Neuro-Bayesian Methods for Realtime Scene Perception

by

Austin J. Garrett

Department of Electrical Engineering and Computer Science

Proposal for Thesis Research in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

December 2019

Author . . . .
Department of Electrical Engineering and Computer Science
December 11, 2019

Certified by. . . . .
Vikash Mansinghka
Research Scientist
Thesis Supervisor

Accepted by . . . .
Placeholder
Chairman, Department Committee on Graduate Theses

Neuro-Bayesian Methods for Realtime Scene Perception

by

Austin J. Garrett

Submitted to the Department of Electrical Engineering and Computer Science
on December 11, 2019, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, I consider Bayesian approaches to doing inference over structured scenes. This includes the design and implementation of a probabilistic language for representing semantic relations between multiple objects. I further consider a strategy for integrating neural techniques into MCMC-based techniques for inverse graphics, where a generative model is constructed to predict the behavior of neural detectors, allowing for error correction and semantic structuring of the output. I finally discuss the development of a framework for realtime particle filter based inference based on the Robot Operating System and Gen probabilistic programming language, as well as its integration into the broader Cora project for probabilistic modeling of intuitive physics.

Thesis Supervisor: Vikash Mansinghka
Title: Research Scientist

Chapter 1

Thesis Proposal

1.1 Introduction

The past decade has seen a flourishing of highly powerful computer vision techniques, due to the success of deep learning techniques accelerated on massively parallel hardware.

Yet the shortcomings of neural techniques have only grown increasingly urgent in proportion to their dominance. While these approaches have resulted in the amortization of traditionally very difficult and high-dimensional computer vision problems, the lack of semantic structuring in their output has led to great challenges in their deployment in practical domains where interpretability and composability are desperately needed to integrate these components into larger frameworks.

Meanwhile probabilistic techniques offer a more principled approach to modeling structured data, and have seen a rapid growth in the collective ecosystem. Such environments are desirable for their expressibility, corresponding to the fact that they often represent intuitive models of rationality inspired from cognitive and neuroscience which increasingly rely on such modeling to explain the human reasoning.

1.2 Prior and Related Works

1.2.1 Accelerating Inverse Graphics

An alternative perspective on the problem of pose estimation comes from the perspective of “analysis-by-synthesis”, in which a generative model allows for MCMC-based inference, integrating a notion of uncertainty and robustness into the estimation procedure. Typically to make inference efficient, these approaches include some amount of discriminative training to amortize the inference procedure with data-driven kernels. *Picture* is a probabilistic programming language for scene perception that proposed among others, arbitrary blocked Gibbs moves, gradient-based proposals, and elliptical slice sampling in order to incorporate bottom-up amortization of MCMC [2]. Jampani et. al. created the *informed sampler*, a mixed inference kernel between a trained state-independent discriminator proposal, and a local Gaussian metropolis-hastings move [1]. There has also been work on directly using neural networks as initialization for MCMC. Yildirim et. al. leveraged convolutional neural networks to initialize a Markov chain for face processing [6].

1.2.2 Neural Approaches to Pose Estimation

Only recently have neural approaches successfully begun to make headway on full 6D (3 spatial, 3 rotational) pose estimation for multiple objects. These algorithms offer a powerful modeling perspective that can be used as an intermediate step for segmentation, bounding box regression, or feature extraction. However, they often suffer from the same brittleness and lack of insight as deep learning in general.

[5] combines heuristical Hough voting and convolutional feature extraction to approach the problem. [4] uses a neural network to estimate belief maps of keypoints in 2D image coordinates that are fed into a standard perspective-n-point (PnP) algorithm to recover the full 6D pose. It was trained only on synthetic data, using domain randomization to generalize to the real world. [3] uses a render-and-compare based loss inspired by inverse graphics methods to train the neural regressor.

