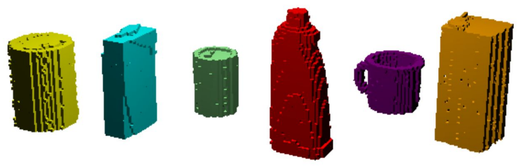
Few-shot Bayesian inference of 3D objects and scenes via probabilistic programs

Nishad Gothoskar, Ben Zinberg, Marco Cusumano-Towner, Falk Pollok,
Joshua Tenenbaum, Dan Gutfreund, Vikash Mansinghka
MIT IBM

1. Overview

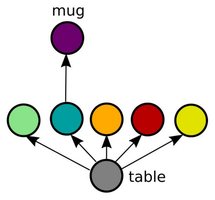
We propose a generative probabilistic programming–based architecture for modeling 3D objects and scenes, and use our architecture to do accurate and robust object pose estimation from RGBD images.

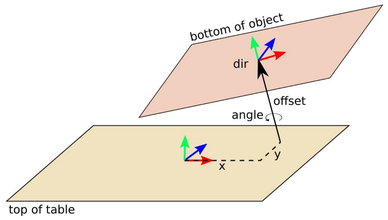
Objects are represented as distributions over **3D voxel models**:



Scenes are represented using **scene graphs**:

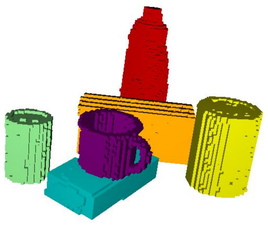
The graph structure defines the objects (nodes) in the scene and the physical contacts between them (edges)



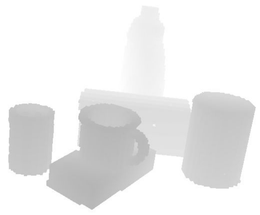


The parameters of the graph are are contact-constrained relative poses for each edge (or an absolute 6DoF pose for root nodes)

Together, the **3D voxel models** of shape, **scene graph structure**, and **scene graph parameters** define a 3D scene...



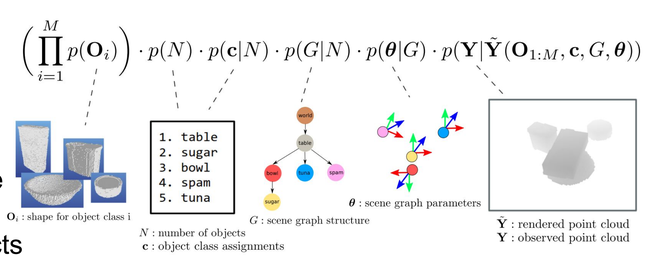
...and with a real-time graphics engine like OpenGL, the scene can be rendered into a depth image:



Our probabilistic program models this **generative process of a 3D scene**:

1. Sample 3D voxel shape from learned shape prior
2. Sample number and type of objects
3. Sample scene graph structure
4. Sample absolute poses for the root objects
5. Sample relative poses for objects in contact
6. Renderer an ideal depth image of the latent scene
7. Sample a point cloud from a likelihood parameterized by the ideal depth image, that injects noise and missing data

$$\left(\prod_{i=1}^{M} p(\mathbf{O}_i)\right) \cdot p(N) \cdot p(\mathbf{c}|N) \cdot p(G|N) \cdot p(\boldsymbol{\theta}|G) \cdot p(\mathbf{Y}|\tilde{\mathbf{Y}}(\mathbf{O}_{1:M}, \mathbf{c}, G, \boldsymbol{\theta}))$$

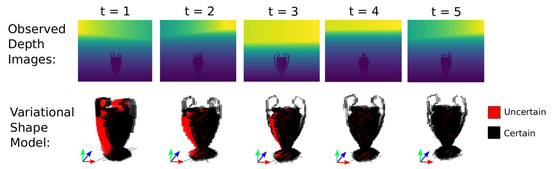


$\mathbf{O}_i$: shape for object class i
N : number of objects
$\mathbf{c}$: object class assignments
G : scene graph structure
$\boldsymbol{\theta}$: scene graph parameters
$\tilde{\mathbf{Y}}$: rendered point cloud
$\mathbf{Y}$: observed point cloud

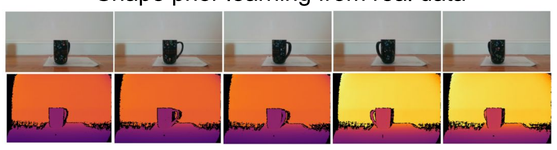
We have defined a probabilistic generative model of a latent 3D scene and depth image observation. Now, conditioning on an observed depth image, we can use inference in our generative model to infer the latent 3D scene:

2. Learning & Inference

Shape prior learning from synthetic data



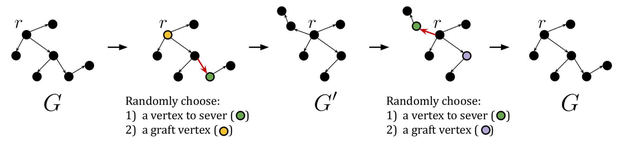
Shape prior learning from real data



Our system learns voxel shape models for new objects by variational inference from a handful of isolated views of the object. This learned distribution serves as the shape prior in the scene generative model. We show that we acquire an accurate 3D shape prior from just 5 depth images on both real and synthetic scenes.

Now, with the learned shape priors for each object type, we parse scenes containing them via inference over the scene graph structure and parameters.

For inference of scene graph structure, we use Involutive MCMC moves proposing to add/delete edges



For inference of the scene graph parameters, we use MCMC proposals incorporating heuristics such as iterative closest point (ICP) and random walk.

3. Results

We evaluate our method on the standard YCB-Video dataset of real RGBD images and a synthetic dataset of difficult scenes containing occlusions and physical contact. The task is to estimate all object poses from RGBD image input.

Results on synthetic data set

Scene Type	Object Class	Accuracy 3DP3	3DP3*	DF	R6D
Single Object	002_master_chef_can	**0.99**	0.95	0.45	0.03
	003_cracker_box	**0.55**	0.39	0.16	0.00
	004_sugar_box	**0.90**	0.87	0.17	0.00
	005_tomato_soup_can	**0.88**	0.81	0.18	0.00
	006_mustard_bottle	**0.86**	0.79	0.48	0.01
Single	002_master_chef_can	**0.86**	0.79	0.28	0.02
	003_cracker_box	**0.41**	0.24	0.16	0.00
	004_sugar_box	**0.63**	0.61	0.14	0.01
	005_tomato_soup_can	**0.67**	0.52	0.13	0.00
	006_mustard_bottle	**0.73**	0.60	0.44	0.03
Partial View	002_master_chef_can	**0.81**	0.80	0.11	0.00
	003_cracker_box	**0.18**	0.16	0.00	0.00
	004_sugar_box	**0.63**	0.59	0.00	0.00
	005_tomato_soup_can	**0.34**	0.33	0.00	0.00
	006_mustard_bottle	0.55	**0.62**	0.08	0.00
Partially Occluded	002_master_chef_can	**0.71**	0.52	0.04	0.00
	003_cracker_box	0.37	0.35	**0.59**	0.00
	004_sugar_box	0.02	0.01	**0.06**	0.00
	005_tomato_soup_can	**0.04**	0.00	0.01	0.00
	006_mustard_bottle	**0.70**	0.43	0.55	0.03

Results on real data set

Object Class	# of Scenes	0.5cm Threshold Accuracy 3DP3	3DP3*	DF
002_master_chef_can	1006	0.74	0.79	**0.84**
003_cracker_box	868	**0.90**	0.83	0.79
004_sugar_box	1182	**1.00**	0.99	0.98
005_tomato_soup_can	1440	**0.95**	0.93	0.93
006_mustard_bottle	357	**0.99**	0.98	0.94
007_tuna_fish_can	1148	0.81	0.80	**0.91**
008_pudding_box	214	**1.00**	0.97	0.70
009_gelatin_box	214	**1.00**	**1.00**	**1.00**
010_potted_meat_can	766	**0.80**	0.78	0.79
011_banana	379	**0.98**	0.96	0.82
019_pitcher_base	570	**1.00**	0.99	0.99
021_bleach_cleanser	1029	**0.94**	0.88	0.80
024_bowl	406	**0.93**	0.87	0.50
025_mug	636	0.89	0.89	**0.92**
035_power_drill	1057	**0.98**	0.96	0.88
036_wood_block	242	**0.36**	0.33	0.07
037_scissors	181	**0.75**	0.69	0.20
040_large_marker	648	**1.00**	**1.00**	0.99
051_large_clamp	712	**0.68**	0.64	0.25
052_extra_large_clamp	682	**0.33**	0.27	0.12
061_foam_brick	288	**0.26**	0.24	0.01

We measure the error between the predicted pose and the ground truth pose. Our system consistently outperforms the neural baselines and ablations of our method that don't treat contact specially. Below, we show the types of improvements our system makes over the baseline methods.