

Learning To Grasp

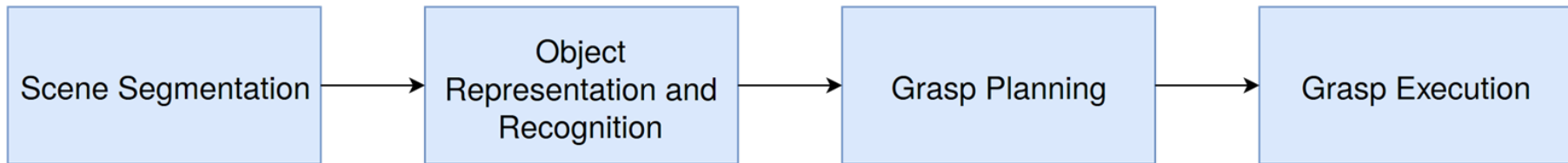
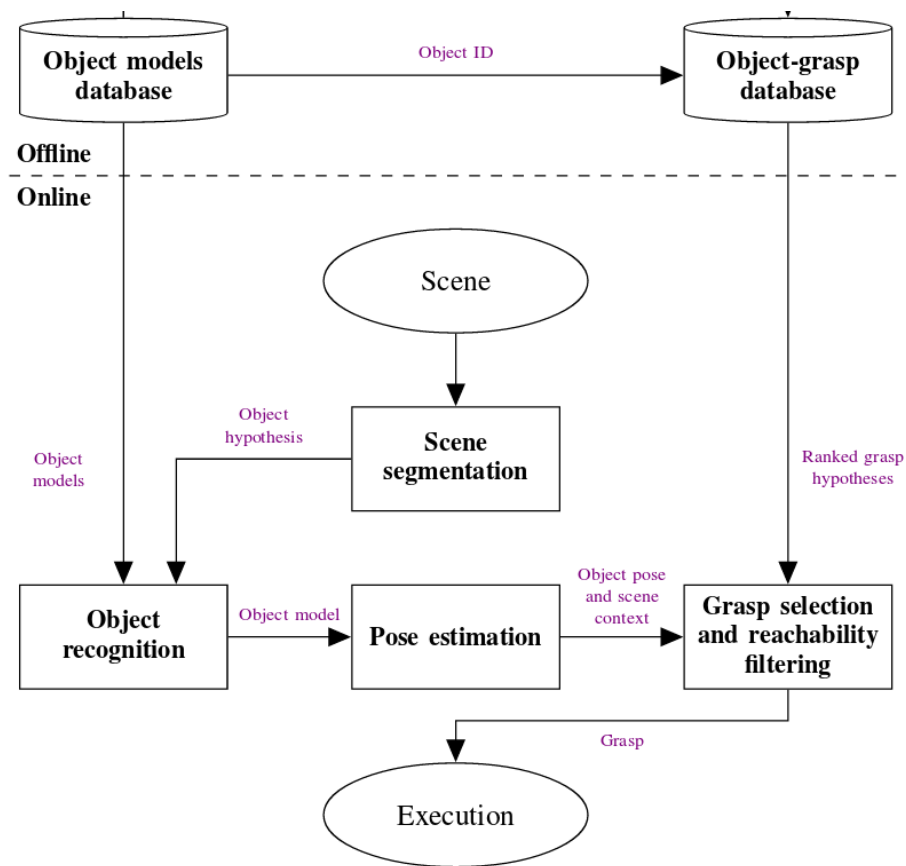
Jake Varley

Overview

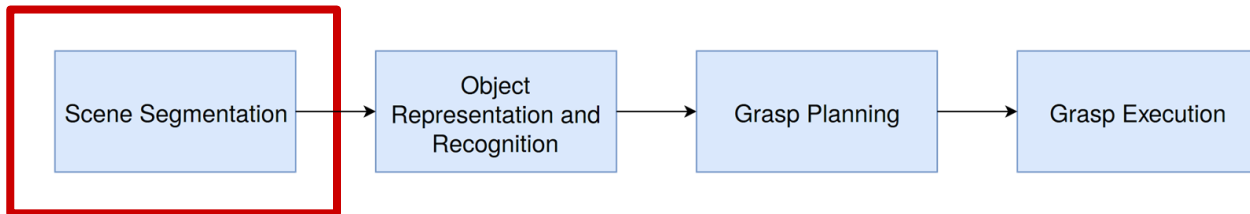
- What is a grasping pipeline?
- A current grasping pipeline
- Recent trends in related fields
- A future grasping pipeline

A Grasping Pipeline

Data Driven Grasp Synthesis - a survey 2014
<https://arxiv.org/pdf/1309.2660v2.pdf>



Scene Segmentation



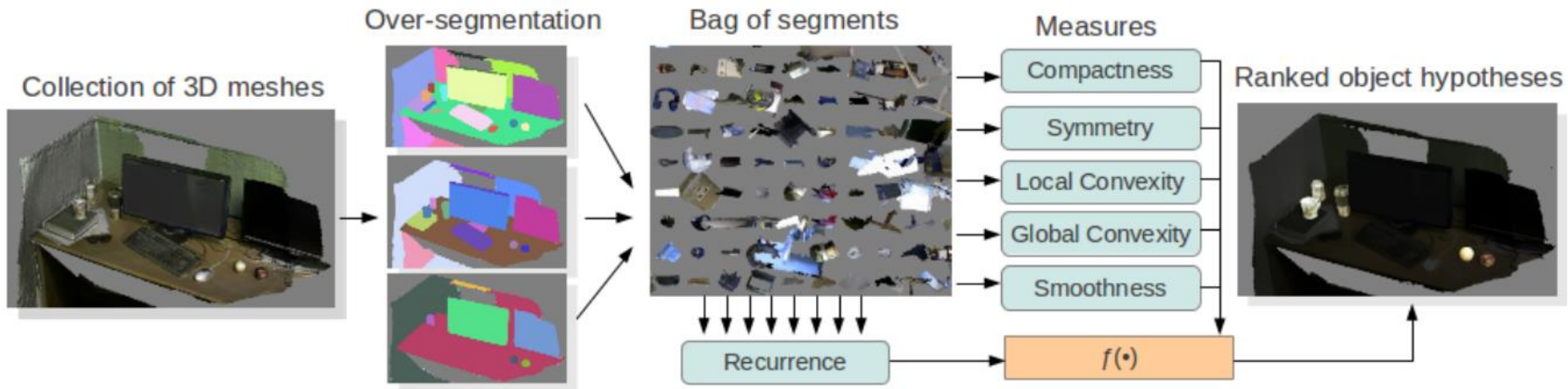
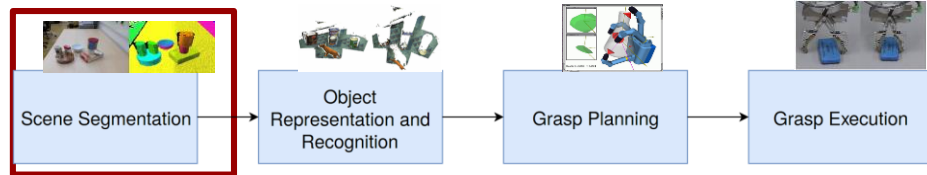
Need to understand what we are going to interact with...

- Euclidean Clustering
- Object Detector



Image From:
Andrej Karpathy, et al. Object discovery in 3d scenes via
shape analysis. ICRA, 2013

Object Discovery in 3D scenes via shape analysis



- Segment 58 scenes using several thresholds
- Train an SVM on 6 handcrafted features to predict whether each segment is an **object** or not?

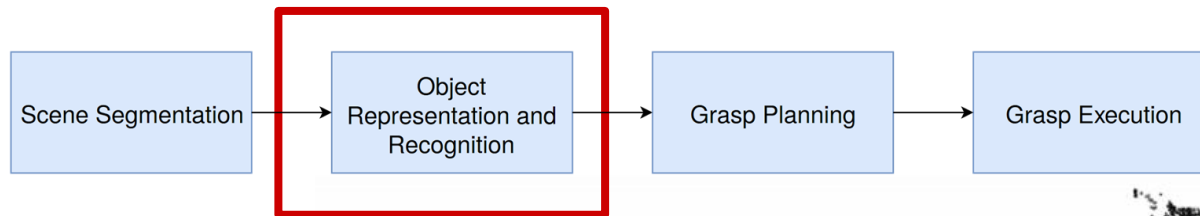
Pros:

- Easy to understand
- Fast

Cons:

- Objectness is vague
- Not dense
- Handcrafted features

Object Modeling



We have a segmented scene, and a region of interest, but the back half is missing....