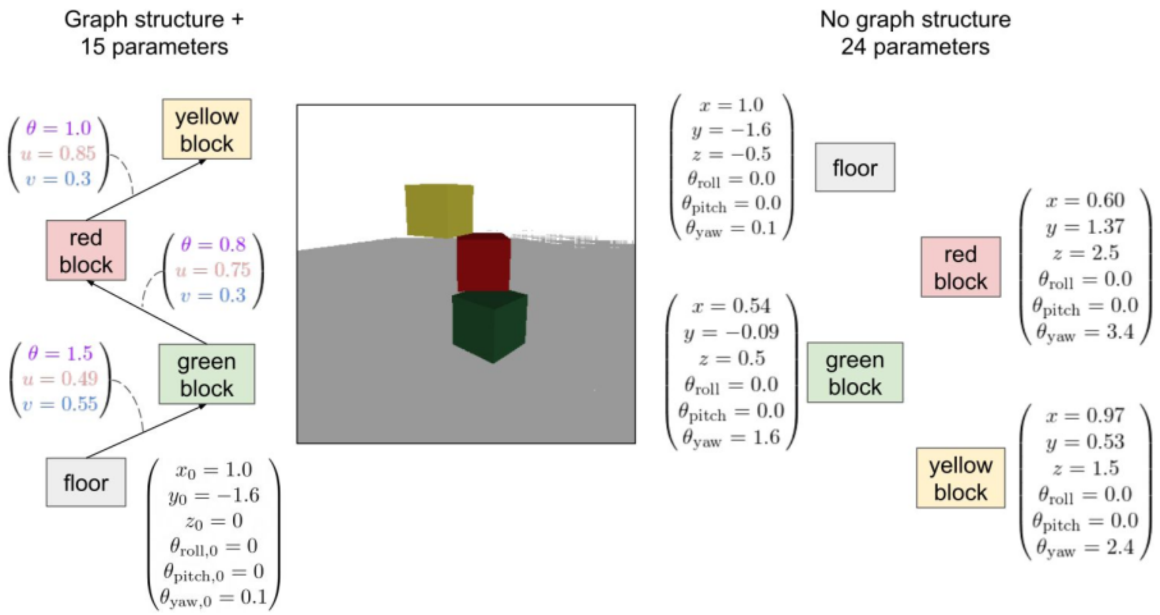

Graph structure +
15 parameters
No graph structure
24 parameters
yellow block
red block
green block
floor
$\theta = 1.0$
$u = 0.85$
$v = 0.3$
$\theta = 0.8$
$u = 0.75$
$v = 0.3$
$\theta = 1.5$
$u = 0.49$
$v = 0.55$
$x_0 = 1.0$
$y_0 = -1.6$
$z_0 = 0$
$\theta_{\text{roll},0} = 0$
$\theta_{\text{pitch},0} = 0$
$\theta_{\text{yaw},0} = 0.1$
floor
$x = 1.0$
$y = -1.6$
$z = -0.5$
$\theta_{\text{roll}} = 0.0$
$\theta_{\text{pitch}} = 0.0$
$\theta_{\text{yaw}} = 0.1$
green block
$x = 0.54$
$y = -0.09$
$z = 0.5$
$\theta_{\text{roll}} = 0.0$
$\theta_{\text{pitch}} = 0.0$
$\theta_{\text{yaw}} = 1.6$
red block
$x = 0.60$
$y = 1.37$
$z = 2.5$
$\theta_{\text{roll}} = 0.0$
$\theta_{\text{pitch}} = 0.0$
$\theta_{\text{yaw}} = 3.4$
yellow block
$x = 0.97$
$y = 0.53$
$z = 1.5$
$\theta_{\text{roll}} = 0.0$
$\theta_{\text{pitch}} = 0.0$
$\theta_{\text{yaw}} = 2.4$

