PERCH: Perception via Search for Multi-Object Recognition and Localization

Venkatraman Narayanan and Maxim Likhachev
The Robotics Institute, Carnegie Mellon University

In many robotic domains such as flexible automated manufacturing or personal assistance, a fundamental perception task is that of identifying and localizing objects whose 3D models are known. Canonical approaches to this problem include discriminative methods that find correspondences between feature descriptors computed over the model and observed data. While these methods have been employed successfully, they can be unreliable when the feature descriptors cannot capture variations in observed data: a classic example being occlusion. As a step towards deliberative reasoning, we present PERCH: PErception via SeaRCH, an algorithm that seeks to find the best explanation of the observed sensor data by hypothesizing possible scenes in a generative fashion. Our contributions are: i) formulating the multi-object recognition and localization task as an optimization problem over the space of hypothesized scenes, ii) exploiting structure in the optimization to cast it as a combinatorial search problem on what we call the Monotone Scene Generation Tree, and iii) leveraging parallelization and recent advances in multi-heuristic search in making combinatorial search tractable. We prove that our system can guaranteeably produce the best explanation of the scene under the chosen cost function, and validate our claims on real world RGB-D test data. Our experimental results show that we can identify and localize objects under heavy occlusion—cases where state-of-the-art methods struggle.

While model-based recognition and pose estimation of objects has been an active area of research for decades in the computer vision community [4, 10], the proliferation of low-cost depth sensors such as the Microsoft Kinect has introduced a plethora of opportunities and challenges. Model-based object recognition and localization in the present 3D era falls broadly under two approaches: *local* and *global* recognition systems. The former operate by matching local 3D descriptors (e.g., Spin Images [7], Fast Point Feature Histograms (FPFH) [11]) between the model and test scenes and then estimating a geometrically feasible rigid transform. *Global* recognition systems encode the notion of an object by capturing shape and viewpoint information jointly in a descriptor. These approaches employ a training phase to build a library of global descriptors corresponding to different observed instances (for e.g., an object viewed from different viewpoints) and attempt to match the descriptor computed at observation time to the closest one in the library. Examples of such systems include Clustered Viewpoint Feature Histogram (CVFH) [1], OUR-CVFH [3], Global Radius-based Surface Descriptors (GRSD) [8] etc. Other approaches to estimating object pose include local voting schemes [5] or template matching [6] to first detect objects, and then using global descriptor matching or ICP for pose refinement.

Although both local and global feature-based approaches have enjoyed popularity owing to their speed and intuitive appeal, they suffer when used for identifying and localizing multiple objects in the scene (Fig. 1). The limitation is perhaps best described by the following lines from the book by Stevens and Beveridge [12]: "Searching for individual objects in isolation precludes explicit reasoning about occlusion. Although the absence of a model feature can be detected (i.e., no corresponding data feature), the absence cannot be explained (why is there no corresponding data feature?). As the number of missing features increase, recognition performance degrades". In this work, we present an approach that explicitly models inter-object occlusion through rendering possible configurations of objects.

Technical Details. The problem we consider is that of localizing tabletop objects from depth data such as a full point cloud, or a 2.5D Kinect sensor. The problem statement is as follows: given 3D models of N unique objects, a point cloud (I) of a scene containing $K \geq N$ objects (possibly containing replicates of the N unique objects), and the 6 degrees of freedom (DoF) pose of the sensor, we are required to find the 3 DoF pose (x, y, θ) of each of the K objects in the scene. We make the following assumptions: a)

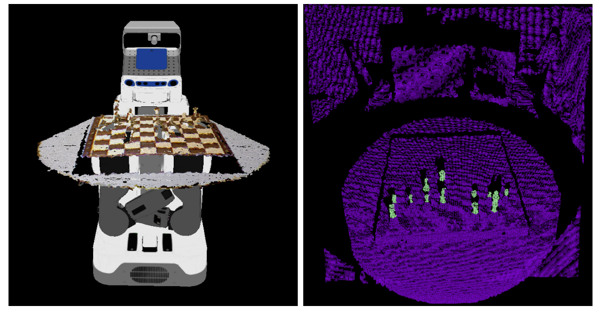


Figure 1: Identifying and localizing the pose of multiple objects simultaneously is challenging in many domains such as robotic manipulation because of inter-object occlusions. Illustration shows multiple chess pieces occluding each other.

The number (K) and type of objects in the scene are known ahead of time (but not the correspondences themselves), b) The objects in the scene vary only in position and yaw (3 DoF) with respect to their 3D models, and c) We have access to the intrinsic parameters of the sensor, so that we can render scenes using the available 3D models.