- General Completion
- Instance Recognition



Exploiting Symmetries and extrusions for grasping household objects

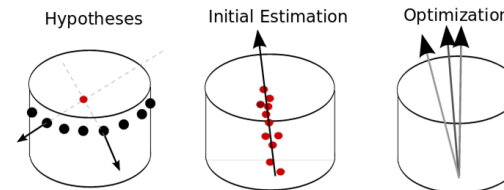
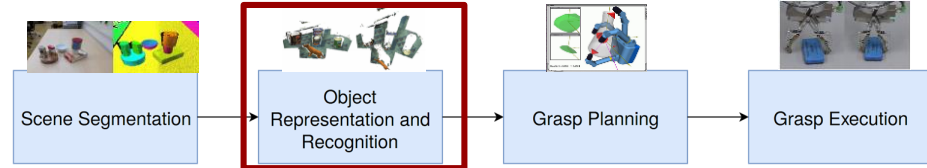
- Reflect points over symmetry plane
- Determine best linear or revolute extrusion for mirrored points.

Pros:

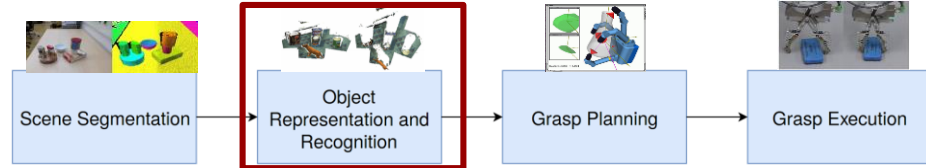
- Many objects exhibit symmetry

Cons:

- Just a heuristic



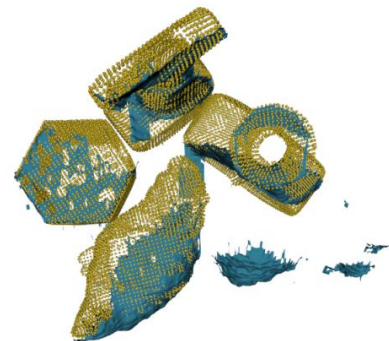
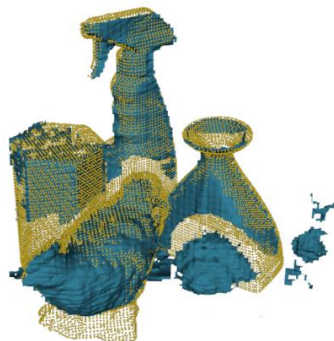
An efficient ransac for 3d object recognition in noisy and occluded scenes



- Database of Object Instances

- Find them in the scene

- Hashtable: pairs of oriented points to model pose.



- Randomly sample hypothesis

- Filter based on evidence

and agreement with visible scene

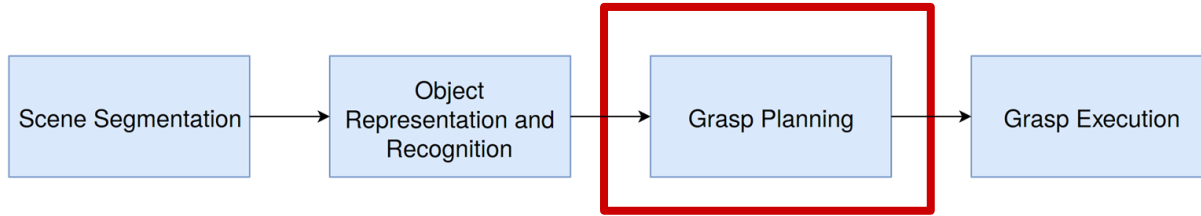
Pros:

- Fast cuda implementation

Cons:

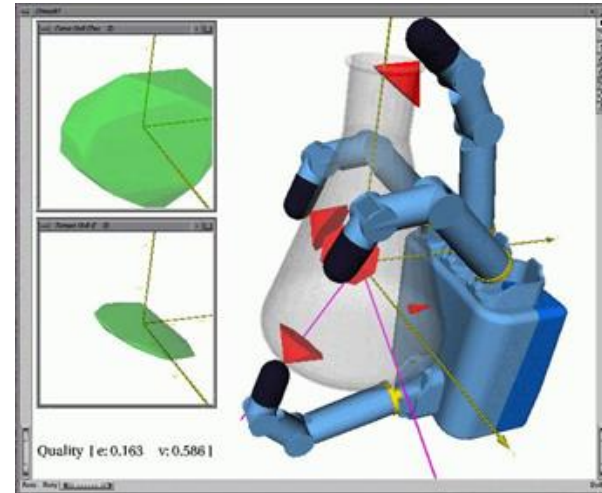
- Exact model matching
- Lots of magic numbers
- Tens of objects only

Grasp Planning

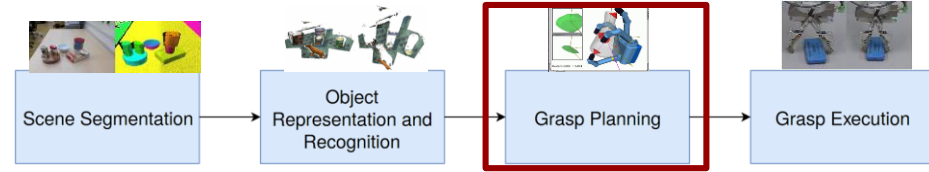


We have a segmented scene, and a completed object to grasp, but how should we pick it up...

- Search for a Grasp
- Precompute a database of grasps
- Grasping rectangles for simple grippers



Hand Posture subspaces for dexterous robotic grasping























- Eigengrasps: First two principal components account for more than 80% of the variance
- Search for “Good” grasps in Eigengrasp space

Pros:

- Reduced dimensionality allows for fast search

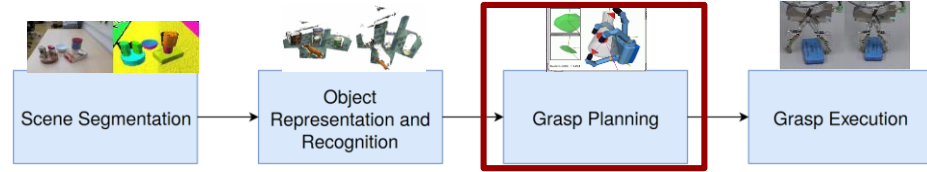
Cons:

- Heuristic energy functions
 - Volume Energy

Model	DOFs	Eigengrasp 1			Eigengrasp 2		
		Description	min	max	Description	min	max
Gripper	4	Prox. joints flexion			Dist. joints flexion		
Barrett	4	Spread angle opening			Finger flexion		
DLR	12	Prox. joints flexion Finger abduction			Dist. joints flexion Thumb flexion		
Robonaut	14	Thumb flexion MCP flexion Index abduction			Thumb flexion MCP extension PIP flexion		
Human	20	Thumb rotation Thumb flexion MCP flexion Index abduction			Thumb flexion MCP extension PIP flexion		

Grasplt! Demo

Data-driven grasping



Step 1: Creating a grasp database of 3D models annotated with precomputed grasps and quality scores.

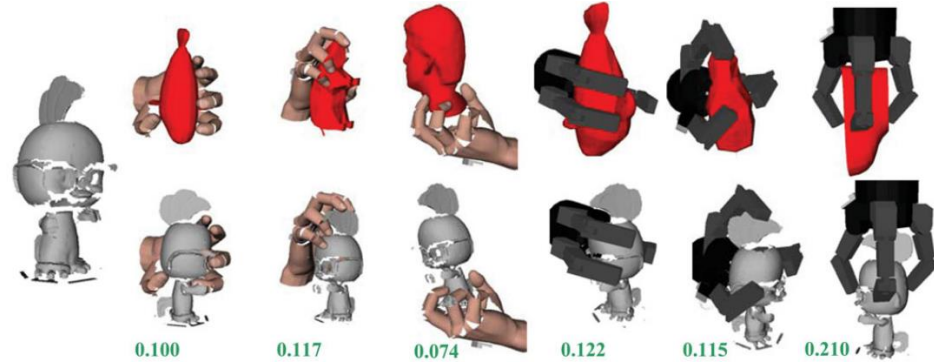
Step 2: Indexing the database for retrieval using partial 3D geometry.

