieter Fox. "DA-RNN: Semantic mapping with data associated recurrent neural networks." *arXiv preprint arXiv:1703.03098* (2017).

Thank you!