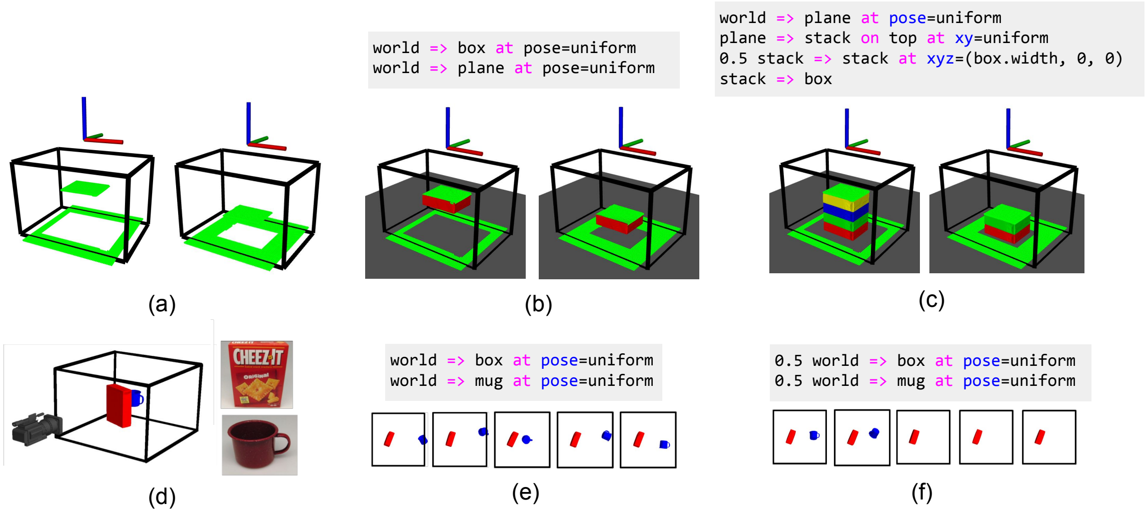
Figure 1-1: Two identical representations of three blocks stacked on top of a floor in a virtual environment. The right side assumes no semantic relational information, with 24 parameters required to specify the positions of all the objects in the scene. The left side represents semantic information about the relationship between objects, namely edges represent an "on-top of" relationship. Under this scene structure, only 15 parameters are necessary for specifying the same scene. TODO: cite Ben

1.2.3 Bayesian Inference over Structured Scenes

Extending from this previous work on inverse graphics, we wish to construct generative models that contain richer *semantic information* about objects in the scene – namely human-interpretable information about the relations between objects. This can be crucial in reducing the complexity of inference by reducing the dimensionality of the parametrization to be inferred.

Scene Graph Representation

To encode structural information, we leverage a common computer vision representation called a *scene graph*. In this graph, poses are represented as transformations from object to object, or from implicit world frame to object. Directed edges specify the type of relative pose, which encodes information about the way objects are geometrically situated. For example, an edge may represent "contact", which requires specifying a discrete plane of contact, a 2D translation, and a single rotational degree



```
world => box at pose=uniform
world => plane at pose=uniform
```

```
world => plane at pose=uniform
plane => stack on top at xy=uniform
0.5 stack => stack at xyz=(box.width, 0, 0)
stack => box
```

```
world => box at pose=uniform
world => mug at pose=uniform
```

```
0.5 world => box at pose=uniform
0.5 world => mug at pose=uniform
```

Figure 1-2: Two scenarios with prior knowledge specified as programs in our probabilistic scene description language. (a) shows two depth measurements made by a depth camera. (b) shows a program in which boxes have random poses in 3D space, and the resulting inferred pose of the box. (c) shows a program that assumes boxes are in stacks that rest on the floor, and the inferred number of boxes that explain the data for each observation. (d) shows another scenario, where a depth camera observes a box that is occluding a mug. (e) shows a program that asserts that both objects exist (but at unknown poses). The resulting inferences show the mug must exist somewhere behind the box to be consistent with this knowledge and the observation. (f) shows a program that allows for either object to not exist, and the resulting joint inferences about the mug's existence and pose.

about the plane's normal.

Probabilistic Scene Description Languages

It can often be desirable to specify knowledge in a programmatic way. One way to define a generative model over a scene graph representation is through a probabilistic context free grammar called a *scene description language*. Such a language can further increase the expressibility of the scene graph concept by adding existential uncertainty to the objects themselves. This is encoded by some probability over an expansion of the production rules in the PCFG.

Even without the use of data-driven proposals, we can demonstrate how this language for uncertain knowledge representation can recover common-sense reasoning that naturally combines prior knowledge with observation to obtain posteriors with rich structural information. TODO: expand me, relate back to scene graph more?