Step 3: Finding matches in the database using only the sensor data, which is typically incomplete.

Step 4: Aligning the object to each of the matched models from the database.

Step 5: Selecting a grasp from the candidate grasps provided by the aligned matches

Step 6: Executing the grasp and evaluating the results.



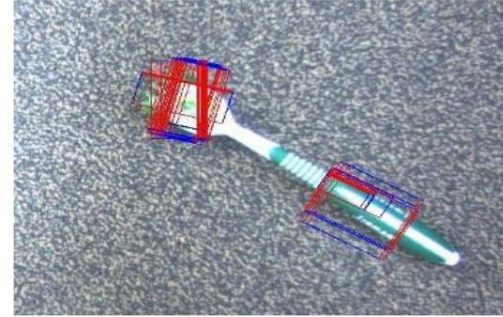
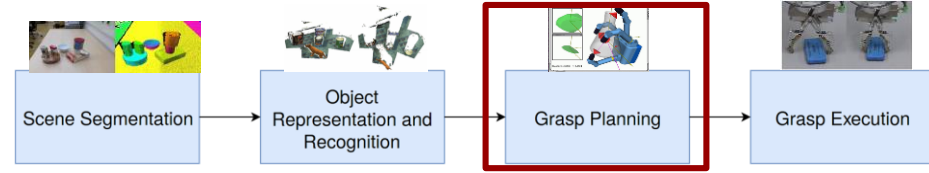
Pros:

- Data Driven, not a heuristic

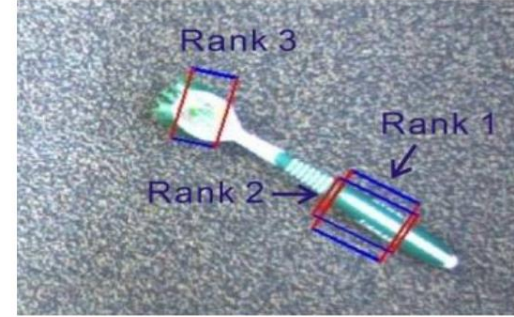
Cons:

- Grasp transfer is rigid

Efficient Grasping from RGBD Images: Learning using a new rectangle representation



(a) The top 100 rectangles after the first-step search.



(b) The top 3 rectangles after the second-step rank.

Fig. 6: Example results from two-step search.

Gripper pose is 5 DOF

- x, y , width, height, theta

Search:

- Quick first pass search for candidates
- More advanced features to rank candidates

Pros:

- Data Driven

Cons:

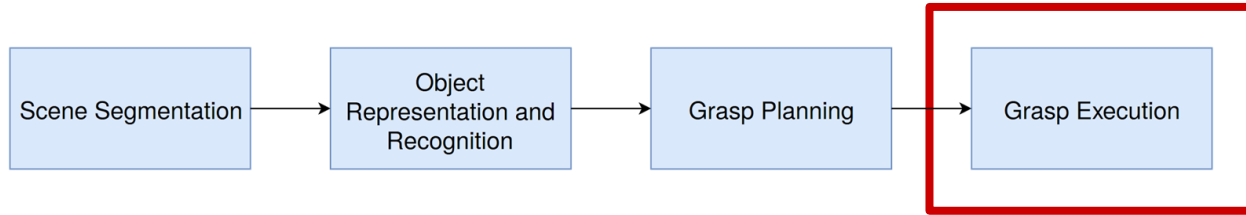
- All grasps are from above

Yun Jiang, Stephen Moseson, and Ashutosh Saxena.

Efficient grasping from rgbd images: Learning using a new rectangle representation.

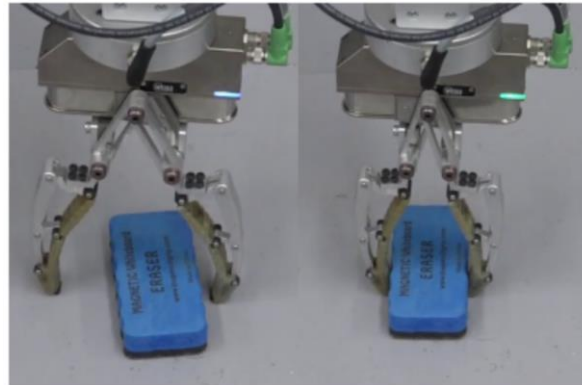
In Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011

Grasp Execution

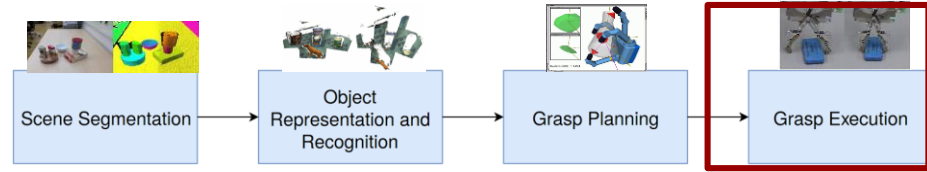


We have a segmented scene, a completed object, and a planned grasp, but how do we execute it?...

- Open Loop Grasp Execution



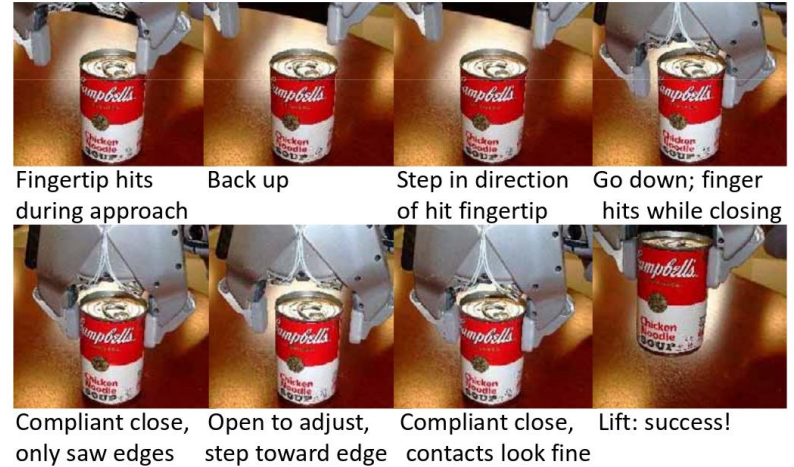
Grasp Execution



- Open Loop Grasp Execution is still mainstream
- No out of the box working solutions using feedback in general use
- Closest:

Hsiao, K., Chitta, S., Ciocarlie, M., & Jones, E. G.. Contact-reactive grasping of objects with partial shape information. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*

Dang, Hao, and Peter K. Allen. "Stable grasping under pose uncertainty using tactile feedback." *Autonomous Robots* 36.4 (2014)



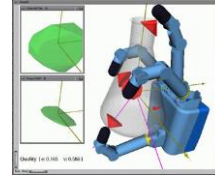
Pros:

- Able to integrate feedback

Cons:

- React poorly if object is perturbed in the process
- No vision
- heuristic

A Prototypical Grasping Pipeline: Now



Scene Segmentation

Object
Representation and
Recognition

Grasp Planning

Grasp Execution

- Euclidean Cluster Extraction
- Object Discovery

- RANSAC Instance matching
- Symmetry based completion

- Grasp Database
- Anneal through C_{space} via grasp quality heuristic
- Grasping Rectangles

- Open Loop Grasp Execution

General Problems:

Heuristics, Hand Crafted Features, Overly constrained, Small datasets, Little Sensory Feedback

How To Move Forward

- Problems: Heuristics, Hand Crafted Features, Overly Constrained, Small Datasets, Little Sensory Feedback
- Massive improvements in tangential fields in last 3 years:
 - **Big Data:** Significantly more available training data
 - **Simulation:** RGBD Rendering, Maintained contact during physics simulations
 - **Deep Learning:** Powerful classifiers
- Many of these improvements are being leveraged to alleviate current problems in grasping.