We formulate the problem of identifying and obtaining the 3 DoF poses of objects $O_1, O_2, \ldots, O_K$ as that of finding the minimizer of the following 'explanation cost':

$$J(O_{1:K}) = J_{observed}(O_{1:K}) + J_{rendered}(O_{1:K})$$
$$J_{observed}(O_{1:K}) = \sum_{p \in I} \mathbb{1}_{[p \text{ is unexplained by } R_K]}$$
$$J_{rendered}(O_{1:K}) = \sum_{p \in R_K} \mathbb{1}_{[p \text{ is unexplained by } I]}$$

where R_K is the rendered point cloud containing K objects and the indicator function $\mathbb{1}_{[p \text{ is unexplained by } C]}$ for a point cloud C and point p is:

$$\mathbb{1}_{[p \text{ is unexplained by } C]} = \begin{cases} 1 & \text{if } \min_{p' \in C} \|p' - p\| > \delta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for some sensor noise threshold δ. Intuitively, the formulation seeks a rendering of the scene that best explains the observed sensor data. In the ideal scenario where there is no noise in the observed scene and where we have access to a perfect renderer, we could do an exhaustive search over the joint object poses to obtain a solution with zero cost. However, this naive approach is a recipe for computational disaster: even when we have only 3 objects in the scene and discretize our positions to 100 grid locations and 10 different orientations, we would have to synthesize/render 10^9 scenes to find the global optimum.

The following insight allows us to circumvent exhaustive search: by enforcing a specific ordering in which object poses are assigned, the explanation cost can be decomposed into a sum of per-object costs, thereby paving the way for a smarter search scheme. Specifically, the constraint on the ordering is that every time an object is added to the scene, it does not occlude any of the existing objects. In other words, the number of points in the rendered point cloud should be monotonically non-decreasing. This constraint on the ordering and the decomposition of the cost function results in the minimization problem being reduced to a tree search problem on what we call the Monotone Scene Generation Tree (MSGT). Specifically, the minimization problem is now equivalent to finding the shortest-cost path from the root node (empty scene) to a goal node (state with all object poses assigned) in the MSGT (Fig. 2). Although tree-search is much more tractable than exhaustive search over the joint object poses, the branching factor is still large. To speed up search, we leverage recent work in heuristic search [9] that permits the use of multiple arbitrary heuristics, while still preserving guarantees on completeness and bounds on solution quality.

This research was sponsored by ARL, under the Robotics CTA program grant W911NF-10-2-0016.

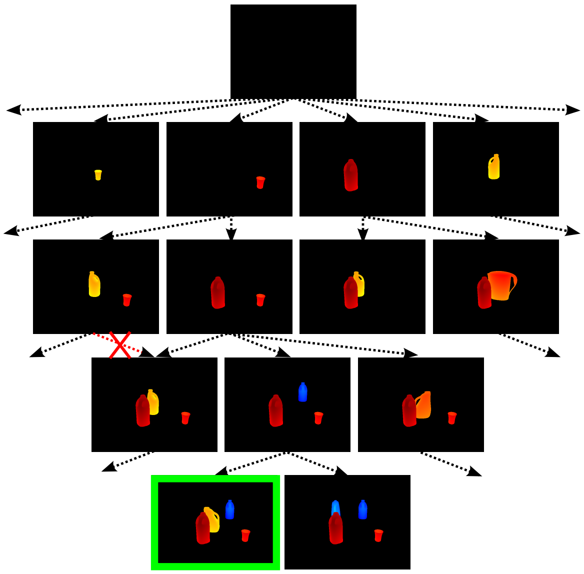


Figure 2: Portion of a Monotone Scene Generation Tree (MSGT): the root of the tree is the empty scene, and new objects are added progressively as we traverse down the tree. Notice how child states never introduce an object that occludes objects already in the parent state. A counter-example (marked by the red cross) is also shown. Any state on the K^{th} level of the tree is a goal state, and the task is to find the one that has the lowest cost path from the root—marked by a green bounding box in this example.

Experiments. To evaluate the performance of PERCH for multi-object recognition and pose estimation in challenging scenarios where objects could be occluding each other, we pick the occlusion dataset described by Aldoma et al. [2] that contains objects partially touching and occluding each other. The dataset contains 3D CAD models of 36 common household objects, and 22 RGB-D tabletop scenes with 80 object instances that vary only in yaw and translation. We compared PERCH with two baselines: the first is the OUR-CVFH descriptor [3] that was designed to be robust to occlusions. We trained the OUR-CVFH pipeline by rendering 642 views of every 3D CAD model from viewpoints sampled around the object. Our second baseline is an ICP-based optimization one, which we refer to as Brute Force without Rendering (BFw/oR). Here, we slide the 3D model of every object in the scene over the observed point cloud, and perform a local ICP-alignment at every step. The set of object poses that have the best total fitness score is taken as the solution.