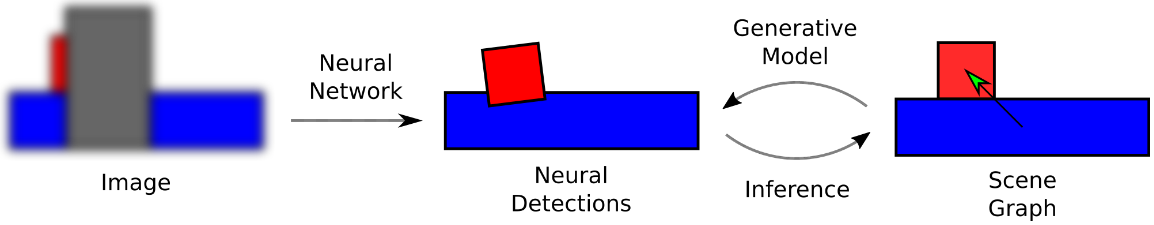


Figure 1-3: Neural detections can be inaccurate, and violate semantic relations between objects (eg. allow for object interpenetration). Given a neural detector, we can treat the flat and noisy detections as observations under a generative model, given prior structural information about the scene. Thus we implicitly model and correct for the failure modes of neural detections using uncertain prior structural knowledge. Using custom inference kernels, we can potentially recover the network's uncertainty, resolve impossible scenarios that violate prior knowledge, or even recover qualitative pose relationships like contact (green arrow) to enrich neural detections.

1.3 Proposed Work

Month	Work to be completed
December	Thesis proposal and ROS-based infrastructure for online particle filtering and associated visualization
January	Experimentation with observational models for modeling neural detections. Writing submission to the RSS conference.
February	Refactoring code base and integration of the ROS particle filter and visualization components into Cora.
March	Further integration/experimentation with the neural detector component (training on synthetic data, replacing with alternative architectures, training the neural component with error-correction on inference.)
April	Development of thesis, documentation of work, and writing.
May	Finalization of thesis and final documentation of work and code base.

1.3.1 Neuro-predictive Generative Modeling

Modeling a rendering pipeline in a fully Bayesian way suffers from certain computational challenges. While guaranteed to converge to a correct posterior with unlimited compute time, the non-asymptotic properties of MCMC are poorly-understood. Preliminary work in inverse graphics techniques applied to structured scenes supports the hypothesis that naive analysis-by-synthesis approaches that leverage a full rendering pipeline often fail to explore all the modes of the posterior in reasonable time bounds

TODO: WHO?

Concretely, for scene graph $\mathcal{S}$ and continuous parameterization $\vec{\nu}_{\mathcal{S}}$, a rendering-based likelihood relies on modeling the image data using a full rendering pipeline $R(\mathcal{S}, \vec{\nu}_{\mathcal{S}}) = X$, where X is a rendered image. For robustness, the likelihood is modeled not directly on X, but on a noisy function of the pixel data. The noise is modeled is a mixture of a uniform distribution on the range of possible values and a normal distribution with mean centered at the true rendered pixel value $R(\mathcal{S}, \vec{\nu}_{\mathcal{S}})$ with fixed variance σ^2, leaving the full likelihood on noisy image data Y as

$$p(Y|\mathcal{S}, \vec{\nu}_{\mathcal{S}}) = \prod_{r=1}^{R} \prod_{c=1}^{C} \left(0.1 \cdot \frac{1}{D} + 0.9 \cdot \mathcal{N}(Y_{r,c}; R(\mathcal{S}, \vec{\nu}_{\mathcal{S}})_{r,c}, \sigma)\right) \tag{1.1}$$

1.1 is very high-dimensional, making it prone to capture by local minima. Neural networks empirically demonstrate strong performance at locating strong *maximum a posteriori* estimates in bounded compute time. Thus, we may consider combining neural techniques with MCMC to improve performance of inference.

One perspective may consider neural networks as amortizations of the inverse generative procedure. Consider a neural detector $\phi_{\mathcal{S}}$ that outputs poses under a given parametrization $\mathcal{S}$ (often a full unconstrained 6D pose), such that ideally $\phi_{\mathcal{S}}(R(\mathcal{S}, \vec{\nu}_{\mathcal{S}})) = \vec{\nu}_{\mathcal{S}}$. Given image data Y, we can use the corresponding estimate $\phi_{\mathcal{S}}(Y)$ as an initialization for some MCMC technique.