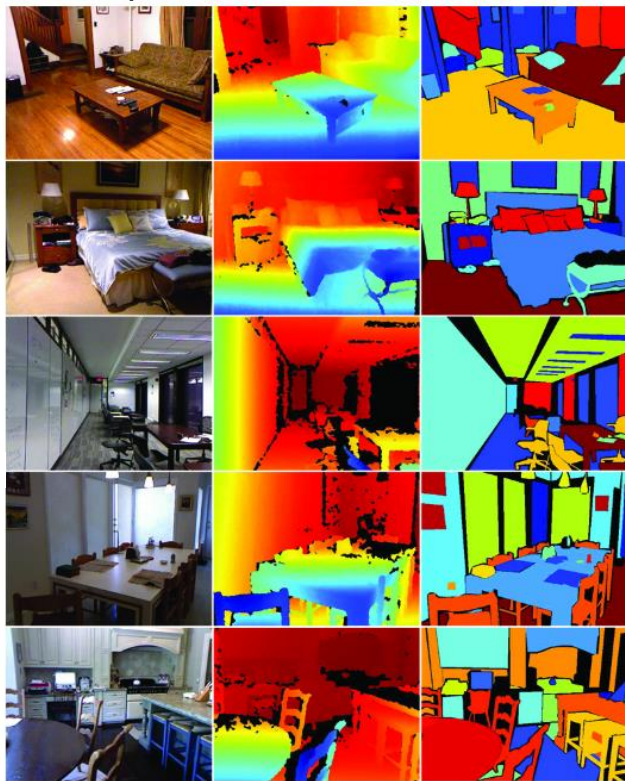
Big Data

Many of the approaches shown are heuristics validated on very small datasets.

- Are heuristics that work for these small dataset really representative?
- Difficult to develop data driven approaches if the data doesn't exist

Big Data

NYU Depth V2



RGBD from Kinect

1449 densely labeled pairs of aligned RGB and depth images

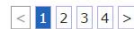
407,024 new unlabeled frames

Each object is labeled with a class and an instance number (cup1, cup2, cup3, etc)

40 Categories

ShapeNet

Displaying 1 to 160 of 498



- 3 Million Models
- 220,000 categorized into 3135 categories (WordNet synsets)

Simulation

Part of the reason large datasets are slow to come into existence is because it requires a large amount of effort:

- Sensors change
- Takes time
- Often difficult to label ground truth

Simulation

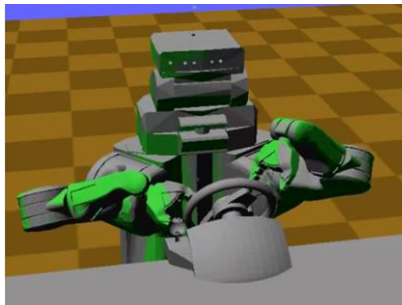
1) Embree: photo realistic rendering

Wald, Ingo, et al. "Embree: a kernel framework for efficient CPU ray tracing." *ACM Transactions on Graphics* 33.4 (2014).



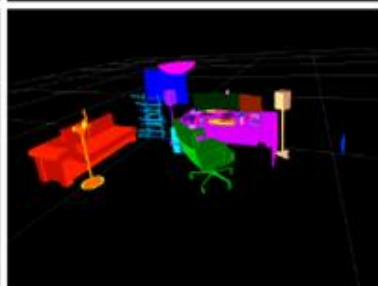
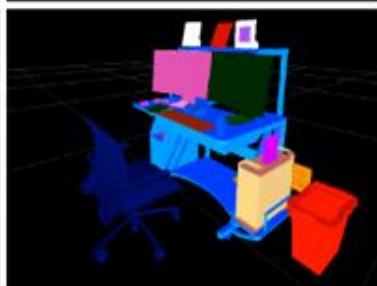
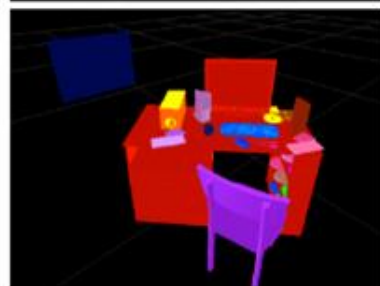
2) SceneNet: scene generation

Handa, Ankur, et al. "Scenetnet: Understanding real world indoor scenes with synthetic data." *arXiv preprint* (2015).



3) KlampT: contact simulation

Hauser, Kris. "Robust contact generation for robot simulation with unstructured meshes." *Robotics Research*. 2016.



Deep Learning

How to do data driven robotics:

- Before:
 - hand crafted features
 - Small datasets that work well with those features
- Now:
 - Let the network learn features from lots of data
 - Generate lots of data
 - Determine a good representation of the data

Deep Learning

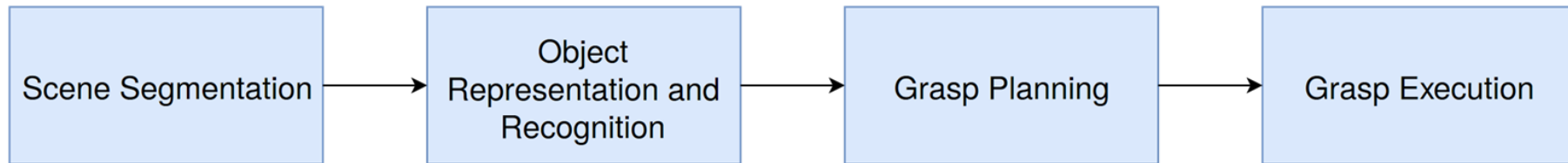
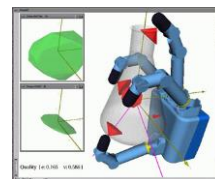
- ImageNet Challenge started in 2010:
 - 2012 winning team used deep learning
 - No Image Classification Task since 2014. Too easy
 - 2016:
 - [Object localization](#) for 1000 categories.
 - [Object detection](#) for 200 fully labeled categories.
 - [Object detection from video](#) for 30 fully labeled categories.
 - [Scene classification](#) for 365 scene categories
 - [Scene parsing](#)^{New} for 150 stu

- Nvidia Tesla K80 24GB g
 - 3D Convolutions



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

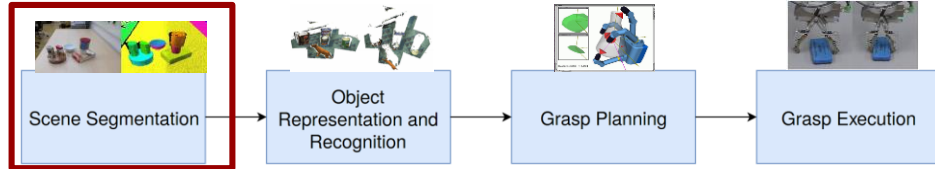
<http://xkcd.com/1425/>
From 9/24/2014



A Prototypical Grasping Pipeline: 5 Years from now

- Dense RGBD Per Pixel Semantic Labeling
- Data driven scene and shape completion
- Learned grasp quality derived from simulation
- Learned closed loop torque control using visual and tactile feedback

