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 guaranteeadly 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 [3] 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 increases, 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 tablopt 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 $(x, y, \theta)$ of each of the $K$ objects in the scene. We make the following assumptions: a)
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 \((x, y, \theta)\) for an object is considered correct if \(||(x, y) - (x_{\text{true}}, y_{\text{true}})||_2 < \Delta t\) and \(\text{shortestangulardifference}(\theta, \theta_{\text{true}}) < \Delta \theta\). 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.