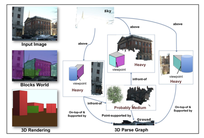
Structured, differentiable models of 3D scenes via Generative Scene Graphs

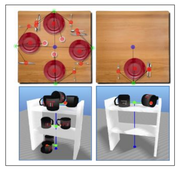
Ben Zinberg, Marco Cusumano-Towner, Vikash K. Mansinghka
MIT Probabilistic Computing Project

3D scene graphs in artificial intelligence

Perception of 3D scenes

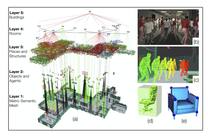
Blocks World Revisited
Gupta, Efros, Hebert (ECCV 2010)





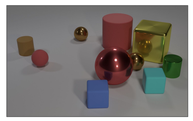
Scene Grammars
Izatt & Tedrake (ICRA 2020)

3D Dynamic Scene Graphs
Rosinol et al. (RSS 2020)



Generating synthetic data

CLEVR
Johnson et al. (CVPR 2017)

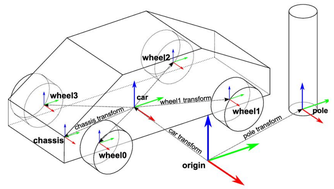


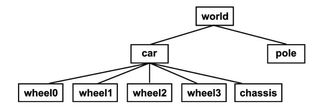
SCENIC
Fremont et al. (PLDI 2019)

Scene graph representations for scalable 3D graphics

Minimal car model:

7 objects, 23 polygons





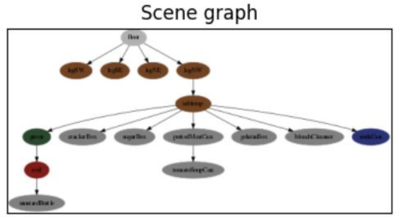
2015 game-engine car model:

~100s of objects, ~500K polygons

Generative scene graphs in Gen

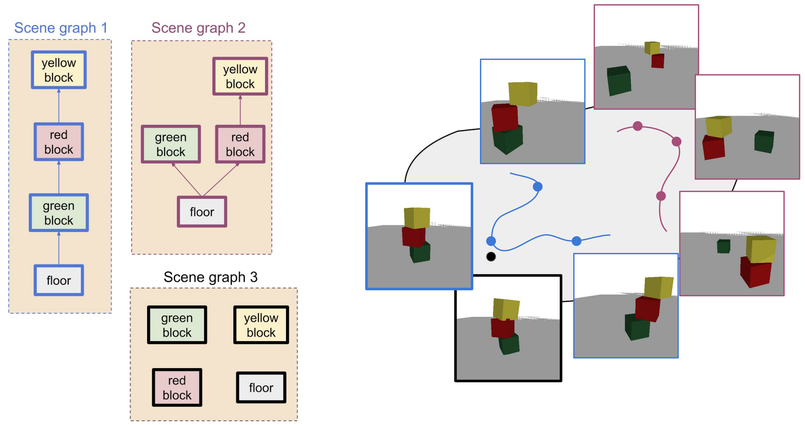
RGBA with bounding boxes

Scene graph



1. **Extensible object library** with simple shapes & YCB ground-truth meshes
2. **PyBullet renderer** for debugging and synthetic data generation
3. **Inference-friendly parameterizations** of 6DOF poses and contacts
4. **Intersection & occlusion testing** via bounding box approximations
5. **Gradient-based inference** for fine-tuning poses
6. **Monte Carlo inference** for structure learning & large-scale pose search

Structured submanifolds of the parameter space



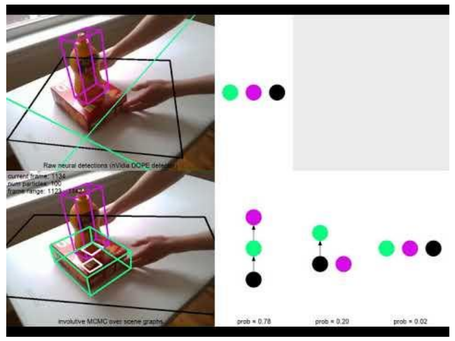
Perceiving 3D scene graphs

Deep learning object detections

(from nVidia DOPE)

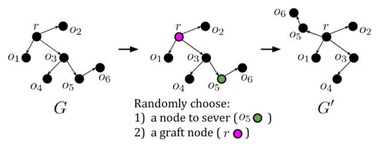
Inferring objects & contact graphs

(from Gen)



Garrett, Zinberg, Cusumano-Towner, Felip et al. (in prep)

Involutive MCMC in Gen: Structure Moves on Scene Graphs



Cusumano-Towner, Lew, and Mansinghka *(2018; PLDI 2019; arXiv 2020)*

```
@involution function toggleEdge(trace, parentObj, childObj)
  if trace[:edge_exists]
    # Delete the edge
    @write(trace, :edge_exists, false)
    relPose3D = (x, y, 0, slack) = trace[:contact => (parentObj, childObj)]
    relPose6D = Pose(x, y, slack.z, (yaw=0, pitch=slack.pitch, roll=slack.roll))
    @write(trace, :absolutePose => childObj,
    trace[:absolutePose => parentObj] ∘ relPose6D)
  else
    # Create the edge
    @write(trace, :edge_exists, true)
    (parentFace, childFace) = trace[:contactFace => (parentObj, childObj)]
    offset = (getContactPlane(trace[:shape => parentObj], parentFace)
              \ (trace[:absolutePose => parentObj]
                 \ trace[:absolutePose => childObj]
                 * getContactPlane(trace[:shape => childObj], childFace)))
    (closestPlanarContact, slack) = explainAsPlanarContact(poseOffset)
    @write(trace, :contact => (parentObj, childObj),
           (closestPlanarContact, slack))
  end
end
```

Current research – applications

1. **Robust 3D scene graph state estimation from noisy neural detections** via sequential Monte Carlo inference given a dynamic open-universe prior
2. **Grounded language understanding** via CCG parsing into GSGs
3. **Intuitive physics via probabilistic programming (IP3)** modeling object permanence, motion continuity, shape constancy, and collisions
4. **Generating synthetic data with controlled clutter & occlusion** to test systematic generalization of neural approaches to computer vision

Current research – scene graph PPL infrastructure

1. **Undirected scene graph edges** for over-determined contacts
2. **Optimized automatic differentiation** by exploiting scene graph structure
3. **Visualization of distributions over scene graphs** for modeling & inference
4. **Variational GSG approximations** to concisely summarize 3D scene posteriors