Scene Segmentation



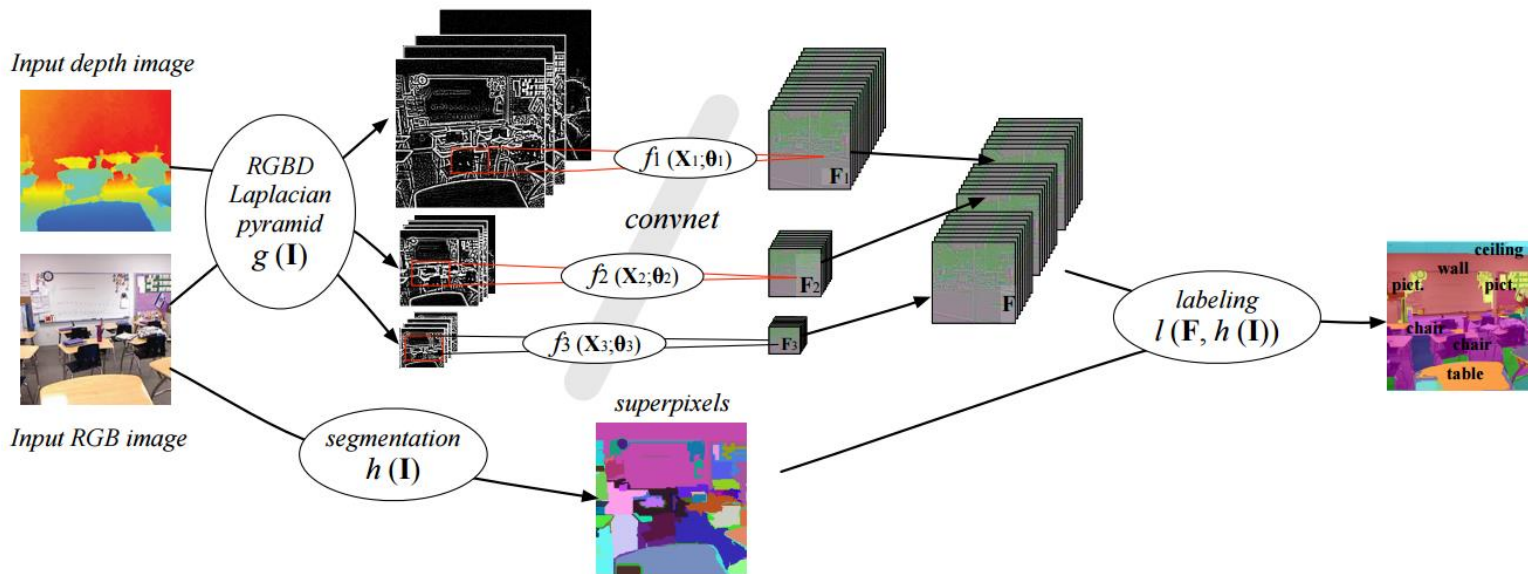
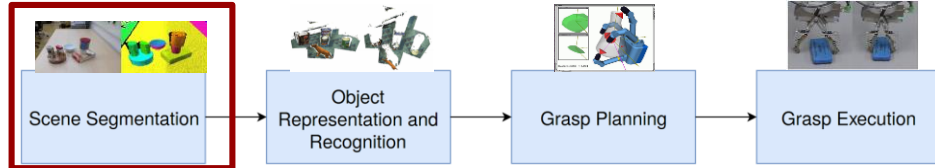
Before:

- Objectness detector
- PCL Euclidean cluster extraction

Now:

- Semantic per pixel/voxel/surflet labeling
- Powered by algorithms developed for ImageNet adapted to NYU-Depth V2 dataset.

Indoor Semantic Segmentation

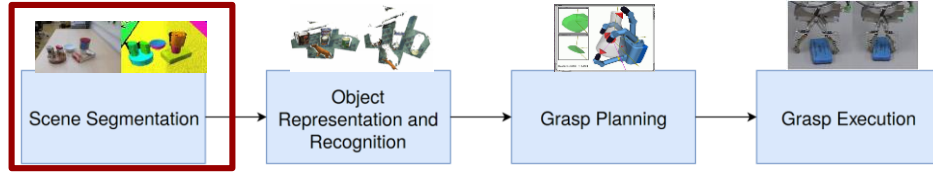


Coupric et al (NYU 2013): 52.4% per pixel accuracy with 16 categories

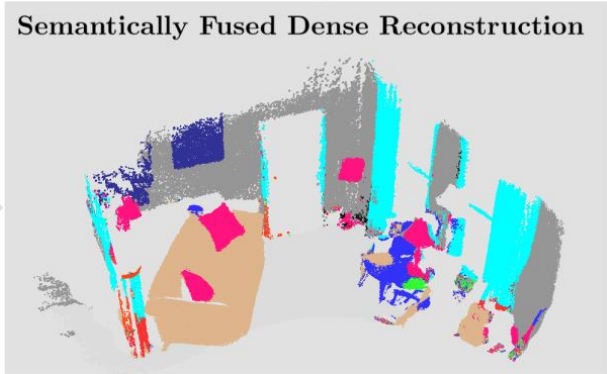
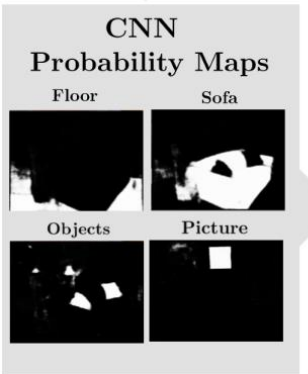
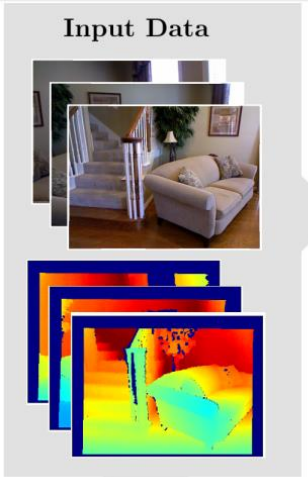
Long et al (UC Berkeley 2015): 65% per pixel accuracy with 40 categories

- Camille Coupric, Clement Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. arXiv preprint arXiv:1301.3572, 2013
- Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015

Semantic Fusion

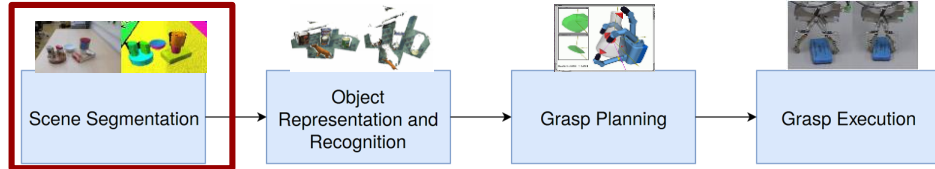


- Elastic Fusion Slam
- RGBD-CNN for per pixel labels
- Project pixels to surfels
- Bayesian update for per-surfel semantic label estimate



John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semantic fusion: Dense 3d semantic mapping with convolutional neural networks. arXiv preprint arXiv:1609.05130, 2016

Scene Segmentation

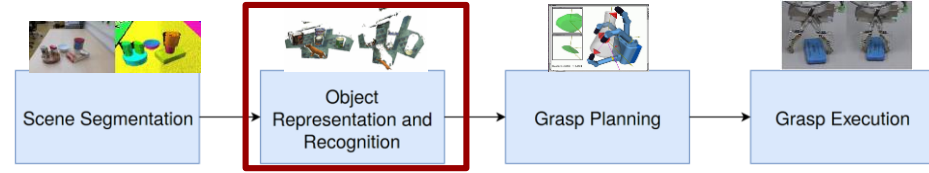


Deep Learning enabled per pixel labeling is here:

- Pros:
 - Dense labels
 - Semantic labels
 - Incredibly fast
- Current Hurdles:
 - We need more data! (Especially rgbd data)
 - Object category is not interesting enough for robotics
 - Sensors improve and old datasets lose utility.

Sensor limitations: Transparent and Reflective materials

Object Modeling



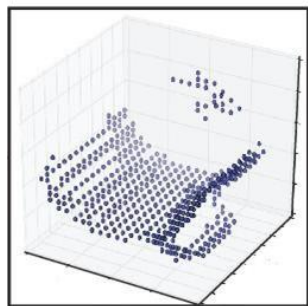
Before:

- Exact model matching approaches
- Simple symmetry and extrusion approaches for general completion

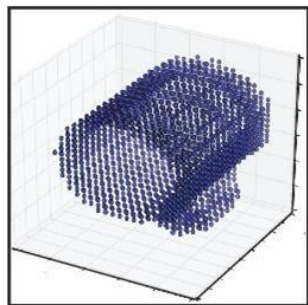
Now:

- Data Driven techniques for general completion

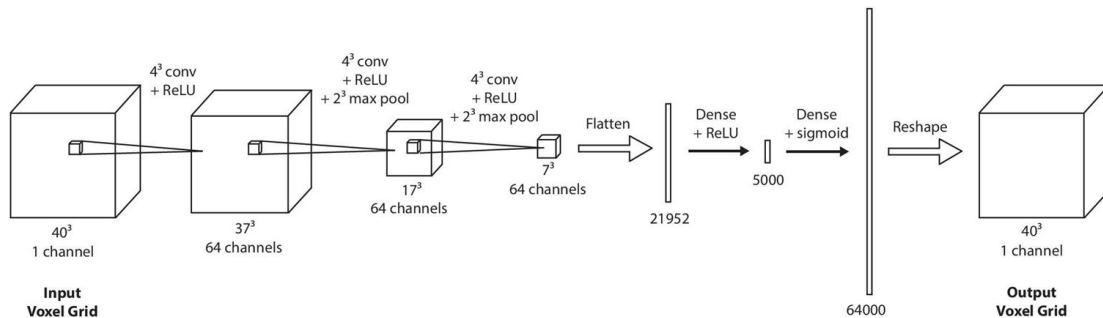
Shape Completion Enabled Robotic Grasping