However, this method is rather unprincipled in that it uses a point estimate without uncertainty quantification, thus implicitly relying on $\phi_{\mathcal{S}}$ robustly predicting the mode of the distribution. This assumption is not guaranteed, and in fact when neural networks do fail, they often fail catastrophically, estimating wildly incorrect poses. Furthermore, because it is only a heuristic and not a full proposal distribution, it is not obvious how to combine neural detections across multiple timesteps into a coherent picture. The issue fundamentally lies in the fact that this approach contains no way to reason about the behavior of the bottom-up proposals. If we wish to robustly use these models, we need to leverage information about their behavior.

Crucially, we can observe that the failure modes of these neural techniques are often quite predictable with the use of certain easy to compute statistics. When objects are too close or too far away, at strange angles, or heavily occluded, the neural detector is much more likely to fail. To account for these configurations, we propose instead modeling the neural detections as observations to a generative model that attempts to retrieve the latent scene graph $\mathcal{S}$ that predicts the detections from the neural network $\phi_{\mathcal{S}}(Y)$.

A minimal observational model may be a simple mixture between a Gaussian and a uniform distribution over the entire space

$$p(\phi_{\mathcal{S}}(Y)|\mathcal{S}, \vec{\nu}_{\mathcal{S}}) = \mathcal{N}() \tag{1.2}$$

Even such a simple observational model can be sufficient to filter some noise from the bottom-up proposal, by roughly modeling that the neural network sometimes makes mistakes (although in this case we don't use information about whether this is more or less likely at any given time).

1.3.2 Experimentation

The first task concrete task is creating the infrastructure sufficient for creating a minimal demo. For this we will develop an end-to-end system that uses DOPE+Gen+ROS to track a fixed set of objects, integrating prior knowledge about temporal consistency of object trajectories via state space modeling. For this, we can use a simple particle filter over object trajectories. This will help us nail down the end-to-end integration aspect, and also serves as the minimal example of "filtering the output of a neural network via prior knowledge". For simplicity we will initially work with an observation model of the form 1.2. Even with this simple model, we hypothesis it is possible to filter spurious neural detections using physical assumptions of object persistence and intertial trajectories.

Use the full pipeline to infer scene structure, integrating prior knowledge about scene graphs (e.g. contact relationships) as well as temporal consistency. This version will assume a static scene graph, but can later be adapted into an online algorithm by running the algorithm on the past e.g. 10 time steps of data in sliding windows. Use the full pipeline with a prior about temporal consistency of scene graphs (e.g. scene graphs change relatively infrequently).

Reversible Jump MCMC

Allowing MH moves between model parametrizations requires special mathematical consideration to ensure detailed balance is satisfied. TODO: Insert math

1.3.3 Engineering Infrastructure

All of this experimentation requires sophisticated and novel infrastructure for the prototyping of complex custom inference kernels and observational models to be applied in realtime

Realtime Particle Filtering

Visualization

TODO: replace me

Cora Project

TODO: replace me

1.3.4 Extensions

If time permits, there are several possible extensions to this work that may be considered. One relates TODO: replace me

1.4 Conclusion

TODO: replace me

Bibliography

[1] Varun Jampani, Sebastian Nowozin, Matthew Loper, and Peter V. Gehler. The informed sampler: A discriminative approach to bayesian inference in generative computer vision models. *CoRR*, abs/1402.0859, 2014.

[2] Tejas D Kulkarni, Pushmeet Kohli, Joshua B Tenenbaum, and Vikash Mansinghka. Picture: A probabilistic programming language for scene perception. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4390–4399, 2015.

[3] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018.

[4] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *CoRR*, abs/1809.10790, 2018.

[5] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[6] Ilker Yildirim, Tejas D Kulkarni, Winrich A Freiwald, and Joshua B Tenenbaum. Efficient analysis-by-synthesis in vision: A computational framework, behavioral tests, and comparison with neural representations. In *Proceedings of the thirty-seventh annual conference of the cognitive science society*, 2015.