Figure 3 shows some qualitative examples of PERCH's results, while Fig. 4 provides a quantitative comparison with the baselines. The latter shows the number of correct poses produced by each of the methods (out of 80 objects), for the following definition of 'correct pose': a predicted pose (x, y, θ) for an object is considered correct if $\|(x,y) - (x_{\text{true}}, y_{\text{true}})\|_2 < \Delta t$ and $\textsc{ShortestAngularDifference}(\theta, \theta_{\text{true}}) < \Delta\theta$. We see that PERCH consistently dominates the baselines for different definitions of correct pose, with the improvements being most significant when very accurate poses are desired (translation error under 1 cm). The most computationally expensive part of PERCH, rendering scenes during the search, is embarrassingly parallel. We parallelized our implementation with the MPI framework and ran the experiments on an Amazon AWS cluster of 2 m4.10x machines with 40 virtual cores each. The mean time to find a solution per scene was 6.5 minutes. We also note that PERCH does not require any training time, unlike the global descriptor pipelines.

In summary, we presented PERCH, an algorithm for multi-object recognition and localization that uses search to find the 'best' explanation of an observed scene. Our contributions were the formulation of multi-object localization as an optimization over rendered scenes, and exploiting structure in the optimization to cast it as a tree search problem. Our results demonstrate that PERCH can robustly identify and localize objects even under heavy occlusion.

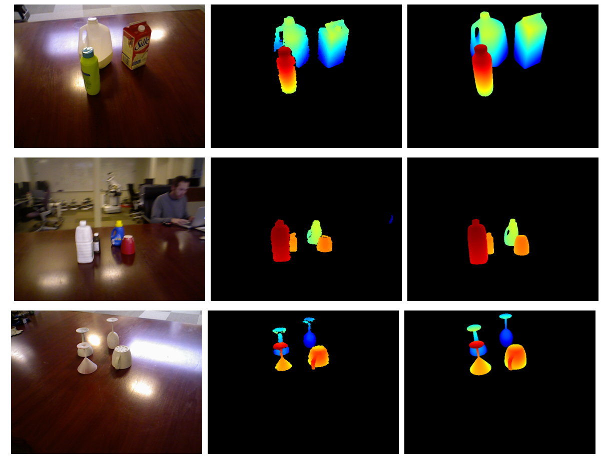
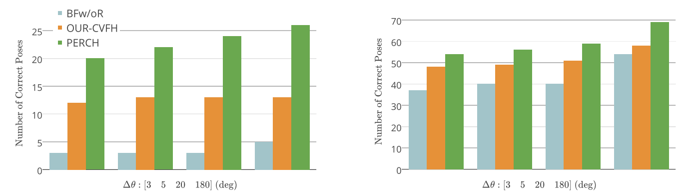


Figure 3: Examples showing the output of PERCH on the occlusion dataset. *Left*: RGB-D scenes in the dataset. *Middle*: Depth images of the corresponding input RGB-D scenes, *Right*: The depth image reconstructed by PERCH through rendering object poses.



(a) $\Delta t = 0.01m$ (b) $\Delta t = 0.1m$

Figure 4: Number of objects whose poses were correctly classified by the baseline methods (BFw/oR, OUR-CVFH) and PERCH, for different definitions of 'correct pose'.

[1] Aitor Aldoma, Markus Vincze, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu, and Gary Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 585–592. IEEE, 2011.

[2] Aitor Aldoma, Zoltan-Csaba Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. Point cloud library. *IEEE Robotics & Automation Magazine*, 1070(9932/12), 2012.

[3] Aitor Aldoma, Federico Tombari, Radu Bogdan Rusu, and Markus Vincze. *OUR-CVFH–Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation*. Springer, 2012.

[4] Rodney A Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial intelligence*, 17(1):285–348, 1981.

[5] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, pages 998–1005. IEEE, 2010.

[6] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision–ACCV 2012*. Springer.

[7] Andrew E Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[8] Zoltan-Csaba Marton, Dejan Pangercic, Nico Blodow, and Michael Beetz. Combined 2d–3d categorization and classification for multimodal perception systems. *IJRR*, 30(11):1378–1402, 2011.

[9] Venkatraman Narayanan, Sandip Aine, and Maxim Likhachev. Improved Multi-Heuristic A* for Searching with Uncalibrated Heuristics. In *Eighth Annual Symposium on Combinatorial Search*, 2015.

[10] Lawrence Gilman Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[11] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*. IEEE, 2009.

[12] Mark R Stevens and J Ross Beveridge. *Integrating graphics and vision for object recognition*, volume 589. Springer Science & Business Media, 2000.