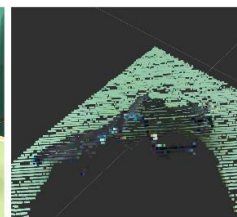
X



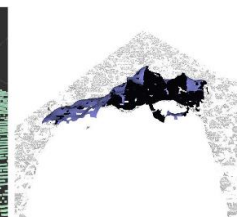
Y



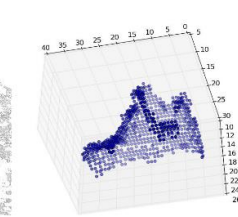
(a) Image of Occluded Side



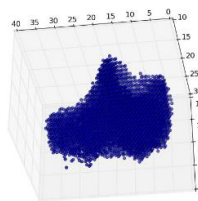
(b) Point Cloud



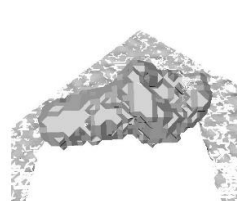
(c) Segmented and Meshed



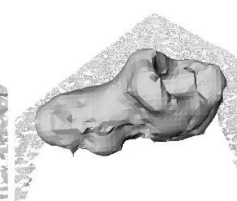
(d) CNN Input



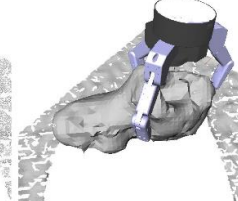
(e) CNN Output



(f) Mesh

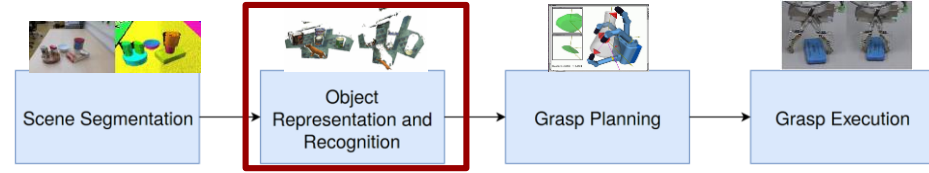


(g) Smoothed Mesh



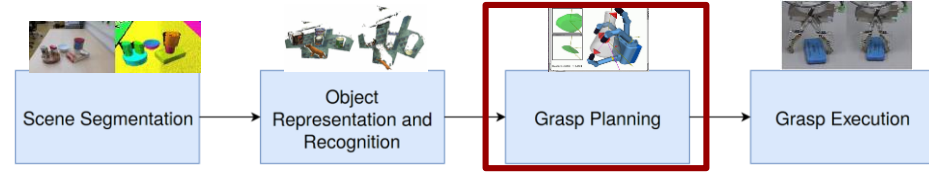
(h) Grasp Planning

Object Modeling



- Yet to be a large scale dataset for general scene completion similar to NYU-Depth V2 dataset for semantic segmentation
- The 3d models exist:
 - ShapeNet: 3,000,000 models, 220,000 models out of which are classified into 3,135 categories

Grasp Planning



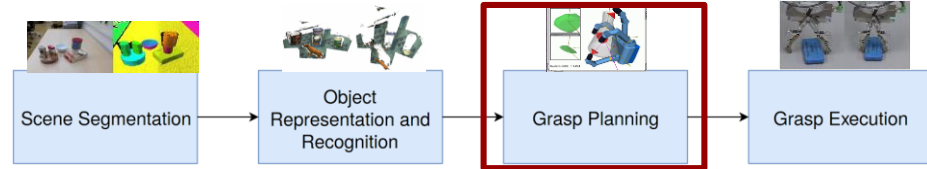
Before:

- Search in low dimensional space via handcrafted quality functions
- Database of objects and corresponding grasps
- Data driven parallel jaw grasps

Now:

- Deep Learning for data driven quality functions
- Simulated grasp executions to label training grasps

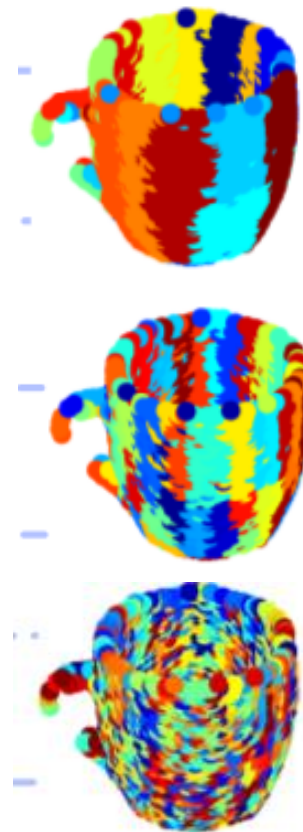
Hierarchical Fingertip space for multi-fingered precision grasping



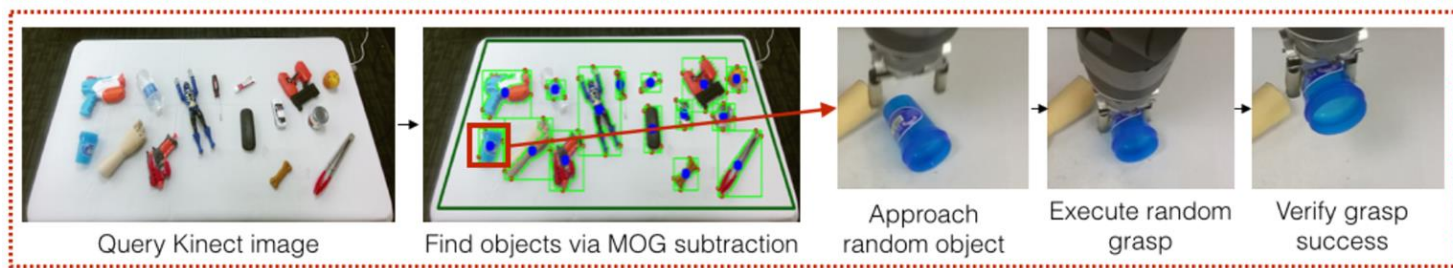
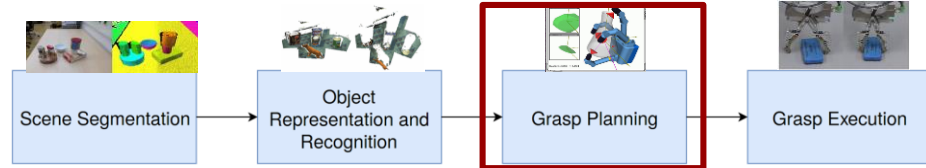
(A) extract a hierarchical fingertip space

(B) Using the fingertip space hierarchy and reachability, search for contacts and initial hand configuration.

(C) grasp realized by local contact positions optimization with respect to the synthesized contacts



Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours.



- RGB CNN
- 18 way binary classification
- 73% accuracy

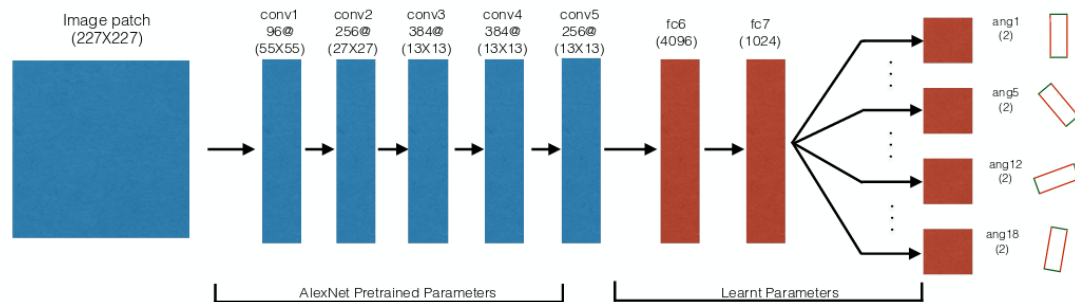
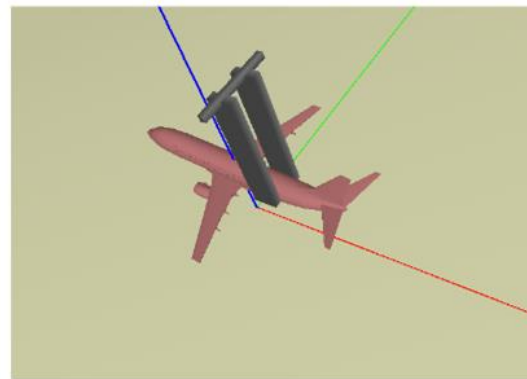
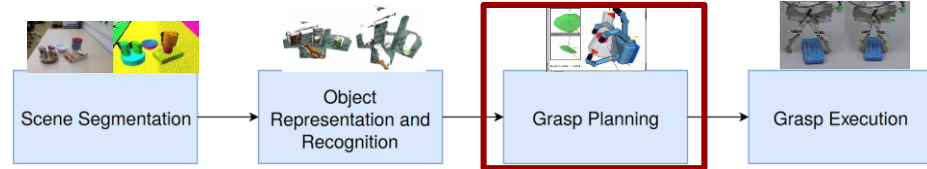


Fig. 5. Our CNN architecture is similar to AlexNet [6]. We initialize our convolutional layers from ImageNet-trained Alexnet.

Deep Learning a grasp function for grasping under gripper pose uncertainty

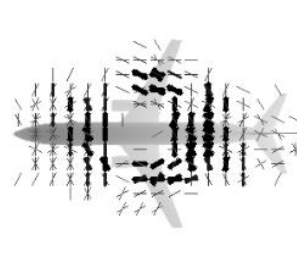
- Parallel Jaw Gripper grasps
- Supervised learning approach
- Training data evaluated with physics in simulation with gravity
- 80.3% accuracy



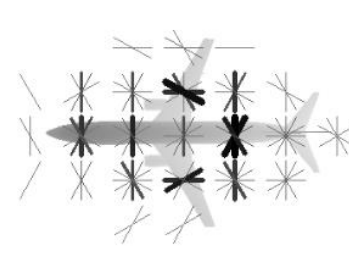
(a) Simulated depth image



(b) Simulated depth image after adding the noise model

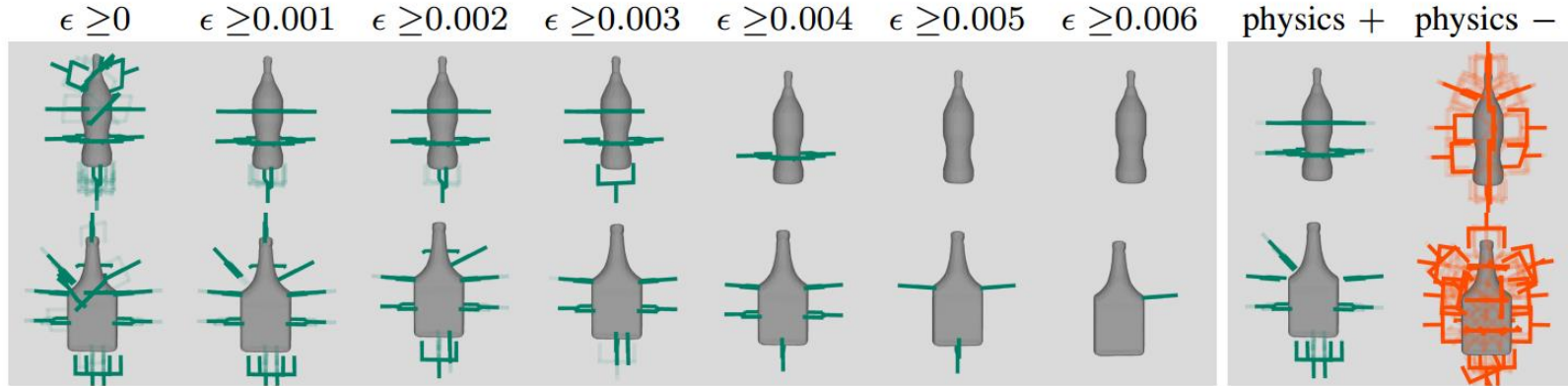
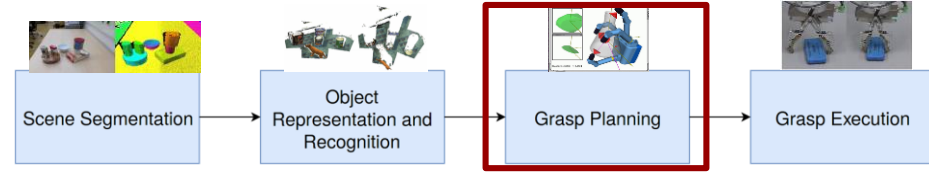


(c) Visualisation of the grasp function



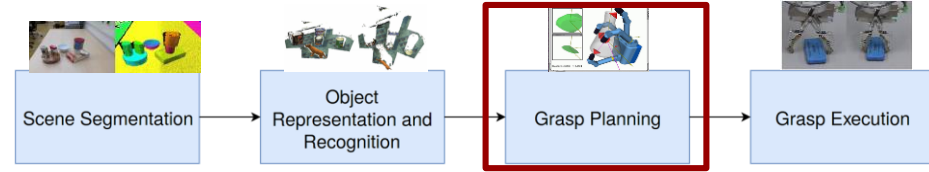
(d) Visualisation of the grasp function at a coarser scale

Leveraging big data for grasp planning



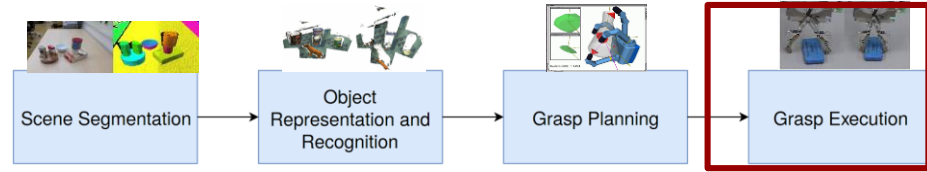
- Large scale database of parallel jaw gripper grasps
- Crowdsourcing show physics simulation better predictor to grasp success than epsilon-metric
- Train CNN to recognize good grasp locations

Grasp Planning



- Pros
 - Parallel Jaw Gripper from above is done
- Current Hurdles:
 - Higher dimensional grasp planning problems
 - Can we extend grasping rectangles?
 - Do we just need more efficient search algorithms?
- Possible Solution:

Grasp Execution



Before:

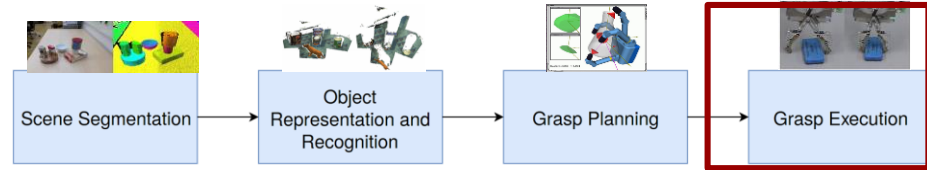
- Open Loop Grasp Execution

Now:

- Deep learning enabled mapping from sensory information to movement

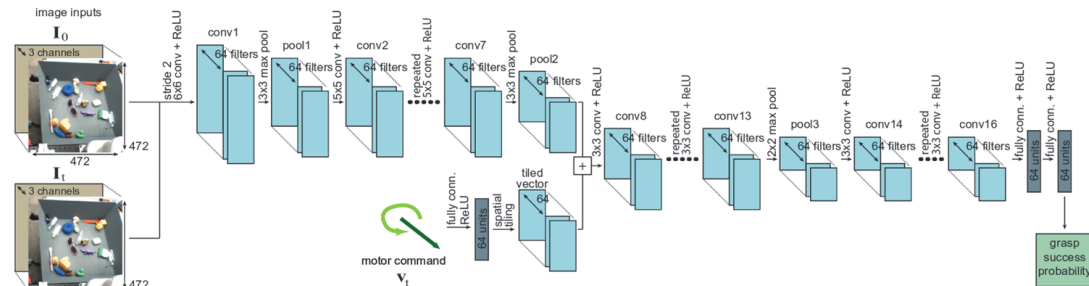
Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection

- CNN to predict the probability that task-space motion of the gripper will result in a successful grasp
- Servoing algorithm powered by the CNN



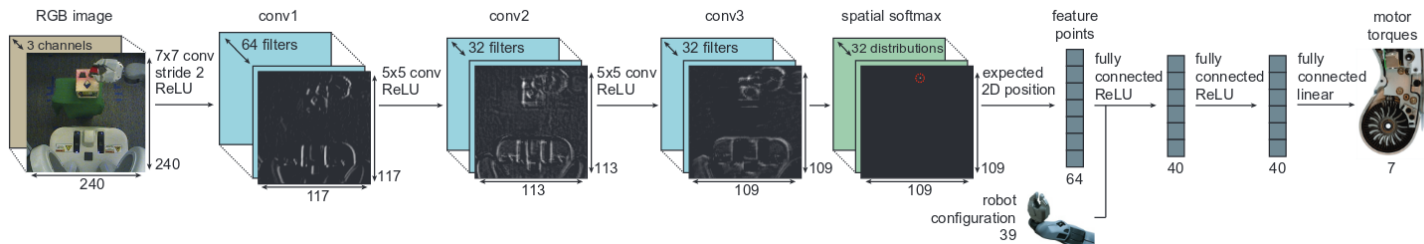
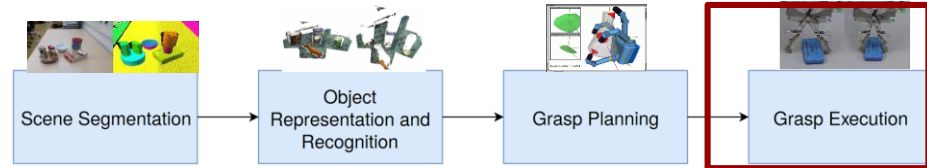
Algorithm 1 Servoing mechanism $f(\mathbf{I}_t)$

- 1: Given current image \mathbf{I}_t and network g .
- 2: Infer \mathbf{v}_t^* using g and CEM.
- 3: Evaluate $p = g(\mathbf{I}_t, \emptyset) / g(\mathbf{I}_t, \mathbf{v}_t^*)$.
- 4: **if** $p > 0.9$ **then**
- 5: Output \emptyset , close gripper.
- 6: **else if** $p \leq 0.5$ **then**
- 7: Modify \mathbf{v}_t^* to raise gripper height and execute \mathbf{v}_t^* .
- 8: **else**
- 9: Execute \mathbf{v}_t^* .
- 10: **end if**

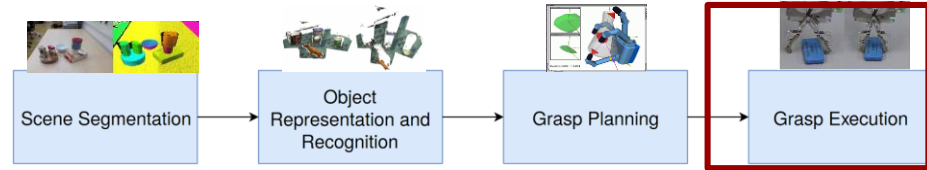


End-to-end training of deep visuomotor policies

- CNN mapping raw images -> torques
- Train several Linear Gaussian Controllers choose an action given full scene info (exact object pose, end effector pose). Different LGC for different start configurations
- Training a CNN to replication the linear gaussian controllers. Using observations (image, + encoder values)



Towards Adapting Deep Visuomotor Representations from Simulated to Real Environments



An initial step toward pretraining deep visuomotor policies entirely in simulation, significantly reducing physical demands when learning complex policies

3 Loss terms:

- 1) Standard pose estimation loss
- 2) Domain confusion loss to align the synthetic and real domains in feature space.
- 3) contrastive loss to align specific pairs in feature space

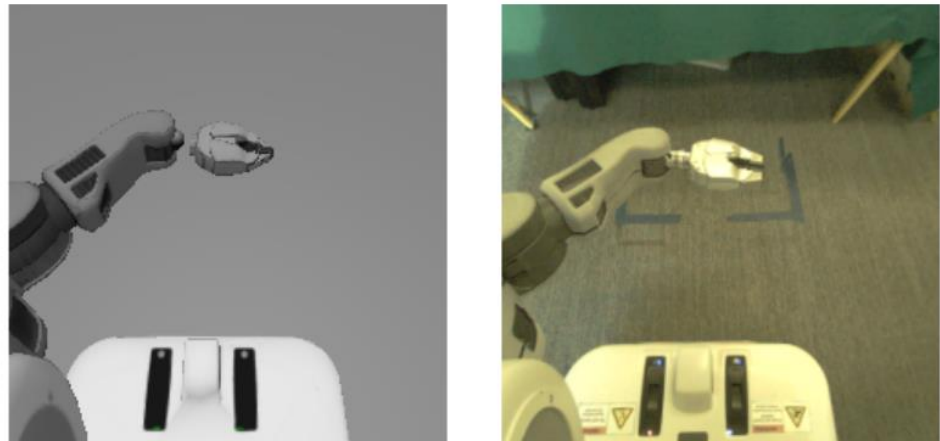
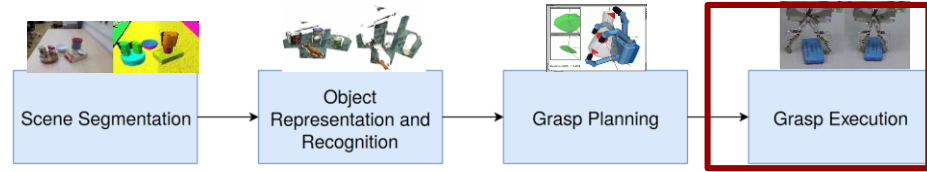


Fig. 3. A pair of corresponding synthetic (left) and real-world (right) images used for our pose estimation evaluation. The task is to predict the 3D pose of the PR2's gripper.

Grasp Execution



- Pros

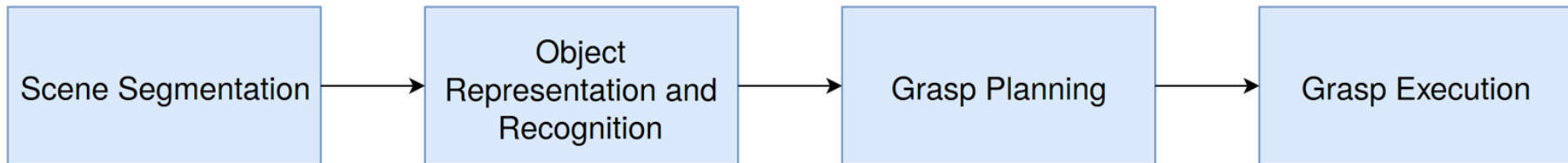
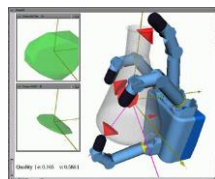
- Simple Controllers possible to train

- Current Hurdles:

- Each controller learns to reach a very specific goal

- Possible Solutions

- Have lots of controllers for different canonical grasp types for different objects.
 - Train them in simulation



A Prototypical Grasping Pipeline: Now

- Object Discovery
- Euclidean Cluster Extraction
- RANSAC Instance matching
- Symmetry based completion
- Grasp Database
- Anneal through C_{space} via grasp quality heuristic
- Grasping Rectangles
- Open Loop Grasp Execution

A Prototypical Grasping Pipeline: 5 Years from now

- Dense RGBD Per Pixel Semantic Labeling
- Data driven scene and shape completion
- Learned grasp quality derived from simulation
- Learned closed loop torque control using visual and tactile feedback

Future Research Directions

- Segmentation:
 - No per pixel labeled rgb-d dataset geared towards robotics
- Object Modeling:
 - No large scale dataset with partial observations and ground truth model with pose
- Grasp Planning:
 - Multi-fingered grasp planning integrating:
 - Anneal over Hierarchical Fingertip or EigenGrasp space